

ULRR

Using RFID to output reliability indicators in PC refurbishment

Item Type	Thesis
Authors	Cronin, Éanna
Download date	2026-06-09 14:53:14
Item License	https://creativecommons.org/licenses/by-nc-sa/1.0/
Link to Item	https://hdl.handle.net/10344/2835



UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

**Using RFID to Output Reliability Indicators
in PC Refurbishment**

by

Éanna Cronin

A thesis submitted to the University of Limerick
in fulfilment of the requirements for the degree of

Doctor of Philosophy

in the

Department of Computer and Electronic Engineering
University of Limerick
Ireland

Supervisor

Dr. Colin Fitzpatrick, University of Limerick, Ireland

ABSTRACT

In order to foster a market for reuse, environmental impacts must be examined across the entire life cycle of a product. For products such as Personal Computers (PCs), research suggests that the reduction of manufacturing burden through lifetime extension is a preferable means of mitigating some of the adverse effects associated with the total life-cycle impacts associated with these products. As a means of overcoming distrust in the secondary PC market, information can be provided to buyers about the quality of a system, and the value can be determined in the context of market forces. The work presented here outlines a process for the storage and extraction of information which utilises the environmental thermal conditions of a PC throughout its lifetime to indicate the expected continued reliability of the system. The requirements for a technological solution for extraction of this information are presented, and Radio Frequency Identification (RFID) is suggested in the context of its capability to fulfil these requirements. The work presents the process for development of an RFID system for extraction of data, including extensions to existing RFID protocols, allowing the transmission of the volumes of data which need to be considered in extracting enough data to characterise PC lifetimes. The use of load characterisation for reduction of total data volume is presented, and the final solution presented here performs load characterisation in hardware. While the results suggest that the developed technology is not perfectly appropriate for the function outlined, there is considerable potential for further developments in the area.

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Name

Date

DEDICATION

*To my parents, Deirdre and Jerry
For their unwavering patience*

Civilization is a limitless multiplication of unnecessary necessities.

- *Mark Twain*

ACKNOWLEDGEMENTS

This work would not have been possible without the support of Dr. Colin Fitzpatrick, my supervisor. Throughout the process of the doctorate he has been a constant source of guidance and encouragement, and had he not always been available with advice there is no way that this dissertation would have ever reached the point of completion. The advice provided by my examiners, Dr Thomas Newe and Dr David Chesmore was also critical to the completion of this work.

The friendship and assistance of friends and colleagues in the research group has also been invaluable throughout this process. The co-researchers in the group have provided support and advice during difficult times, and in particular without Paddy Finn, Stewart Hickey and Maurice O'Connell, the doctoral process would have been far less agreeable.

Leah Bernini has become a pillar of strength over the last number of years, providing love, encouragement and emotional support throughout the journey, and advice and strength whenever needed. Her patience, understanding and counsel provided precious motivation when all other sources ran dry.

Finally, without the financial support of Deirdre Fay and Jerry Cronin, the work would have come to a premature halt some time ago. Their sympathy, tolerance and home-cooked meals are the foundation on which this thesis is built.

This work is funded as part of the Science, Technology, Research and Innovation for the Environment (STRIVE) Programme 2007-2013. The programme is financed by the Irish Government under the National Development Plan 2007-2013. It is administered on behalf of the Department of the Environment, Heritage and Local Government by the Environmental Protection Agency which has the statutory function of co-ordinating and promoting environmental research.

TABLE OF CONTENTS

Abstract	i
Declaration	iii
Dedication	v
Acknowledgements	ix
List of Tables	xvii
List of Figures	xix
List of Appendices	xxi
List of Abbreviations	xxiii
Chapter 1 Introduction	1
1.1 Contribution	2
1.2 Outline	3
Chapter 2 Literature Review	7
2.1 Environmental Impact	7
2.1.1 <i>Manufacturing</i>	8
2.1.2 <i>Use Phase</i>	9
2.1.3 <i>End of Life</i>	12
2.2 Reliability Monitoring	17
2.3 Life Consumption Monitoring	19
2.4 Factors Impacting Reliability	21
2.4.1 <i>Operating Time</i>	21
2.4.2 <i>Temperature</i>	22
2.4.3 <i>Temperature Cycles</i>	23
2.4.4 <i>Humidity</i>	25
2.4.5 <i>Vibrations and Shocks</i>	26
2.5 Radio Frequency Identification	27

2.5.1 Applications and Technologies.....	27
2.5.2 Principles of Operation.....	29
2.5.3 Low Frequency.....	30
2.5.4 High Frequency.....	31
2.5.5 Ultra-High Frequency.....	31
2.5.6 Microwave.....	32
2.6 Smart Tags.....	32
2.7 Conclusion.....	34
2.8 References.....	35
Chapter 3 Background & State of the Industry.....	41
3.1 Introduction.....	41
3.2 Environmental Benefits.....	42
3.3 Technological Reliance: Social and Economic.....	43
3.3.1 Social Effects.....	43
3.3.2 Economic Effects.....	48
3.3.3 Conclusion.....	49
3.4 Environmental Stewardship.....	50
3.4.1 Environmental Labelling.....	50
3.4.2 Energy Star.....	51
3.4.3 EPEAT.....	52
3.4.4 US Executive Order.....	53
3.5 Technological Trends.....	53
3.5.1 Cloud Computing.....	54
3.5.2 Digital Delivery.....	56
3.5.3 Solid-State drives.....	57
3.6 End of Use.....	58
3.6.1 Functional and Physical Obsolescence.....	58
3.6.2 Storage.....	61
3.6.3 Remanufacturing.....	61
3.6.4 Reuse.....	62
3.7 Conclusion.....	63

3.8	References	64
Chapter 4 Proposed Solution: RFID and LCM.....67		
4.1	Introduction	67
4.2	Signalling.....	68
4.3	Triage.....	70
4.4	Proposed Solution.....	72
4.4.1	<i>Prototype development</i>	74
4.5	Advantages of Data Availability	76
4.6	Data Reduction	78
4.6.1	<i>Ordered Overall Range</i>	79
4.6.2	<i>Rain flow Cycle Counting</i>	80
4.6.3	<i>Implementation</i>	82
4.7	Conclusion.....	85
4.8	References	86
Chapter 5 Methodology: Software Development89		
5.1	Introduction	89
5.2	Modulation	90
5.2.1	<i>PCD to PICC</i>	90
5.2.2	<i>PICC to PCD</i>	92
5.2.3	<i>Error Checking</i>	93
5.3	Physical Communications Interface	93
5.3.1	<i>Software Solution</i>	94
5.3.2	<i>Encoding</i>	95
5.3.3	<i>Decoding</i>	99
5.4	State Machine Implementation.....	101
5.5	Simulation.....	106
5.5.1	<i>Stimuli</i>	108
5.5.2	<i>Logging</i>	113
5.6	Data Reduction - ATMega16 Software	114
5.7	Data Reduction - PC Software.....	120
5.8	Conclusion.....	120

Chapter 6 Methodology: Hardware Development	123
6.1 Introduction	123
6.2 Programming Interface	124
6.3 RFID hardware	126
6.3.1 Receiver	126
6.3.2 Transmitter	131
6.3.3 Signalling & reset.....	133
6.4 Software modifications.....	134
6.5 Optimal timing.....	136
6.6 USB Interface Hardware	138
6.7 Conclusion.....	139
Chapter 7 Results.....	141
7.1 Introduction	141
7.2 Data Reduction	141
7.2.1 Analysis.....	145
7.3 Data Extraction.....	146
7.3.1 Analysis.....	150
Chapter 8 Distribution and Implementation.....	151
8.1 Introduction	151
8.2 Manufacturers.....	152
8.2.1 Advantages	153
8.2.2 Benefits to Manufacturers	155
8.2.3 Drawbacks.....	156
8.2.4 Configuration	157
8.3 WEEE Centres.....	158
8.3.1 Benefits	158
8.3.2 Drawbacks.....	159
8.3.3 Configuration	159
8.4 Large-scale Recycling and Refurbishing Centres	161
8.4.1 Benefits	161
8.4.2 Drawbacks.....	162

8.4.3 <i>Configuration</i>	163
8.5 Distributed independent take-back	164
8.5.1 <i>Advantages</i>	165
8.5.2 <i>Drawbacks</i>	165
8.5.3 <i>Configuration</i>	166
8.6 Conclusion.....	167
Chapter 9 Conclusions and Future Work.....	169
9.1 Contribution.....	169
9.2 Future Work.....	171
Appendices.....	175

LIST OF TABLES

Table 1: Distribution of duration values in duration blocks	84
Table 2: Signal sequences for PCD to PICC communication defined in ISO 14443 standard	91
Table 3: Information coding for PCD to PICC communications.....	92
Table 4: Signal sequences for PICC to PCD communication defined in ISO 14443 standard	92
Table 5: Information coding for PICC to PCD communications.....	93
Table 6: Synopsis of commands used in state machine	102
Table 7: LED signals for state machine	133
Table 8: Additional features in MIFARE Magic software.....	136

LIST OF FIGURES

Figure 1: Overview of data extraction system operation	73
Figure 2: PCI hardware interactions	74
Figure 3(a): Sample data showing range of temperature samples, and (b): retrieved trend data	79
Figure 4: Rainflow cycle counting, demonstrating various half-cycle terminations	81
Figure 5: Data bin structure in memory	82
Figure 6: Initial structure of data bins in memory	83
Figure 7: Data storage requirements: Rainflow Cycle Counting compared to Linear Storage.....	83
Figure 8: ISO 14443 definition for "pause" in communication	91
Figure 9: Possible bit sequences at output of microcontroller	97
Figure 10: Flow diagram outlining operation of transmission section of assembly code	98
Figure 11: Flow chart for receiving and decoding of data	101
Figure 12: AVR Studio 5 debugging interface	107
Figure 13: Layout of stimulus file for AVR Studio 4	109
Figure 14: PICC output sequences with delays highlighted for stimulus file generation	111
Figure 15: Example of input and output calculations for stimulus files	113
Figure 16: CRC generation and testing software	113
Figure 17: Sample of graphed output of log file from AVR Studio 4	114
Figure 18: Elektor Electronics MIFARE RFID reader	124

Figure 19(a): MIFARE UltraLight Interface, (b) MIFARE Terminal Interface.....	125
Figure 20(a): Six-pin header of AVR ISP; (b) ISP Programming circuit.....	126
Figure 21: Incoming “pause” compared with signal after capacitor and diode	129
Figure 22: Pause at output of comparator	130
Figure 23 :Receiving circuit including component values.....	130
Figure 24(a): Elektor description of REQA signal; (b): Signal measured at microcontroller input.....	131
Figure 25(a): Measured induced voltage from MIFARE Card & (b) from prototype ..	132
Figure 26(a): Complete RFID hardware solution diagram; (b) Prototype circuit.....	134
Figure 27: Mifare Magic interface, with additional options for LCM PICCs	135
Figure 28: Visualisation of optimal overhead in transmission of data	138
Figure 29: Interface between USB cable and ATMega16	139
Figure 30: Graph of cycle duration against cycle amplitude counts in desktop computer	142
Figure 31: Graph of cycle duration against cycle amplitude counts in laptop computer	142
Figure 32: Graph of cycle duration against cycle mean counts in desktop computer ..	143
Figure 33: Graph of cycle duration against cycle mean counts in laptop computer.....	143
Figure 34: Graph of cycle amplitude against cycle mean counts in desktop computer	144
Figure 35: Graph of cycle amplitude against cycle mean counts in laptop computer ..	144
Figure 36: Visualisation of practical data transmission timing.....	149

LIST OF APPENDICES

Appendix A RFID CODE SOLUTION	177
1. <i>Final application code for ATmega16</i>	177
2. <i>Previous code implementation</i>	190
Appendix B DATA REDUCTION CODE	201
1. <i>PC-based code</i>	201
2. <i>ATmega16 Code</i>	209
Appendix C MODIFICATIONS TO READER HARDWARE CODE	219
Appendix D MODIFICATION OF MIFARE MAGIC VISUAL BASIC SOFTWARE	227
Appendix E PUBLICATIONS	233
1. <i>Mechatronics, University of Limerick, 8th June 2008</i>	233
2. <i>International Symposium on Sustainable Systems and Technology, Arizona, May 2009</i>	241
3. <i>16th CIRP International Conference on Life Cycle Engineering, Cairo, 2009</i>	249
4. <i>International Symposium on Sustainable Systems and Technology, Washington, May 2010</i>	255
5. <i>CARE Innovation, Vienna, November 2010</i>	261

LIST OF ABBREVIATIONS

AM	Amplitude Modulation	PECI	Platform Environment Control Interface
ATQA	Answer to Request Type A	PHM	Prognostics and Health Management
CPU	Central Processing Unit	PICC	Proximity Integrated Circuit(s) Card
DfEoL	Design for End of Life	PoF	Physics of Failure
DfS	Design for Sustainability	REQA	Request Command, Type A
EoL	End of Life	RF	Radio Frequency
EoU	End of Use	RFID	Radio Frequency Identification
EPR	Extended Producer Responsibility	RoHS	Restriction of Hazardous Substances
ErP	Energy Related Products	SAK	Select Acknowledge
GWP	Global Warming Potential	SEL	Select
HDD	Hard Disk Drive	SSD	Solid State Drive
HF	High Frequency	TAAS	Technology as a Service
ICT	Information and Communications Technology	UHF	Ultra-High Frequency
LCA	Life-Cycle Assessment	UI	User Interface
LCM	Life Consumption Monitoring	UID	Unique Identification Number
LF	Low Frequency	USB	Universal Serial Bus
MTTF	Mean Time to Failure	WEEE	Waste Electronic and Electrical Equipment
NVB	Number of Valid Bits	WUP	Wake Up
PC	Personal Computer		
PCD	Proximity Coupling Device		

Chapter 1

INTRODUCTION

Through this work, the environmental, social and economic effects of information and communications technology (ICT) and in particular PCs are examined. Social and economic benefits which are enabled through the use of ICT infrastructure are manifest in modern society, and are far more apparent than the direct environmental impacts. Developed and developing societies alike rely on the continued progress of ICT for economic survival, and educational, medical and cultural benefits will continue to improve with on-going technological developments. ICT is touted as the key to escape from poverty in developing nations and underprivileged communities, and growth in penetration of PCs in developing nations now exceeds that of developed nations.

In this context, the suggestion of a reduction in reliance on ICT seems completely ludicrous for societies wishing to remain globally competitive; however the less obvious impacts of the increasing penetration of ICT infrastructure on the environment necessitate a serious examination of the usage patterns. Because of the significant impact of the manufacturing phase, and the potential for mismanagement of PCs during the disposal process to have negative environmental consequences, lifetime extension through reuse has been repeatedly suggested as an area in which PCs can reduce their overall environmental impact.

The development of widespread or formal reuse infrastructures, however, has been conspicuously absent. While manufacturers may be required, under various legislative and voluntary schemes, to ensure the correct handling of their products at the disposal stage, and while some of these schemes recognise the advantages of reuse in terms of waste reduction, it has proven difficult to efficiently demonstrate whether products are designed to be suitable for reuse.

1.1 Contribution

This work considers second hand computer markets with regard to their viability, potential benefits (compared to other end of life goods), and reasons for secondary markets' failure to develop organically. It then suggests a technological solution, which would enable analysis of viability for reuse of end of life PCs.

Various Radio Frequency (RF) technologies and protocols are examined with regard to the suitability for end-of-life evaluation of reuse potential of PCs with the goal of development of metrics of reusability based on prognostic health monitoring (PHM) techniques and principles. This is the first application of PHM technology in a consumer market with focus on reuse. A low-cost extension to existing Radio Frequency Identification protocols is proposed, allowing the serial chaining of data packages. This enables the extraction of data volumes orders of magnitude greater than those catered for under current protocols, allowing volumes of data associated with PHM technologies to be handled.

Data characterisation and reduction techniques are examined to ensure that a consistently low volume of data is stored throughout the lifetime of a PC using a binning structure for stress cycles rather than linear storage of data to memory. These

techniques use load characterisation to evaluate the overall impact of the thermal lifetime load.

1.2 Outline

In chapter 2 of this work, a comprehensive review of the literature is presented, with emphasis on the impact which PCs have on the environment at each phase in their life cycle. Methods currently in use in other industries for reliability monitoring and prediction are presented. A brief introduction to RFID technology is given, detailing the various types of technology which fall under the umbrella of RFID, their similarities and distinctions, and the characteristic performance of each. Finally, some research has examined RFID in the context of waste handling in the past, and this is examined here, as it provides a context for the development of the technology into the area proposed by this work

In chapter 3, examining the failure of a reuse market suggests that there is significant lack of confidence among consumers in used computers. Furthermore, any market which does develop is unlikely to be economically efficient, as adverse selection operates in markets such as this where the seller has private information about the quality of the available product, so lower quality systems will tend to naturally flood the market. In comparable markets in the past, signalling techniques have been used to enable sellers to access information relevant to the quality of the product, however in the case of PCs there is not currently a clear metric for the quality of a used PC, with the result that development of a signalling system would prove difficult under current conditions.

Prognostic health monitoring methods are used in some industries to estimate the likelihood of failure for products and structures. Chapter 4 presents the possibility

that, by applying similar techniques to PCs, the cumulative effect of the loads which a system undergoes could be estimated, allowing a suitably informed individual to gauge the probable remaining life expectancy of the system. As thermal loads have a significant effect on the reliability of electronic components, lifetime temperature information would provide a valuable input into any such calculation. Using the thermal sensors currently embedded in PC processors would be a relatively low-cost method of gaining access to this type of information; however its extraction at the disposal phase remains a significant barrier to its use for signalling purposes.

Chapter 4 lays out the solution proposed in this project; using Radio Frequency Identification (RFID) in combination with existing reliability monitoring technologies to provide a signalling methodology for PCs at end of life. The requirements for a system which uses environmental stresses to estimate continued reliability are laid out, and the data requirements of a system are considered. A method of load characterisation called rain flow cycle counting is introduced, which limits the total amount of data which would be extracted over an RFID interface.

In chapter 5 and chapter 6, the design of the reader circuitry and the requirements for modification of reader operation and user interface software are examined from a hardware and software perspective respectively. The tag developed in this way conforms to all requisite RFID protocols, while allowing the transfer of a far greater amount of data than traditional RFID tags. The design process for the tag is explored, and its capacity to handle the volume of data required is examined.

In chapter 7, results from the design and testing process are presented, and the efficacy of the prototype for the intended purpose is examined, and suggestions for increasing its utility in this regard are presented. Some results of rain flow counting

analysis of processors are presented, and an overview of the conclusions which can reasonably be drawn from the data is laid out.

Chapter 8 presents an overview of the scenarios in which the technology could be applied in the context of existing legislative structures and reverse logistics infrastructure. A variety of options are presented, any of which could operate independently or in parallel with others. Technological and logistical requirements for each solution are considered, and the advantages and drawbacks of the different suggestions are considered from an environmental and practical perspective.

Finally, in chapter 9 the suitability of the selected technology for the purpose outlined is reassessed in the context of the results obtained from testing of the prototype device and areas in which further research is required, both for the successful implementation of a system of this type and for the handling of the likely profile of electronic waste in the near future, are defined.

Chapter 2

LITERATURE REVIEW

2.1 Environmental Impact

The environmental impact of the PC industry is a relevant and growing concern. Although the penetration rate of computers continues to increase in both developed and developing nations, the underlying trend in developed nations is in fact away from the desktop PC and towards more mobile solutions, such as laptops, smart phones and tablets. As a result, the sales of desktop systems has somewhat levelled out over the past number of years. In developing nations, however, the penetration rate of desktop systems is still increasing rapidly [1, 2].

There are numerous studies examining the lifetime environmental impact of electronics, and several focusing on the specific impact of PCs [3, 4]. These studies show differing results for the quantified impacts of PCs for each life cycle phase, depending on the assumptions made about use patterns, length of life, and the factors which are taken into consideration when estimating total impact.

Life-Cycle Assessment (LCA) is a process used to attempt to calculate the total environmental impact of a product or process throughout its life cycle. This is a difficult process, as it requires quantifying the environmental effects of a wide variety of activities, from the initial design of the product, to materials extraction, transport, production and assembly in the manufacturing phase, to the energy required to operate in the use phase, to the disassembly and recycling process. As a result of the

complexity, there are several types of LCA, which may yield different results for the analysis of similar products [5, 6].

2.1.1 Manufacturing

Studies into the environmental impact of the manufacture of computers reveal that this phase in the life cycle is hugely significant in terms of the overall environmental impact. While different studies use different metrics to quantify the total environmental impact, it is generally considered to be either the phase with the greatest impact [4, 6], or second greatest, after only the use phase [3, 5, 7]. Regardless of the precise proportions, the production phase of the life of PCs contributes disproportionately to their impacts compared to other consumer products [8]. Further analyses shows that the desktop portion of the computer outweighs the peripherals such as mouse, keyboard and screen, in terms of environmental impact in this phase, and that of the components of a PC case itself, the motherboard represents over half of the environmental impact [5]. This is based on the energy mix of a fabrication plant in China, which has a higher impact than the global average, owing to the lower efficiency of power generation and transmission. Although there are silicon fabrication plants in other countries, China currently contribute approximately 20% of global silicon production, and based on China's development plans, it is thought that over half of the world's global warming potential (GWP) from silicon production will be caused by Chinese production by 2020 [7].

The reason cited for the high environmental impact of motherboards is the chemicals used in the fabrication of silicon chips, as the process requires the use of a large amount of high-purity chemicals and clean water, with the water use at a typical silicon wafer production plant estimated to be over 2 million gallons daily. Further, outputs from the silicon development process may include harmful emissions to air,

water and ground systems [9]. There are some disparities observed in the estimates of computers' impact, which have been attributed to the inaccuracy or lack of availability of accurate or consistent data on chemical use and emissions [7, 9]. Because of the high purity of materials required, and in some cases the rare-earth metals which must be mined in order to produce components, the manufacture of electronic products is a highly energy-intensive process. It has been commented that the production of electronic microchips requires a disproportionate amount of materials and results in the emission of significantly greater volumes of harmful materials than other comparable products, compared to their weight [9].

Looking forward, as the penetration rate of computers increases, the impacts of the increased volume of semiconductor manufacture is somewhat mitigated by efficiency improvements in the process for the manufacture of the components. Manufacturing energy use remains relatively static across the development of new technologies [7, 10]. While the energy use per unit of computational power and per kHz clock speed has decreased exponentially, the effect has been completely balanced by the rapid progress in increasing both the number of transistors and the clock speed of processors. The result is that there is no clear change in energy requirements for semiconductor manufacturing over the last 17 years [10].

2.1.2 Use Phase

There is significant deviation in the estimates for both the use phase environmental impact of PCs and their life time. Estimates range from 2.9 years [6] to over 6 years [3], with a range of values in between, dependent on the country which is considered [1, 5, 7, 9]. This is attributed to the differing length of use phases in different countries, and to the differing definitions of the term "use phase" [11, 12]. It has been pointed out that a significant proportion of used PCs are stored for a period of time

between their retirement and disposal [4], resulting in skews in the life cycle estimation methods [11, 13]. A more accurate method of estimating life cycle, which accounts for the shorter use phases in some studies, has been suggested to be considering the rate at which computers are replaced [6]. This more accurately estimates the number of computers in active use at any time, as many computers are stored for a period of time after retirement and before disposal. While this period of storage would be considered to be in the use phase according to some studies, there is no associated energy consumption and environmental impact.

The use of computers now represents a significant proportion of the energy demand of modern society. A study from Japan estimates that home PCs alone account for 0.64% of the energy demand in that country [13], while a prominent study of office ITD equipment in the US in 2002 attributed 3% of the country's power consumption to office ICT equipment [14]. Other estimates put the energy use of ICT devices in homes and offices at 6% of the total US energy use [15]. As ICT becomes more pervasive, these figures can be expected to increase.

Because the majority of the impact of this phase is centred on electricity use [1], the energy mix of the country in which the computer is being used will have a significant scaling effect on the overall impact of this phase of the lifecycle. The majority of impacts in this regard are in the form of carbon emissions and radioactive waste from power generation [16]. As developed countries make advances in sustainable energy [17, 18], the environmental impact of energy use in these countries decreases proportionately. While this somewhat mitigates the effects of this phase, energy efficient computers should be used in combination with green energy supplies and efficient power management settings to minimise their environmental impact [4]. New regulations have led manufacturers to implement changes reducing power

consumption of computers' idle modes in recent years [3, 19]; as well as using energy efficient default power settings, however wide-scale consumer behaviour change in this area could heavily reduce the use phase impacts of PCs, and evidence suggest significant scope for improvement in this area [3, 14, 20].

Recent studies point towards a gradual decrease in the length of the use phase for successive generations of computers in the US [11], which decreases the significance of the use phase as a proportion of the total life cycle impact of an individual computer. If these computers are being replaced on a like-for-like basis, however, the overall environmental impact is made worse, as the disposal of old computers will coincide with the manufacturing of new ones, and manufacturing and disposal must happen more often in this scenario [21].

In many industries, the idea of lifetime optimisation is cited as being preferable to lifetime extension[22]. This is related to the fact that, with improvements in environmental performance of products during the use phase, the environmental load of manufacture of a new product might be counterbalanced by the lower impact of the product in operation for a use phase longer than the threshold duration. For this concept to work, however, it is necessary for the operating environmental impact to be significantly lower in the use phase. This is generally caused by more efficient energy use, and is particularly noticeable in the area of white goods, such as refrigerators, where energy use improvements regularly make replacement an environmentally preferable option to lifetime extension.

In PCs, however, the overall energy consumed by a typical processor during normal operation has increased over the same time [10]. The result is that in general, lifetime extension is an environmentally preferable option to system replacement, as

extending the lifetime of a current-generation PC would have lower environmental effects than the equivalent use phase of a new PC, even excluding manufacturing and disposal impacts. The exception to this is when the system is being replaced with a breakthrough technology, which performs the function of the previous system, but with significantly lower use phase energy requirements. An example of this is replacement of a PC with a laptop or tablet, or replacement of energy-intensive Cathode-ray tube (CRT) monitors with more efficient liquid crystal displays (LCDs) [21]. In these cases, replacement may be the environmentally benign option, although this remains heavily dependent on the use-phase characteristics of the products in question.

The evolution of computing from product to service-based provision and the corresponding increase in web-based and cloud services has moved a significant portion of the impact of computing out of the home and office to server farms and data centres. This aspect of the impact of computing is often neglected in the computation of life-cycle energy, however it should be considered when calculating the energy of the use phase, as the energy consumption associated with data centres is estimated to be half of all IT-related electrical impacts [23], at 29 GW in 2008, and the rate of growth of energy consumption for data centres is higher even than the worldwide growth in power consumption of PCs, at a predicted 12% annually [24].

2.1.3 End of Life

The end of life (EoL) is an area to which considerable attention has been given over the last number of years. Trends show a steady increase in the number of both desktop and laptop PCs which are reaching EoL [1, 25], and considering the continued sales growth in both portable and desktop markets, this is a trend which can be expected to continue for the foreseeable future. While recycling has become standard in developed nations, and this is regarded as a preferable result than landfill, further

improvements remain to be made, as laid out in the EU waste framework directive [26]. This mandates that governments should take measures to first prevent waste entirely and then prepare for reuse before recycling goods.

As increasing numbers of industries are being encouraged to increase their environmental performance, an increasing number of examples of “win-win” situations are emerging, where the environmentally beneficial method of handling materials at end of life is also beneficial economically for the business, through recovery of materials [27].

2.1.3.1 Recycling

The optimal method for end of life recycling for a product is heavily dependent on the earlier life cycle phases. For example, the processes of design and manufacturing will have an effect on the optimal end of life disassembly process. In acknowledgement of this fact, there has been an increase in the number of manufacturers actively considering design for sustainability (DfS) and design for end-of-life (DfEoL) [25, 28] in the design of their products. Legislation, such as the restriction of hazardous substances (RoHS) rules in the EU to a certain extent mandates DfS design practices, while the Waste Electrical and Electronic Equipment (WEEE) directive in the EU and similar legislation elsewhere provides an economic incentive for DfEoL [27], as manufacturers are held financially responsible for the handling of electronic waste. This is an example of extended producer responsibility (EPR), a system developed in the early 1990s to ensure that the responsibilities for EoL management remain with those responsible for placing them on the market [29].

Through correct handling of EoL PCs, the impact of the recycling process can be minimised. Under existing WEEE legislation, suppliers of electronic equipment are

currently responsible for the end of life handling of materials, and ensuring that the appropriate recycling processes are carried out [26]. This is implemented on a Europe-wide scale, on the basis of WEEE targets. The initial legislation in Ireland described a target of 4 kg of waste recovery per household, however following the unprecedented success of the initiative, the goals of the WEEE directive are due to be revised, with targets expressed as a proportion of the weight of electronic goods sold. A computer typically contains hazardous chemicals, so disassembly is required in order to ensure that any environmental risks are removed and dealt with correctly [30]. By extracting any valuable materials from equipment, some of the cost of the process may also be recouped, and if carried out correctly, recycling may have a net environmental benefit as recovered materials may be reused in new products [5]. Recent increases in the costs of raw materials have increased the viability of material recovery from used systems, and there is a developing industry centred on the retrieval of high-quality, high value metals from used electronics [31].

2.1.3.2 Export

Used computers have an embedded value, owing to the amount of precious metal present in the components, including gold, other rare metals and copper [15]. Despite the industries in developed nations as mentioned above, in many cases the recovery of these materials is not considered profitable in developed countries owing to cost of labour and to legislation governing processes for materials recovery. In developing nations with lower standards of environmental legislation, however, these materials can profitably be recovered from used systems [32, 33]. What is regarded as electronic waste in developed nations has significant value in the developing world, and several nations, including Nigeria, Peru, China and India have developed informal industries in the recovery of valuable materials from EoL electronics [32, 34-36].

The unregulated recovery of materials from used computers can have a negative environmental impact, however. In countries where the recovery of materials has become a backyard industry it has been shown that the environmental effects of uncontrolled recycling include lead contamination of drinking water supplies [30], release of harmful dioxins and furans from burning of plastic-coated copper wire [35] and hydrocarbons from burning of printed circuit boards [34].

Owing to the toxic components of electronics, their international transportation is illegal under the Basel Convention on the control of transboundary movements of hazardous wastes [37]. As a result, it is necessary for exporters to assert that electronics are functional before transport. However, there remains a market in export of waste from developed nations to developing ones. In order to circumvent legislation on the export of waste, shipments of EoL electronics are often miscategorised as functional prior to export, and sent with the stated goal of reuse. While the literature disagrees on what proportion of these products are reusable [35, 38], the export of systems which are not viable for reuse has been identified as one of the major problems which developed nations must tackle in the 21st century [39].

While this industry is undesirable from an environmental perspective, there is obvious economic benefit to disadvantaged areas in the retrieval of the materials in this way [32, 35]. As a result several first-world companies have recently developed the recycling facilities in developing nations, to improve the recycling methods in use, and ensure fair treatment of workers. In this way, the established industry is not affected, while environmental and social improvements are realised [40].

2.1.3.3 Reuse

Although reuse is suggested by several authors as the preferable method for decreasing the environmental impact of ICT [21], a functional market for the reuse of EoL computers has not emerged in the developed world, although sites such as eBay do provide a forum for the informal trading in used PCs, resulting in some growth in the area [13, 32]. The lack of interest in second-hand goods is in part attributed to the decreasing costs of new PCs, although there are a variety of factors which impact the behaviour of customers in this market. Research was carried out in Ireland regarding the barriers to the organic formation of this market, and the results indicated a lack of consumer confidence in the reliability of used computers restricting the formation of any secondary market [22].

While the purchase of a new computer is accompanied by a guarantee of quality, and sometimes free technical support, no such assurance may be provided with a used system. Because of the perceived technical complexity of computers, many customers are not willing to risk purchasing a used model, even at considerably lower cost, preferring the security associated with a guarantee [13].

In a market for used computers, it is therefore difficult for the average person to distinguish the higher quality systems from those of lesser value. This uncertainty decreases the price that the market will support, resulting in many people with high-quality used computers being incapable of getting value for money for their products, and therefore choosing not to sell them. As a direct result the market becomes saturated with lower quality systems, in a pattern economists identify as adverse selection [41, 42]. Over time, consumer confidence in the market falls further, resulting in a downward spiral in prices for used products. This contributes to the closet effect, which

is the title given to the wide-scale storage of computers, before their eventual disposal, constantly losing value in the meantime owing to on-going technology changes [13, 22].

As the effects of adverse selection affect the market, driving down the prices buyers would be willing to pay for used computers, the endowment effect has the opposite effect on sellers. This inflates the value the user places on an item, purely because it is theirs [43]. As a result, even given information symmetry between buyers and sellers regarding the quality of the used system, an agreement on price may not be emergent, as the owner of the computer is likely to attach some additional value to the system. Additionally, value for the owner of a computer becomes embedded over time, as the functionality of the system is highly customisable to a user's needs. As a computer is customised to suit one owner, its value to that person increases, while remaining static for potential buyers.

2.2 Reliability Monitoring

The process of predicting reliability for critical systems was traditionally performed using fixed formulas and metrics outlined in handbooks such as MIL-HDBK-217 [44]. This handbook provides formulae detailing the number of hours for which a component can be expected to operate, given its physical parameters [44, 45]. These types of reliability prediction methodologies were used by designers to estimate total life-cycle costs of designs, and to predict the reliability of system-level designs incorporating a range of components. Prediction models also allow designers to analyse trade-offs in reliability between different design methodologies, and to predict the cost and rate of failures [8].

The current opinion is that the handbook method does not provide an accurate view of the failure methods for individual products, however, as the environmental

loads to which each product is subject are not considered in the calculations, rather a constant failure rate of components throughout the lifetime of a system is assumed [46]. In high-value products, where maintenance schedules may be determined by these reliability prediction algorithms, the cost of incorrectly estimating the reliability of a product may be scheduled maintenance performed where it was not necessary, overstocking of spare parts, or the failure of a product owing to its environmental load being greater than that which was catered for in its reliability prediction algorithms [45].

In the mid-20th century, failure rates of electronics were modelled as a bath-tub, with high initial failure rates owing to early failure of parts and discovery of manufacturing faults. This led into a period dominated by a low but constant rate of failures owing to external forces, such as environmental stress, or user mishandling. Through development of more robust assemblies these failures can be minimised, but eliminating them entirely is difficult. The final stage in the life of electronics is the wear-out phase, when the accumulated effects of lifetime stresses cause the failure rate to increase again[8]. It has been commented that overall, the retirement of modern electronic devices tends to occur before the wear-out phase, meaning that the characteristic of increasing failures towards the end of life is no longer a factor in the life cycle [46].

Among the criticisms of the traditional reliability prediction methodologies is the fact that they, by necessity, estimate the conditions under which components will be operating, and based on this estimation, provide a prediction of the mean time to failure (MTTF). Owing to the nature of the estimates which are used, it is difficult to account for the effects of non-linear or unpredictable factors, such as temperature gradients, temperature cycles and mechanical shocks [45]. One alternative to these empirical models for failure is a physics-of-failure (PoF) model. Following criticism of empirical

models for reliability prediction in the late 1980s and 1990s [46], a more deterministic approach was proposed, which takes account of the physical processes which govern failure mechanisms. These methods attempt to determine when a particular failure mechanism will occur for a particular component using information about the specific types of failure to which the component may be subject [8]. This provides a more complex analysis of failure than empirical methods, allowing the identification of the probable causes of failure in a component.

2.3 Life Consumption Monitoring

With the goal of preventing unnecessary servicing costs, and of ensuring that catastrophic failures are averted by timely maintenance, a range of methods for the accurate prediction of the reliability of a product have been developed. These are referred to by the umbrella term of Prognostics and Health Management (PHM), and aim to quantify the environmental loads to which a product is subject with the goal of predicting with some degree of accuracy the likelihood of failure of an assembly. There are several methods by which this is achieved [47].

- a) Devices which have a lower tolerance of environmental strains can be used. These will fail in advance of the main product, providing advance warning of system failure.
- b) The parameters of product operation which provide early warning of failure can be monitored.
- c) The factors which contribute to the degradation of products can be monitored. Through knowledge of the combined effect of these factors, the cumulative impact can be estimated.

Although this form of monitoring is commonly used in very high-value, reliability-critical structures, such as bridges and buildings, its implementation in electronics is still relatively uncommon. A noticeable exception, however, is the prevention of catastrophic failure in some processors by means of reduced operation followed by forced shutdown when temperature exceeds set limits. This is a basic example of type b) above, where the temperature is the indicator of imminent failure, triggering the remedial action.

Within PHM, life consumption monitoring (LCM) methodologies are techniques used to attempt to estimate the effects of physical strains on the reliability of a product during the course of its use [48]. This can be seen as an application of the theoretical framework as defined in PoF models to attempt to predict failures at an individual system level. By using measurements from sensors located within a product, these methodologies try to predict the likelihood of the occurrence of failure based on the PoF models [49, 50]. In this way, it is theoretically possible to extrapolate from life-cycle data an approximation of the expected remaining life of a product. One of the primary advantages of LCM is seen to be the potential to allow for preventative maintenance on products, by means of PHM where, by predicting failure before it happens, the costly consequences of that failure can be avoided by appropriate intervention [51]. Measurements consistently outside of a certain range can be a symptom of a fault, so the information returned in this way can allow for non-critical faults to be detected and remedied before critical damage results.

The potential for using LCM techniques to accurately predict the remaining potential of electronics has been discussed in studies by the European Union. In these studies the goal was to provide a prognosis of the long-term potential of a unit under test, and so to accurately predict its probable lifespan. While these studies were

valuable, the Environmental Life Cycle Information Management and Acquisition for consumer products (ELIMA) was focused on the practicalities mainly from a manufacturer's point of view and did not address the potential for the use of life cycle data for the purpose of encouraging an independent market for reuse [52]. The study provided life cycle information which was useful in its own right; however the implementation of the life cycle monitoring techniques used in the ELIMA project for monitoring of consumer products would not be technically or economically feasible. This was attributed to the high cost of the technology required to perform the analysis.

2.4 Factors Impacting Reliability

Although PCs are technically complex systems, and their failures can occur for a large variety of reasons, the majority of contributing effects can be broken down into a relatively small number of factors. By monitoring these effects throughout the lifetime of a PC it may be possible, using LCM methodologies, to provide a reasonably accurate assessment of continued reliability.

2.4.1 Operating Time

As a very simple to observe and record metric, the amount of hours for which a computer is operating is essential to determining the value of a system. During the first hours of use, operation of a computer is actually most likely to result in failure, as a certain proportion of electronic components will normally cease working during this 'burn-in' time. As this is normally included in the warranty of the product, the computers are fixed at no cost to the purchasers. It is worth noting that there would be no comparable initial period during the second life of a PC, as the components will have already passed through the burn-in phase.

Operating time is not, on its own, considered to have a significant effect on the reliability of PCs. As outlined above, mechanical failure for electronics has become less common within the useful life of those electronics [53], with the actual failure rate instead tending to decrease over time. As such, operating time may be taken into account as a factor in the reliability of electronics, but only when considered in the context of the conditions under which the electronics were operating, and the effects that these conditions may have on reliability.

2.4.2 Temperature

The temperature at which a computer operates is widely considered to be one of the principle causes and indicators of failure. Depending on the environment in which a PC is located, the type of applications for which it is used and the cooling solution in use, the operating temperature can vary widely. This is an on-going concern owing to the fact that electronic components are constantly becoming smaller and more powerful. As the capabilities of these components increase, so does the amount of heat which needs to be dissipated, and the challenge of dissipating a greater amount of heat from a smaller electronic package must be repeated for each generation of electronic products [54].

One of the primary effects of temperature gradients is electromigration, whereby the atoms of the conducting material will, over a long period, move in the same direction as the general motion of the electrons causing the current flow. Over time, the general migration of atoms from a point will weaken the electrical conductivity at that point, eventually resulting in increased resistance or even open circuits. Concurrently, the migration of the conductor towards one point will lead to an accumulation of conductive atoms, potentially resulting in unintended short-circuits with nearby components [59].

In the past, the temperature dependence of the reliability of electronics has been expressed as a halving of reliability for every 10° Celsius increase in temperature. Although this statistic is no longer considered valid [55, 56], it is beyond doubt that the environmental factor with the greatest impact on PCs is the temperature at which they operate [57].

This is of particular interest during the use phase owing to the adjustments which PC owners can make in order to control the temperature of their product [58]. These include such simple actions as ensuring the vents on the chassis are not covered or blocked, that the system is not located somewhere that regularly encounters extremes in temperature, such as near a heater. By taking action to reduce the average temperature of their PC, owners can extend the reliable lifetime of the system, preserving its value in a situation where it is to be sold in a secondary market.

2.4.3 Temperature Cycles

While historically, overall temperature was considered to be the principle cause of thermal degradation in electronics [60], it is now known that the majority of thermal effects on reliability are caused by cycles in temperature [56, 61]. These may be observed as the computational load on a processor varies throughout a day, as the temperature is normally dependent on the total computational load on the PC. Other, longer cycles may be observed as a PC is turned off, or goes into standby at night, and is active during the day for long periods.

The effects of temperature changes on mechanical components are primarily related to expansion and contraction of materials. Where two different materials are in contact, the different rates of expansion and contraction with changing temperature can

result in microscopic cracks [62]. These cracks are most common at solder joints in between components. As the component expands and contracts with changing temperature, small stresses are placed on the soldered connections. These stresses are greatest on the corner connections of components; as these are the points at which maximum stress cycles occur. Over time the transmission of signal can be interrupted if these cracks occur at solder joints or on electrical paths. This sort of signal interruption can be fatal to a system, and can also be almost impossible to diagnose and repair.

Many technologically adept people, particularly gamers, will monitor the temperature of their own PCs, as a low average temperature is considered a symptom of a PC which is not being used beyond its limits. A lot of emphasis in the high-end of gaming computers is in the area of cooling, and “over-clockers”, people who tweak their processors to operate outside of the manufacturers’ recommended specifications, will often invest a large portion of the cost of a PC in ensuring that an adequate cooling solution is in place. One of the major risks associated with over-clocking a processor lies in the potential for overheating resulting in critical thermal damage. As such, there are a variety of software solutions available to aid in the monitoring of the temperature of processors [63, 64]. These solutions offer the capability of thermal monitoring from within an operating system through a user interface; however the sources of the thermal data are located within the hardware of the system.

As temperature has such an effect on the reliability of computers, it is already monitored as standard by all processors, and also by some other components, such as graphics processing units. At present, the monitoring of temperature by the processor is, in the case of Intel processors, at least, primarily for the prevention of critical failure owing to instantaneous excessive temperature. If the processor exceeds a certain predefined temperature, T_{junction} , the processor is ‘choked’, reducing the frequency at

which it operates and allowing the system to cool down [65]. If this does not affect the temperature increases, and the processor exceeds a second, higher threshold, the computer will shut down automatically, preventing critical thermal damage to components.

In older processors, the temperature measurements were performed using a thermistor, which would produce a voltage proportional to the temperature of the junction between two metals. There were issues with this approach, as each thermistor would react differently, and needed to be calibrated to the junction temperature of the processor in use.

In more recent generations of processors, Intel has updated its approach through the use of the Platform Environment Control Interface (PECI), a system for temperature monitoring which makes use of digital thermal sensors located at various points in the cores of the processors [65, 66]. The interface itself is a single wire protocol, and components may be added to a motherboard which can be configured to return one of a variety of measurements. It's possible, by setting up the interface through a specific temperature controller, to control the sample rate which is used to measure temperature, and also the range of values returned, to averages over a time period, or the maximum temperature encountered in up to eight thermal sensors over up to four central processing units (CPUs) [67].

2.4.4 Humidity

Humidity can have multiple effects on a computer. If consistently operating in a humid environment, components can be subject to corrosion throughout their lifetime, reducing electrical conductivity, and gradually degrading operation.

In extreme cases, humidity could cause short circuits in components, resulting in catastrophic failure and loss of data. In normal operation in most home or office environments PCs should not be subject to such humidity, but it is worth considering that some laptops and other mobile computers are operated in a very wide variety of environments, and may be subject to stresses which do not normally affect static computers. Laptops and tablets are more likely to suffer spills than desktops, and are also more likely to suffer critical damage, as the chassis of a desktop is often out of the range of any accidental damage from spills, whereas owing to the compact nature of laptops, their functional systems are close together, and at greater risk from spills. This has been acknowledged by some notebook manufacturers through the inclusion of a drain in the keyboard, to harmlessly carry water away from the sensitive components of the computer [68].

2.4.5 Vibrations and Shocks

Where mechanical components such as fans, optical drives and HDDs are in use, a PC will be subject to a certain amount of vibration. While this shouldn't have much appreciable effect on the majority of well-constructed systems, severe vibrations, particularly when combined with temperature cycles, can accelerate the degradation of a system [49, 69]. Looking forward, with the general trend away from mechanical components such as optical drives and mechanical hard disk drives, internally generated vibration is likely to become less of a factor in determination of the reuse value of a system. However with trends towards portable computing, systems are more likely to be operating in environments where external vibrations may be a factor, such as cars, so the impact of this load may become more central to reliability.

2.5 Radio Frequency Identification

RFID was first invented in 1948, and adoption of the technology commercially began as far back as the 1970s, however in the last number of years it has greatly increased in popularity, and been adopted for a variety of applications across a wide range of industries. This is attributed to the decreasing cost of the technologies which make it possible. RFID has passed a tipping point, which in many cases makes its benefits outweigh its expense when compared to more traditional identification technologies such as bar codes [70].

2.5.1 Applications and Technologies

There are several varieties of RFID, depending on the application in question. They are all characterised by several common functions. All RFID communications take place between a “reader” which establishes communications, and a “tag”, which responds with its embedded information. Communications are in the form of electromagnetic waves, propagated between the reader and the tag at a specified frequency [71]. Information is encoded in this signal in a variety of ways, depending on the exact type of RFID standard in use. Both the reader and tag must be capable of both sending and receiving a signal, and of some degree of coding and decoding of signals. In the majority of applications the tag is pre-programmed with a set code, which identifies it to the reader [70]. The majority of RFID applications are for asset tracking or personal identification purposes, as the tags can be embedded in ID cards, or attached easily to products for tracking or stock management.

Low-end RFID technologies are primarily aimed at the functions of asset tracking and security. In these applications, it is not necessary for a reader to be capable of extracting large amounts of data from a tag, or of changing the data stored on the tag.

For security applications, it is often sufficient for the reader to detect simply the presence of any tag within the field. Readers of this type are used as security “gates” at entrances to stores, which activate alarms if the tags on goods have not been deactivated at time of purchase. Generally, Electronic Article Surveillance (EAS) tags of this sort are only required to return a bit if they are activated by a reader. They may also be deactivated entirely at point of sale [72].

More complex, but still low-end RFID technologies are increasing in popularity as a replacement for bar-codes. In these applications, each tag represents a unique serial number, which can be read at a point of sale, or for asset tracking. While within the interrogation field of a reader, a tag of this type will continuously relay its serial number. No advanced features such as two-way communication are necessary for these types of applications.

Further advanced RFID systems feature some sort of memory device, and are capable of processing commands received from a reader, including reading selective memory, writing new data to memory, and dealing with conditions where more than one transponder is located in the field of the reader [72]. There is also the potential for inclusion of advanced features such as authentication between reader and transponder. Authentication, security and privacy have become central topics in the debate around RFID, particularly since its inclusion in passport technology in many countries to store biometric data. Privacy advocates suggest that applications such as this are potentially dangerous, owing to the possibility of private information being accessed remotely without the owner’s knowledge [73].

2.5.2 Principles of Operation

There are two fundamental principles on which all RFID communications are based. These are near-field and far-field communications. Which principle is in use depends on the frequency of the RFID device in question; near-field principles work best at lower frequencies, while far-field operates typically at frequencies greater than 100 MHz. Near-field coupling is based on electromagnetic induction. The reader transmits power in the form of electromagnetic waves. A tag can detect these waves, rectify the signal and pass it into a capacitor in order to store enough energy for operation. In order to respond to the reader, the tag can vary the load on its own coil, changing the amount of current being drawn. The reader detects this current variation in its own antenna, so by precisely changing the load on the tag's antenna, information can be encoded on the signal which can then be decoded by the reader.

As frequency increases, the range at which near-field communications will work decreases, approximated by $c/2\pi f$, where c is the speed of light and f is the frequency of operation [72]. In order to operate at longer distances at higher frequencies, far-field communications are necessary. As inductive coupling does not operate well at higher frequencies, far-field tags operate using the principle of back-scattering. This is achieved by varying the impedance of the tag's antenna in response to communications from the reader. When the antenna is tuned to the frequency of the reader signal, it will absorb most of the incoming electromagnetic waves. By changing the impedance, the tag can be detuned from the reader, causing the reflection of this signal. Using a receiver on the reader, the amount of the signal which is being absorbed or reflected can be detected, so by controlling the impedance of the tag antenna data can be embedded in the reflected reader waves. Tag antennas which operate in this manner use a dipole

antenna, and their design and manufacture is complicated owing to the precision of the specifications required [72, 74].

Some RFID tags, called passive tags, have all of the energy for the entire communication process provided by the reader, and the tag simply modulates the incoming signal using the energy provided by that signal to respond to the reader. “Semi-passive” tags contain a battery, allowing transmission of greater volumes of data greater distances; however the battery is never used except for the purposes of transmission. Active tags constantly use power, and are normally connected to a battery or some type of structural power source, and can be used for continuous monitoring, of environmental conditions, for example.

As well as the active, semi-passive and passive types of RFID technology mentioned, the technology operates on a selection of frequencies, each of which offers its own advantages and drawbacks. There are some fundamental differences in the operating principles of these operating frequencies, both in the communications processes and in the physical methods for propagation of data. These are discussed below.

2.5.3 Low Frequency

Low Frequency (LF) RFID operates in the frequency range of less than 135 kHz. In general, lower frequencies of signal exhibit less absorption as they pass through non-conductive materials. Thus, LF has the advantage of exhibiting superior performance in the range of water or metals, but offers a low maximum read range, typically quoted as up to 33 centimetres, and a low data rate [71], making it unsuitable for the transmission of a large volume of data.

The exact data rate achievable depends on the standard in use, while range is dictated by the amount of energy provided by a reader antenna, the angle between the reader and the tag, and the tag, and the shape and size of the tag antenna itself. Many applications of RFID require a low range tag to prevent the unauthorised access to information stored on the tag. Low Frequency technology is also generally the most inexpensive to implement.

2.5.4 High Frequency

High Frequency (HF) RFID normally operates at 13.56 MHz, and offers a balance of performance and range, exhibiting some capacity for operating in the area of non-metallic obstructions, while offering longer read range and a higher data rate than LF [71]. High-Frequency communications operate using near-field communications, and so do not have the range offered by higher frequencies utilising far-field coupling; however the design of the antenna is considerably simpler.

The higher end of RFID applications, requiring two-way communication and multiple tag handling capabilities, are normally operated in the HF range. A large variety of applications are in common use which take advantage of high frequency RFID, as it offers a good balance between the inexpensive lower frequencies and the greater functionality which is offered by higher frequencies.

2.5.5 Ultra-High Frequency

Ultra-High Frequency (UHF) technologies operate using far-field technology, and electromagnetic back-scatter. Therefore they can achieve higher range than the lower frequencies, typically up to 3m for passive tags, and 15m with battery supported tags [74]. UHF technologies are also typically more resistant to electromagnetic interference than lower frequencies, and are suitable for operation in environments with

a large amount of mechanical equipment. Design of far-field coupling devices is more complex than near-field technologies, and communication is intolerant of barriers in the path of the signal between reader and tag.

2.5.6 Microwave

The final type of RFID technology is in the microwave band, which is any technology transmitting in a frequency range of greater than 2.4GHz. Microwave frequencies share many of the same drawbacks as the UHF technologies, including even higher cost and intolerance to barriers in the path of communications. For these reasons, fewer applications are developed using microwave based RFID than the other frequency ranges.

Over the last decade, decreasing costs of technology and an increased interest in the applications of RFID has sparked a large upsurge in innovation in the area. As technology advances, and more complex applications become more power-efficient, the number of functions served by RFID are increasing [70, 75]. An increasing number of applications are taking advantage of the decreased cost of development in the high and ultra-high frequencies of RFID, and the improvements in range which they can offer [74].

2.6 Smart Tags

The advantages of identification codes or tags on waste have been considered at some length. At present, simple numerical codes are used on packaging in the United States, which identify the types of material present, and can be used by recyclers to identify the correct recycling process. These codes are also used by consumers to ascertain whether specific types of packaging are handled by local recycling facilities. The use of identifying codes in conjunction with user products such as PCs can offer

similar advantages. At EoL, the use of labels which uniquely identify products can allow the optimal handling process for a product to be selected and carried out [76].

Specifically as product codes relate to the recycling of complex items such as computers, some research has suggested the use of globally unique codes stored on RFID tags or bar codes [77, 78], which would, when cross-referenced with a database of such codes, identify the exact composition of each product, along with the manufacturers' recycling guidelines. This would allow recycling centres to identify the composition of a complex product, such as a PC, quickly, and the subsequent disassembly process could then be optimised for the recovery of valuable materials [78]. This ties in with the concept of EPR, as manufacturers would not only be capable of designing their products for ease of disassembly, but also of specifying the recommended method for disassembly of each product, and of communicating this with recyclers. An additional benefit of this approach for the identification of each unique product is that the code would allow the tracking of a product from manufacturing, through point of sale, to disposal [76]. For products where maintenance is required, the codes and database could also be used to keep track of maintenance performed, and the state of a product throughout the life-cycle. This end-to-end approach would allow for tracking products throughout their life cycle, providing manufacturers with invaluable information about the ways in which their products are used.

The logical extension of this type of identification system is the smart-tagging system, in which the information relating to the state of the item is stored directly on the tag itself, allowing its retrieval at any point in the life-cycle of a product. As well as the advantages offered by product-level tagging above, this type of tagging would offer the recyclers information unique to the specific item in question, such as the conditions in which it has been operating, or the maintenance schedule throughout the lifecycle,

without the need to refer to an external database [79]. For this purpose, more advanced technologies than barcodes are required, as the information communicated by a product must update to remain relevant to the current state of the product. RFID has frequently been suggested as a technology which would suit this purpose, owing to its decreasing cost and relatively simple implementation [80].

2.7 Conclusion

At present the prevalence of ICT is causing a considerable burden to the environment, and this burden can be expected to increase as an increasing number of modern technologies continuously reach EoL. Owing to the disproportionately high environmental impact of the manufacturing phase of PCs, lifetime extension through reuse has been identified as a highly desirable method of reducing the overall environmental burden of the industry. A reuse market has failed to develop organically in developed nations, however, and this has been attributed to a lack of confidence in potential buyers in the quality of used computers.

Although trends in electronics are towards longer functional life, greater reliability and retirement prior to physical failure, failures do continue to occur. Processes exist which could aid in the prediction of failure of computer products. Their theoretical framework has been applied to the development of a system of signals which can be used to predict failure on an item-level. A variety of methods of prediction are used, including sensors embedded in products to monitor the causes and precursors to failure. The causes of failure in electronic devices have been examined, and the environmental factors which may lead to failures have been discussed specifically in the context of desktop computers, while considering the additional factors liable to affect the emerging generations of more portable systems.

A brief introduction to the history, technologies and applications of RFID has been given, laying out the advantages and drawbacks of various technologies and frequency bands. RFID has been proposed as the technology which could replace barcodes in asset tracking, and has strong potential to allow management of products at an item-level throughout their life cycles and through the disposal process.

2.8 References

- [1] J. Yu, E. Williams, M. Ju, and Y. Yang, "Forecasting Global Generation of Obsolete Personal Computers," *Environmental Science & Technology*, vol. 44, pp. 3232-3237, 2010/05/01 2010.
- [2] Press Release: PC Sales Will Top 370M Units in 2011. Worldwide Mobile PC Sales to Reach 227M Units. *Computer Industry Almanac*. Available: <http://www.c-i-a.com>
- [3] "Preparatory studies for Eco-design Requirements of EuPs; Lot 3, Personal Computers (desktops and laptops) and Computer Monitors, Final Report," IVF Industrial Research and Development Corporation 2007.
- [4] E. Williams, "Energy intensity of computer manufacturing: Hybrid assessment combining process and economic input-output methods," *Environmental Science & Technology*, vol. 38, pp. 6166-6174, Nov 15 2004.
- [5] H. B. Duan, M. Eugster, R. Hischer, M. Streicher-Porte, and J. H. Li, "Life cycle assessment study of a Chinese desktop personal computer," *Science of the Total Environment*, vol. 407, pp. 1755-1764, Feb 15 2009.
- [6] L. Q. Deng, C. W. Babbitt, and E. D. Williams, "Economic-balance hybrid LCA extended with uncertainty analysis: case study of a laptop computer," *Journal of Cleaner Production*, vol. 19, pp. 1198-1206, Jul 2011.
- [7] S. B. Boyd, A. Horvath, and D. Dornfeld, "Life-Cycle Energy Demand and Global Warming Potential of Computational Logic," *Environmental Science & Technology*, vol. 43, pp. 7303-7309, Oct 1 2009.
- [8] A. Goel and R. J. Graves, "Electronic system reliability: collating prediction models," *Device and Materials Reliability, IEEE Transactions on*, vol. 6, pp. 258-265, 2006.
- [9] E. D. Williams, R. U. Ayres, and M. Heller, "The 1.7 kilogram microchip: Energy and material use in the production of semiconductor devices," *Environmental Science & Technology*, vol. 36, pp. 5504-5510, Dec 15 2002.
- [10] L. Q. Deng and E. D. Williams, "Functionality Versus "Typical Product" Measures of Technological Progress A Case Study of Semiconductor Manufacturing," *Journal of Industrial Ecology*, vol. 15, pp. 108-121, Feb 2011.

- [11] C. W. Babbitt, R. Kahhat, E. Williams, and G. A. Babbitt, "Evolution of Product Lifespan and Implications for Environmental Assessment and Management: A Case Study of Personal Computers in Higher Education," *Environmental Science & Technology*, vol. 43, pp. 5106-5112, Jul 1 2009.
- [12] T. Cooper, *Beyond Recycling: The Longer Life Option*: New Economics Foundation, 1994.
- [13] E. Williams and T. Hatanaka, "Residential computer usage patterns in Japan and associated life cycle energy use," *2005 IEEE International Symposium on Electronics & the Environment, Conference Record*, pp. 177-182, 262, 2005.
- [14] K. Kawamoto, J. G. Koomey, B. Nordman, R. E. Brown, M. A. Piette, M. Ting, and A. K. Meier, "Electricity used by office equipment and network equipment in the US," *Energy*, vol. 27, pp. 255-269, 2002.
- [15] E. Williams, "Environmental effects of information and communications technologies," *Nature*, vol. 479, pp. 354-358, 2011.
- [16] A. Köhler and L. Erdmann, "Expected Environmental Impacts of Pervasive Computing," *Human & Ecological Risk Assessment*, vol. 10, pp. 831-852, 2004.
- [17] "Ireland national climate change strategy 2007-2012," Department of the Environment, Heritage and Local Government 2006.
- [18] N. Lior, "Sustainable energy development: The present (2009) situation and possible paths to the future," *Energy*, vol. 35, pp. 3976-3994, 2010.
- [19] R. A. Reinhardt, J. B. Payne, M. Babbitt, P. Nummikoski, C. A. Maze, and D. Dunning, "Gingival fluid IL-1 beta in postmenopausal females on supportive periodontal therapy.," *Journal of Dental Research*, vol. 77, pp. 230-230, 1998.
- [20] C. A. Webber, J. A. Roberson, M. C. McWhinney, R. E. Brown, M. J. Pinckard, and J. F. Busch, "After-hours power status of office equipment in the USA," *Energy*, vol. 31, pp. 2823-2838, 2006.
- [21] S. Sahni, A. Boustani, T. Gutowski, and S. Graves, "Reusing Personal Computers - Good or Bad for the Environment?," in *Sustainable Systems and Technology, 2010. ISSST '10. IEEE International Symposium on*, 2010.
- [22] N. van Nes and J. Cramer, "Product lifetime optimization: a challenging strategy towards more sustainable consumption patterns," *Journal of Cleaner Production*, vol. 14, pp. 1307-1318, 2006.
- [23] S. Ruth, "Green IT More Than a Three Percent Solution?," *Internet Computing, IEEE*, vol. 13, pp. 74-78, 2009.
- [24] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Davelder, D. Colle, B. Dhoedt, and P. Demeester, "Worldwide energy needs for ICT: The rise of power-aware networking," in *Advanced Networks and Telecommunication Systems, 2008. ANTS '08. 2nd International Symposium on*, 2008, pp. 1-3.

- [25] C. W. Babbitt and E. Ryen, "Bridging the gap in forward and reverse supply chains for evolving electronic products," in *Sustainable Systems and Technology (ISSST), 2010 IEEE International Symposium on*, 2010, pp. 1-6.
- [26] "Waste Framework Directive," in *2008/98/EC*, ed. European Union, 2008.
- [27] J. Quariguasi Frota Neto, G. Walther, J. Bloemhof, J. A. E. E. van Nunen, and T. Spengler, "From closed-loop to sustainable supply chains: the WEEE case," *International Journal of Production Research*, vol. 48, pp. 4463-4481, 2010/08/01 2009.
- [28] C. M. Rosen, J. Bercovitz, and S. Beckman, "Environmental Supply-Chain Management in the Computer Industry: A Transaction Cost Economics Perspective," *Journal of Industrial Ecology*, vol. 4, pp. 83-103, 2000.
- [29] R. Cahill, S. M. Grimes, and D. C. Wilson, "Review Article: Extended producer responsibility for packaging wastes and WEEE - a comparison of implementation and the role of local authorities across Europe," *Waste Management & Research*, vol. 29, pp. 455-479, May 1, 2011 2011.
- [30] (2004, E-Waste, the hidden side of IT equipment's manufacturing and use. *United Nations Environment Programme*.
- [31] (2012, 7th May 2012). *Website of Umicore group*. Available: <http://www.unicore.com/en/>
- [32] E. Williams, R. Kahhat, B. Allenby, E. Kavazanjian, J. Kim, and M. Xu, "Environmental, Social, and Economic Implications of Global Reuse and Recycling of Personal Computers," *Environmental Science & Technology*, vol. 42, pp. 6446-6454, 2008/09/01 2008.
- [33] C. M. Hagelüken, C.; van Kerckhoven, T. , "How WEEE Recycling can Contribute to Improving Access to Critical Raw Materials," presented at the CARE Innovation, Vienna, 2010.
- [34] A. Leung, Z. W. Cai, and M. H. Wong, "Environmental contamination from electronic waste recycling at Guiyu, southeast China," *Journal of Material Cycles and Waste Management*, vol. 8, pp. 21-33, 2006.
- [35] R. Kahhat and E. Williams, "Product or Waste? Importation and End-of-Life Processing of Computers in Peru," *Environmental Science & Technology*, vol. 43, pp. 6010-6016, 2009/08/01 2009.
- [36] L. Byster, S. Westervelt, R. Gutierrez, S. Davis, A. Hussain, M. Dutta, and J. Puckett, "Exporting harm, the high-tech trashing of Asia," ed, 2002.
- [37] "Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and Their Disposal," ed.
- [38] S. Westervelt, R. Gutierrez, Y. Takamyia, and J. Puckett, "The digital dump, exporting re-use and abuse to Africa," ed, 2005.

- [39] "21 Issues for the 21st Century: Result of the UNEP Foresight Process on Emerging Environmental Issues.," United Nations Environment Programme 2012.
- [40] G. L. Obersteiner, R.; Pertl, A.; Scherhauser, S.; Schmied, E. , "Formalisation of Informal Sector Activities in Collection and Transboundary Shipment of Waste in and to CEE – Introduction to the Project “TransWaste”,” presented at the CARE Innovation 2010, Vienna, 2010.
- [41] V. M. Thomas, "Demand and Dematerialization Impacts of Second-Hand Markets," *Journal of Industrial Ecology*, vol. 7, pp. 65-78, 2003.
- [42] S. W. Hickey and C. Fitzpatrick, "Combating adverse selection in secondary PC markets," *Environmental Science & Technology*, vol. 42, pp. 3047-3052, Apr 15 2008.
- [43] (June 21st 2008) The endowment effect: It’s mine, I tell you. *The Economist*.
- [44] "Military Handbook: Reliability Prediction of Electronic Equipment," in *MIL-HDBK-217F*, ed. United States of America, 1991.
- [45] K. L. Wong, "What is wrong with the existing reliability prediction methods?," *Quality and Reliability Engineering International*, vol. 6, pp. 251-257, 1990.
- [46] G. F. Watson, "MIL reliability: a new approach," *IEEE Spectr.*, vol. 29, pp. 46-49, 1992.
- [47] M. Pecht, B. Tuchband, N. Vichare, and Y. Qu Jian, "Prognostics and Health Monitoring of Electronics," in *Thermal, Mechanical and Multi-Physics Simulation Experiments in Microelectronics and Micro-Systems, 2007. EuroSime 2007. International Conference on*, 2007, pp. 1-8.
- [48] A. Ramakrishnan and M. G. Pecht, "A life consumption monitoring methodology for electronic systems," *Ieee Transactions on Components and Packaging Technologies*, vol. 26, pp. 625-634, Sep 2003.
- [49] S. Mishra, S. Ganesan, M. Pecht, and J. S. Xie, "Life consumption monitoring for electronics prognostics," *2004 Ieee Aerospace Conference Proceedings, Vols 1-6*, pp. 3455-3467, 4192, 2004.
- [50] S. Salemi, L. Yang, J. Dai, J. Qin, and J. B. Bernstein, *Physics-of-Failure Based Handbook of Microelectronic Systems: RIAC*, 2008.
- [51] N. M. Vichare and M. G. Pecht, "Prognostics and health management of electronics," *Ieee Transactions on Components and Packaging Technologies*, vol. 29, pp. 222-229, Mar 2006.
- [52] "D-19 Final Report on ELIMA Prospects and Wider Potential for Exploitation," 2005.
- [53] K. L. Wong, "Unified Field (Failure) Theory - Demise of the Bathtub Curve," *Proceedings Annual Reliability and Maintainability Symposium*, pp. 402-407, 1981.

- [54] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The impact of technology scaling on lifetime reliability," *2004 International Conference on Dependable Systems and Networks, Proceedings*, pp. 177-186, 909, 2004.
- [55] L. A. Coleman, "Army to Abandon MIL-HDBK-217," *Military & Aerospace Electronics*, vol. 3, pp. 1,6, September 14 1992.
- [56] J. D. Parry, J. Rantala, and C. J. M. Lasance, "Enhanced electronic system reliability - Challenges for temperature prediction," *Ieee Transactions on Components and Packaging Technologies*, vol. 25, pp. 533-538, Dec 2002.
- [57] P. L. M. P. E. B. Hakim, *Influence of temperature on microelectronics and system reliability*: CRC Press, 1997.
- [58] S. P. Mishra, Michael; Smith, Ted; McNee, Ian; Harris, Roger, "Remaining Life Prediction of Electronic Products Using Life Consumption Monitoring Approach," presented at the European Microelectronics Packaging and Interconnection Symposium, 2002.
- [59] "Failure Mechanisms and Models for Semiconductor Devices," JEDEC Publication JEP122G, October 2011.
- [60] "Military Handbook: MIL-HDBK-217 Reliability Prediction of Electronic Equipment," presented at the US Department of Defense, 1995.
- [61] K. L. Wong, "What is Wrong with Existing Reliability Prediction Methods?," *Quality and Reliability Engineering International*, vol. 6, pp. 251-257, 1990.
- [62] M. Ohring, *Reliability and failure of electronic materials and devices*: Academic Press, 1998.
- [63] (7th November 2011). *Speedfan Temperature, Volatage and Fan Speed Monitoring*. Available: <http://www.almico.com/speedfan.php>
- [64] (7th November 2011). *Core Temp*. Available: <http://www.alcpu.com/CoreTemp/>
- [65] E. Rotem, J. Hermerding, C. Aviad, and C. Harel, "Temperature Measurement in the Intel Core Duo Processor," presented at the Proceedings of 12th International Workshop on Thermal investigations of ICs, THERMINIC 2006, 2006.
- [66] "Dual-Core Intel® Xeon® Processor 5100 Series Datasheet," August 2007.
- [67] (3rd November 2011). *MAX661 PEGI to I²C Convertor - Overview*. Available: <http://www.maxim-ic.com/datasheet/index.mvp/id/5368>
- [68] (3rd November 2011). *HP Business Notebook Innovations: Spill-Resistant Keyboard with Drains*. Available: http://www.hp.com/sbso/solutions/pc_expertise/professional_innovations/reliability.html#DrainSystem
- [69] A. Ramakrishnan and M. G. Pecht, "A life consumption monitoring methodology for electronic systems," *Components and Packaging Technologies, IEEE Transactions on*, vol. 26, pp. 625-634, 2003.

- [70] J. Landt. (2005) The History of RFID. *IEEE Potentials*. 8-11.
- [71] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*: John Wiley & Sons, Inc., 2003.
- [72] R. Want, "An introduction to RFID technology," *Ieee Pervasive Computing*, vol. 5, pp. 25-33, Jan-Mar 2006.
- [73] T. E. Karygiannis, Bernard; Barber, Greg; Bunn, Lynn; Phillips, Ted, "Guidelines for Securing Radio Frequency Identification (RFID) Systems: Recommendations of the National Institute of Standards and Technology," US Department of Commerce 2007.
- [74] K. Finkenzeller, *RFID Handbook*, 2 ed., 2004.
- [75] H. G. Bhatt, B., *RFID Essentials*, 2006.
- [76] M. Stutz, V. M. Thomas, and S. Saar, "Linking bar codes to recycling information for mobile phones," presented at the Proceedings of the International Symposium on Electronics and the Environment, 2004.
- [77] A. G. Kulkarni, A. K. N. Parlikad, D. C. McFarlane, and M. Harrison, "Networked RFID systems in product recovery management," presented at the Proceedings of the International Symposium on Electronics and the Environment, 2005.
- [78] S. Saar and V. Thomas, "Toward Trash That Thinks: Product Tags for Environmental Management," *Journal of Industrial Ecology*, vol. 6, pp. 133-146, 2002.
- [79] C. H. Hans, Karl A. Thoben, Klaus-Dieter, "Improving reverse logistics processes using item-level product lifecycle management," *International Journal of Product Lifecycle Management*, vol. 4, pp. 338 - 359, 2010.
- [80] V. M. Thomas, "Product Self-Management: Evolution in Recycling and Reuse," *Environmental Science & Technology*, vol. 37, pp. 5297-5302, 2003/12/01 2003.

Chapter 3

BACKGROUND & STATE OF THE INDUSTRY

3.1 Introduction

In this chapter, the current state of the computer industry is examined in the context of sustainability implications of computers. As examined in the previous chapter, the reliance of developed societies on ICT imposes a considerable burden on the environment, both through the effects of energy generation for ICT while in use, and through the direct environmental effects of the manufacture and disposal of the billions of computers in use today. In this chapter the environmentally beneficial aspects of ICT, which can help to offset the harmful impacts outlined in the literature review, are examined.

In order to be defined as sustainable, a system must be balanced economically, socially and environmentally. The social and economic effects of ICT are examined, and the benefits are considered in the context of the negative environmental effects. Both socially and economically ICT can also be considered to have negative impacts, and these are presented here also.

As anthropogenic environmental impacts have been explored in greater detail in the last number of years, a number of initiatives have been enacted by governments, industry and environmental organisations to reduce the environmental burden posed by humans. The major directives and initiatives which are aimed at the ICT sector are presented, and the characteristics of each examined. Where innovative technological trends introduce new methods for availing of ICT benefits with reduced environmental

burdens, these are also presented, and the effect of their introduction on the environmental impact of the sector is discussed.

3.2 Environmental Benefits

While there has been considerable attention paid to the negative impacts of technology on the environment, it is also necessary to consider the positive effects of the sector. Although manufacturing, use and disposal contribute definite effects, by using ICT some environmental impacts in other areas of society can be reduced significantly.

An obvious example of this effect is in telecommuting. In many roles, it is no longer always necessary for employees to work in an office owing to the cheap and instantaneous communication provided by ICT. As a result, many people now work from home, and as such do not commute to and from an office on a daily basis. This reduces congestion and the environmental impact of the personal transportation of workers. If adopted on a larger scale, telecommuting could potentially reduce the need for office space, reducing the impacts associated with building, maintaining, heating and cooling office areas. However, although transport accounts for a large proportion of energy use, the maximum direct benefits to be gained from this behaviour change are thought to be relatively modest; estimated at 1-2% of the total US energy use [1].

Smart buildings are another area in which ICT can potentially lead to energy savings. Because a large amount of energy used in homes is currently wasted on heating and lighting for empty spaces there is potential for more intelligent solutions to lead to significant savings. Smart meters will in the near future be capable of informing people what proportion of power use is for each purpose, highlighting inefficiencies and major drains of energy [2]. Programmable and “learning” thermostats also have potential to

decrease the energy wasted in heating empty spaces, by pattern recognition along with remote control through smart phones.

As the capabilities of ICT continue to increase, and it becomes more integrated into infrastructure and the day-to-day lives of a greater number of people, further solutions will emerge which may have further positive effects on the environment. Continued development and penetration of ICT is desirable therefore, for reducing the environmental impact of other areas. This reduction is meaningless from an environmental perspective, however, if the impact of the technology itself is greater than the reduction in impact offered through its implementation.

3.3 Technological Reliance: Social and Economic

It is important when considering any commodity from an environmental perspective that the complimentary effects of this commodity are also considered in an economic and social context. In order to be viable, a commodity must be sustainable not only ecologically, but also socially and economically. In the case of PCs, their economic and social benefits serve to make them indispensable in modern life. This section will discuss the economic and social effects of PCs as a contrast to the examination of their environmental effects. An in-depth examination of these effects is beyond the scope of this work; however the aim of the section is to impart an impression of the influence of technology, both beneficial and harmful, on modern society.

3.3.1 Social Effects

Starting with the development of e-mail as a communicative medium, ICT has repeatedly revolutionised the means by which we communicate. The types of communication media in use constantly evolve and multiply, to suit the developing

demands of the current generation of technology users. The outcomes have completely changed the ways in which we interact, to the extent that even how people think about communication has altered beyond recognition.

The benefits of improved, lower cost, real-time communications are evident in applications such as video-conferencing and instant messaging. Where even a decade ago communications between countries was considered expensive and difficult, it is now normal for voice over internet protocol (VoIP) implementation to be possible using standard mobile phone hardware, meaning that anybody with access to an internet connection can easily make contact across the globe for free, or at least at relatively low cost.

Social communication sites such as Facebook have become the de facto method by which many social groups communicate, developing beyond their initial young adult audience to become an integral part of the lives of people across all ages, social groups and ethnicities. While the increase in communication is seen as desirable from the point of view of the users, critics of the medium sometimes claim that the art of proper interpersonal socialisation is being lost owing to the impersonal interface which has replaced face-to-face interaction. A further danger of this method of interaction is that it can sometimes be difficult to know with whom information is being shared. Examples of unwanted proliferation of information over social networks can be seen in the occasional cases where hundreds or even thousands of people unexpectedly attend private parties, owing to unwanted sharing of information online. One further criticism often aimed at social sites is that their privacy controls can be cryptic, resulting in some information meant for friends often inadvertently being shared with a larger group.

In the recent past, the effects of social media have been visible on a massive scale in the Middle East and North Africa, where Twitter and Facebook are credited with having played a central role in the propagation of information from otherwise censored nations, and allowed media coverage of civil unrest which may in other circumstances have been less publically broadcast. Social media were also used to organise protests in London during the same period in 2011 following the shooting of a man by local police, and the capability of the social media sources to bring together people with a common agenda, for right or wrong, was clearly demonstrated by the subsequent riots.

A further potential negative impact of increased use of social media is that, because many young people have a better understanding of these sites than their elders, it can be difficult for parents to fully comprehend the level of interaction involved. It is therefore difficult for parents to protect young people online, and as a result, certain sites on the internet have developed a reputation for allowing inappropriate interactions. This situation, while unfortunate, may only really be prevented if parents keep informed about the actions of their children online, and educate themselves and their children about internet safety.

Availability of ICT has also revolutionised education in the last two decades. In the developed world, most students will have access to computers at home and at school, and a rapidly growing proportion of students have a laptop or PC of their own. With access to sites such as Wikipedia and Google Scholar, the effort and cost involved in obtaining information has decreased to almost nothing. The effort required to collaborate is also decreased, as access to cloud based services including Dropbox and Google Cloud means that collaborators can be kept aware of each other's progress in real time. As an effect of this, students are able to gain a much broader scope of

knowledge by researching diverse opinions in an area, resulting in much more balanced education.

One drawback of the use of the internet for research is the depth of information available, and owing to the lack of validation and control over information sources there is potential for misinformation to be quickly propagated. In general, this can be avoided by the development of conscientious research methodologies, and verification of any information from multiple sources. However research suggests that a potential corollary of the availability of information is that the use of technology to instantly answer any query may have the effect of decreasing the retention of information in memory [3].

Research has shown that acquiring computer literacy can have a dramatic effect on decreasing the rate of poverty by bridging the “digital divide” which has emerged as a key factor both as a cause and as an effect of the gap between socio-economic levels. In the developing world, and in socio-economically underprivileged communities, there are several schemes devoted to the distribution of laptops among students in an attempt to increase computer literacy [4].

From the perspective of the educator, the availability of ICT has increased the variety of ways education can be imparted. In a class or lecture environment, information can now be communicated in an interactive form, as classroom access to ICT becomes standard. As the free availability of information increases, educators must adapt to the development in the learning processes of students, and may become guides in how research is conducted online [5]. Distance learning has also been revolutionised through the internet, as educators can now reach students who would have otherwise been unavailable owing to distance, time commitments, or disability.

In the area of health, the most obvious benefit to, and drawback of the interaction of ICT and healthcare is the ability of people to self-diagnose illnesses. Services such as WebMD enable users to enter a set of symptoms and return a list of potential diagnoses. This enables users to avoid visits to healthcare professionals for minor ailments, and helps to allay fears associated with unfamiliar symptoms. While the advice of the sites like this is generally qualified by disclaimers advising consultation with healthcare professionals, many users do not check the reliability of a source after receiving a diagnosis. Again, this can be remedied by conscientious research methods, however the result of a misdiagnosis can obviously be drastic [6].

General information is freely available online regarding day to day health and wellbeing, and early education has been shown to have a causal effect on health later in life. As a result, access to knowledge relating to health throughout life may be expected to result in a proportionally healthier population, potentially increasing quality of life and decreasing the cost of healthcare on a national basis [7].

A new benefit of ICT in the healthcare sector is the use of cloud services by healthcare professionals. E-health systems are set to revolutionise how patient records are stored and accessed. With instant access to a patient's medical history, medical professionals would be more capable of making informed decisions in hospitals, even in emergency treatment situations. Professionals are also capable of remotely monitoring patients' condition, and even their vital signs, using mobile devices [8]. One major concern about this form of healthcare, and a cause of its relatively slow adoption, is the availability of confidential information on the internet, and the security associated with that. As any e-health system would consist of sensitive data being handled by people who are not necessarily technically adept, a strong security infrastructure is vital.

Culturally, technology is regarded as a double-edged sword. Through use of ICT for dissemination of art and ideas, culture is more freely available to a greater number of people than ever before [9]. The cultural danger of ICT is seen in the loss of the unique identity of smaller cultures due to constant immersion in international media. Westernisation and cultural homogenisation could be largely accelerated by access to technology, but the technology also supplies the means to strengthen and preserve culture by maintaining cultural records online and using new media to broaden the reach of traditional cultural information, for example through dissemination of material to expatriots.

3.3.2 Economic Effects

Economically, ICT can be seen to have mixed effects. As with many technologies, the development of ICT renders certain jobs obsolete, while introducing new ones. An example of an industry threatened by technology is the print newspaper industry. This is in decline as an increasing number of readers get their news information from blogs and websites. A corresponding economic boost affects successful bloggers, and those newspapers which successfully transition to online platforms, in the form of income from advertisements hosted on their websites. Ability to adapt to changing technological trends is a requirement in order to survive for many industries, but it also provides potential benefits associated with exposure to larger audiences and new demographics. A new trend can be seen in the transition from purely web-based delivery of services to the development of applications for smartphones and tablets. For many “bricks and mortar” companies, adaption to new technological trends may make the difference between growth and decline.

The economic effects of ICT on individuals or small companies are largely dependent on their level of understanding and uptake. As a stronger emphasis is placed

on social media for purchasing decisions, companies with established social media profiles have a better chance of success. Consequently, the position of “Social Media Expert” has become common, as those with more experience in social media sites capitalise on their knowledge by helping companies to maximise their online marketability.

Businesses can also expand their customer pool through online sales, reaching customers who would otherwise be unable to obtain their products. This is possible for large companies through their websites, but the internet provides opportunities for smaller companies and individuals through sites such as eBay and Amazon.

3.3.3 Conclusion

While there are certain adverse effects associated with technology adoption, the benefits can greatly outweigh the drawbacks. Technology has become integrated into people’s lives to such a degree that most people would be unable to conceive of a life without PCs or other technology, and this trend seems certain to continue, and even to expand to socio-economic groups for whom ICT had traditionally been considered a luxury.

The conclusion to be drawn from these examinations is that many of the economic and social disadvantages of technology are only realised through failure or refusal to adapt to technological trends in a timely manner. A policy, either on a personal or national level, of reducing technological capabilities - while potentially beneficial to the environment - would be disastrous both economically and socially.

Given our dependence on ICT, and fact that the reduction of this dependence does not seem desirable or practical from an economic or social perspective, efforts must be made to improve the environmental sustainability of our reliance. The

environmental effects of ICT must be examined, and methods of minimising the detrimental effects should be implemented.

3.4 Environmental Stewardship

There are numerous organisations globally who are invested in the monitoring and control of the environmental impact of technologies. These provide guidance to manufacturers on best standards in environmental design and manufacture, as well as information to consumers about the environmental impacts of products and the relative performance of different manufacturers.

3.4.1 Environmental Labelling

Eco-Labels offer consumers the tools to differentiate between products on the basis of environmental performance. Their goal is to break environmental performance into simple to comprehend metrics and give customers a general overview of the total environmental quality of a product, as well as a metric for comparison of competing products on the basis of environmental performance.

As well as identifying the best in class products, there are a number of initiatives to reduce the use of products which do not meet a minimum requirement of environmental efficiency through regulatory and legislative channels. Although legislative systems can be effective in rendering environmental benefits, they have the drawback of requiring long processes of validation and discussion before it is possible to implement them. An example of this is the case of the European Energy related Products (ErP) directive, developments of which were intended to introduce strict requirements for all products to be awarded with the “CE” mark, a requirement of selling products in the European Union. Unfortunately, the requirements for PCs, when included in the directive, were implemented as a voluntary series of recommendations,

as it would be difficult for each new product to be tested to the extent that would be required for certification before release. Only recently has draft legislation been presented for this category of product[10]. In this, some Eco-Labels have somewhat of an advantage, particularly in a field where the status quo changes as quickly as in electronics.

3.4.2 Energy Star

For years the best known environmental label for computers, monitors and other office equipment, the Energy Star is awarded on a product-by-product basis to acknowledge performance which is in line with environmental best practice at any time. This label is primarily focused on energy use and efficiency, and is updated to reflect improvements in technology and their expected influence on manufacturing standards. This provides manufacturers with the information they require regarding the current best practices in environmental manufacturing processes.

For computers, the Energy Star standard defines rules governing the typical power supply efficiency, and the power consumption while in active, sleep, idle and off operating modes. A tiered system has also been introduced which defines four categories of computers dependent on number of processing cores, physical memory and the presence of a graphics card. Each tier of computer has a custom set of requirements which apply to it. This allows high-end computers to also achieve the standard, while retaining the incentive to those manufacturing more economical systems to also conform to best practice.

In order to ensure that their products conform to the requirements of the Energy Star program, manufacturers are required to test products themselves, and to register the results with the Energy Star program. The Energy Star is then awarded, although units

may subsequently be subject to testing to verify their rating. By allowing self-assessment in this way, products can be awarded certification in a manner which works with the fast development cycles of computer systems, although the system is potentially open to abuse by companies misrepresenting their products' test results[11].

3.4.3 EPEAT

Originally developed in the United States, but recently gaining popularity around the world, the Electronic Product Environmental Assessment Tool (EPEAT) system builds on the specifications of the Energy Star to attempt to identify products which exhibit exceptional environmental performance throughout their life cycles. It operates by administering an award; bronze, silver or gold, dependent on the overall environmental performance of a product. The award covers several categories of product, and allows manufacturers with exceptional environmental performance to be properly acknowledged while also providing a system allowing customers to differentiate products based on their environmental performance.

Compared to Energy Star, which governs primarily energy efficiency in the use phase, EPEAT attempts to render a more thorough overview of all life cycle stages by defining requirements and optional achievements under the following headings:

- Materials;
- Design for end of life;
- Life cycle extension;
- Energy conservation;
- Active end of life management;
- Packaging;
- General corporate environmental performance.

By developing a more thorough overview of the product life impacts, the EPEAT system identifies the products which perform best at all stages of the life-cycle. While each area has obligatory achievements, there are also optional goals, and attainment of 50% or 75% of these goals result in systems being awarded the EPEAT Silver or Gold label respectively [12].

Similar to the Energy Star, EPEAT operates based on self-declaration and random testing, allowing declaration of a system before release based on test results, and not affecting the release schedule of a manufacturer. Products which are awarded a standard may be tested by EPEAT to ensure that the declarations given were true.

For PCs, one of the prerequisites for any EPEAT award is acquisition of the current Energy Star certification, ensuring that the use-phase requirements are kept up to date. For EoL, however, while design for disassembly and efficient recycling is encouraged, EPEAT are seeking ways in which to encourage lifetime extension, as it is difficult to quantify efforts from manufacturers in this regard.

3.4.4 US Executive Order

In 2009, the US implemented rules in the ‘Federal Leadership in Environmental, Energy, and Economic Performance’ executive order that established that all Federal agencies must “ensure procurement preference for EPEAT-registered electronic products” [13]. By combining the eco-label with the legislative process in this way, it is ensured that any updates which EPEAT introduce are automatically adopted by the procurement agencies, circumventing the legislative delays referenced previously.

3.5 Technological Trends

Owing to the rate of progress of ICT, it’s important to consider not just the state of the industry at any given moment, but also to take into account the advancements

which are being made, and the effects that these may have in the future. This section considers technological trends which suggest that in the near future use phase of computers may be of less significance than has traditionally been the case. While the overbearing technological trends in terms of power consumption and manufacturing efficiency have been discussed in the literature review section, these do not take account of disruptive technologies, which fundamentally change the impact of ICT by changing the way in which we interact with it, thereby potentially mitigating some of the environmental impact.

3.5.1 Cloud Computing

Cloud computing is a technological direction which offers economies of scale to the personal computer market. Leveraging the availability of high speed broadband to most homes, cloud computing has the potential to shift the computation burden from the home PC to large servers. As a result, hardware requirements for the home PC are reduced. This reduces both the manufacturing burden of the computers and the use phase power requirements.

Web-based data storage services such as Dropbox are becoming a dominant force in data storage. As confidence in the availability of internet connections anywhere grows, consumers are willing to rely on more of their services being provided by internet-based companies. This gives the benefit of access to information from anywhere, and the ability to easily share information. It also provides a measure of data security, as any information stored in this way is protected from damage which occurs to a PC. While the Universal Serial Bus (USB) drive has not yet been replaced as the primary form of data transport and storage, the benefits of instant, secure access to data from anywhere has the potential to decrease the importance of physical storage, and the associated semiconductor manufacturing requirements.

A further advancement based on web-based storage is online document editing services. Using services such as Google Documents (now rebranded Google Drive) and Office 365, it is possible to directly edit documents within a browser window, and store them online without ever creating a physical copy in the static memory of the PC. These documents can then be accessed and edited from any PC with browser capabilities, regardless of the specific operating system or software installed. A further advantage of online data editing lies in the development cycle. While traditional document editors' functionality is relatively static within a version; only improving when upgrades become available, web-based services tend to be continuously in development, and the features and services can dynamically change at a much faster rate. While the capabilities of the online editors cannot yet compare with the functionality of their PC-based counterparts, constant development and a lower time to market means that the differences are becoming less significant for many users.

“Content Cloud” computing has recently been used in more personal ways by companies such as Google, Amazon and Apple. Amazon led the pack in this with the Kindle e-book reader, which can automatically synchronise to a user's e-book collection whenever it connected to the internet, and Amazon offered free global access to wireless internet for life to any customer purchasing the ‘3G’ model. All three of these companies now offer a personal media cloud, which guarantees the availability of a person's media on internet-capable devices.

While cloud computing services have the potential to reduce power requirements for the end-user of a device, as well as the initial manufacturing costs and impact, the indirect cost of greater reliance on the internet is realised in the increase in the number of data centres which provide these internet services. These operate more efficiently than home computers, as economies of scale mean that a data centre can optimise the

amount of processing power being used to meet current demand, changing the number of processors as demand alters, and reducing the power consumed when compared to the equivalent operation of many home PCs.

3.5.2 Digital Delivery

One area of use phase impact which is rarely considered through LCA studies is the software which supports the functioning of the PCs. Delivery of software to users has transportation impacts, as well as the impacts of the manufacture and disposal of packaging and data media such as DVDs.

Over the last number of years an increasing number of software producers have taken advantage of the penetration of high speed broadband to introduce the practice of ‘digital distribution’, allowing users to download software over the internet. While distribution of certain software, such as drivers, updates and firmware through the internet is well-established., companies are now offering complete functional programs in this way. Initially this a popular method of distributing games with online components, as the process aided digital rights management (DRM) for games by linking the purchased game to a user account. More recently Microsoft has introduced digital delivery for their products, including the Windows operating system prior to release, and activation on release day. This incentivises digital delivery for enthusiastic customers by reducing the effort required to obtain the product at launch [14].

Open-source software has a unique advantage in this area, owing to the relative lack of DRM concerns. While most developers pay expensive server costs to ensure their software is stored securely and distributed quickly to paying customers only, open source software such as Linux can utilise distributed computing by making their products freely available using delivery methods such as file-sharing websites or

torrents. In this way, the product is stored on users' computers, and the distribution occurs primarily between current users and new downloaders. This removes the economic burden of data storage and delivery on the developers.

3.5.3 Solid-State drives

While the movement towards all-online computing may have begun, for the majority of people it is not yet the reality. In the meantime, flash drive technology offers several benefits over the traditional spindle-based hard drives. The semiconductor based technology is significantly less prone to mechanical failure than the moving parts of standard drives. It is much more suitable for applications in mobile computing, where the motion of the computer would have the potential to damage the spindle or memory plate of a Hard Disk Drive (HDD).

For most people, the primary benefit of flash-based solid-state drives (SSDs) compared to HDDs is access time. Because flash memory doesn't need to "spin up" in the same manner as other drives, data access is instantaneous. Speed of data transfer is greatly improved, which has particular benefit when considered in the context of restarting computers. Because information is available on the SSD almost immediately on receiving power, boot times can be drastically reduced [15]. This has the potential to incentivise powering down computers rather than leaving them in idle mode overnight, which could theoretically affect overall environmental impact.

SSDs in the past had lower capacity at far higher cost per GB than HDDs. These disparities are decreasing as integration of SSDs into computer systems is increasing. Combined with the lower dependence on local storage offered by cloud computing, lower capacity SSDs have the potential to become the standard, and HDDs the exception, for personal computing.

3.6 End of Use

In this section the range of End of Use (EoU) processes are examined based on their practicality to in relation to PCs. It is relevant for this project that the difference between EoU and EoL is correctly understood. EoU occurs when a particular user ceases to use a product. EoL, for the purposes of this work, will be defined as the point at which the product is treated in the waste stream. A product can therefore have several users, and EoUs before its eventual EoL. Similarly, even if it has only one user, a product's EoU can occur significantly before the EoL if it is stored for a time before disposal, a common trend with computers.

3.6.1 Functional and Physical Obsolescence

In identifying the optimal EoU process for a PC, it's important to understand the reasons for which PCs are replaced. In a large proportion of cases, systems are replaced despite still being technically functional, either because their performance has degraded to the point where it is no longer acceptable, or because the users' requirements have changed, and a system offering more functionality is required. This is defined as functional obsolescence, as the system is still operational, but no longer capable of meeting its desired function[16].

Physical obsolescence occurs when a product is no longer capable of operating in the manner for which it was originally designed. An example is in older laptop batteries, where after a certain number of charge/discharge cycles it's necessary to replace them, as they no longer hold charge. Advances in reliability can be expected to result in a reduction in the occurrence of physical obsolescence in computers during their operating lives [17].

Functional obsolescence can be examined from two perspectives. The first is the gradual degradation which the performance of most computers experience throughout their lives, and the second is the emergence of newer technologies rendering current systems obsolete. In the case of the former type, the ‘slow-down’ that systems experience throughout their lifetimes can often be attributed to a general accumulation of programs and files that occurs through normal use of a computer. A new PC is generally provided with only the operating system and certain core programs. As it becomes customised to a user’s needs, and as those needs change throughout its life, programs and files are added to and removed from the file system. Through this process the system becomes fragmented, with pieces of files dispersed throughout hard drives. The result is a longer processing time for the PC. This is easily solved through good PC housekeeping; simple folder structuring, removal of unnecessary programs and files and regular defragmentation of the file system. Many programs will also install background processes for faster loading or service provision that also result in an overall slow-down of the system. While the incremental effects may be minor, the cumulative impacts become noticeable throughout the life of the PC. Functional obsolescence by this means may be delayed by uninstalling all non-essential programs, or in some cases, by a complete reinstallation of the operating system.

The second type of functional obsolescence occurs with the change of a user’s requirements or the introduction of a disruptive technology necessitating an upgrade. In the past, technological watersheds have occurred around the introduction of new versions of operating systems as users used the introduction of newer software as a cue to acquire appropriate hardware. Similarly, when core components such as office suites were updated, users would need to acquire newer systems before adopting them, owing to the increased operating requirements. Recently, however, developers have focused on

reducing the operating requirements of newer software to allow it to run effectively on a greater range of hardware. This can be seen in the operating requirements of Microsoft Windows and Microsoft Office over time. While earlier generations of the products took advantage of technological advances to introduce increasing capabilities, resulting in higher operating requirements, recent developments have required less significant advancements in hardware capabilities, despite the continuing increase in the capabilities of technology [18-20]. The plateau in requirements of the core functions of PCs has the potential to extend the functional lifetime of systems by further delaying functional obsolescence

Disruptive technologies as a cause of functional obsolescence have also decreased. In the past, the introduction and widespread adoption of a new technology, such as CD, DVD or ZIP drives necessitated the widespread upgrade of computers to ensure compatibility. In general the trend was towards higher data density, and upgrades were necessary on a regular basis in order to keep pace with technological trends. This has been changed by the ubiquitous adoption of the Universal Serial Bus (USB) interface, and more recently by the spate of online storage services. All modern computers come with several USB ports, enabling the attachment of a USB key drive or external hard drive, or any of a massive variety of peripheral devices, including external drives for accessing any other form of media.

While advances in USB technology have occurred, they have been rare, and characterised by backwards compatibility, offering performance benefits in data rate while still supporting older peripherals. While disruptive technologies are by their nature hard to predict, continuation of this trend also points towards levelling-off of the hardware requirements of PCs, and a corresponding increase in their lifetimes, as functional obsolescence by this means is delayed.

Hardware indications are towards increased PC lifetime and delayed obsolescence, but this is not found to correspond to user behaviour, as computer use phase durations are still gradually reducing [21]. The result is operational systems being replaced, and entering the EoU phase. There are several potential destinations for systems at this point.

3.6.2 Storage

The literature shows that a large amount of people will store an EoU computer for some length of time before eventually disposing of it [21, 22]. From an environmental perspective, this could be regarded as benign, as the material is not entering the waste stream, and in the interim technological processes for treatment of used computers could improve. Large-scale storage of used computers in landfill has even been proposed as a short-term solution to the e-waste problem [23]. The opportunity cost of storage is the devaluation over time and eventual functional obsolescence of a system, resulting in its disposal at a later stage.

3.6.3 Remanufacturing

For the purposes of this project, remanufacturing will be considered to include the process of upgrading; keeping the same case, and replacing components, and also the process of rebuilding; using functional components from a defunct PC in another system.

Owing to the interchangeable nature of the majority of components in computers, there is high potential for remanufacturing of non-functional systems. The form factor of a component is important in determining if it can be integrated into a used system. If the form factor of the new component and the one it replaces are not the same, replacement may be difficult or impossible.

The form factor of the standard desktop HDD has remained the same since 1983. This is also true for optical drives, PCI peripherals, power supplies and even whole cases. The component which undergoes the most frequent changes in form factor is the processor [24], for which socket sizes are updated approximately annually. While this means that it would be difficult to remanufacture a computer with a non-functional processor, the environmental costs of new processors are such that it would be inefficient to do so in any case, based on the proportion of the total life cycle impact which is embedded in the manufacture of motherboard and processor.

Because of the relative ease of integration of functional components between systems, remanufacturing offers considerable environmental benefit as an end of use process. Depending on the level of replacement of components required, considerable value can be returned to a non-functional system, with low level of technical skills and little or no specialist equipment required. Through provision of a new operating system, many systems can be returned to a 'better than new' state and become viable for resale in secondary markets.

3.6.4 Reuse

Reuse, to distinguish from remanufacturing for the purpose of this project, is considered to be specifically when no hardware is replaced and a computer begins its second life as-is. In general, this is recognised as the optimal method for lifetime extension of PCs, owing to low cost, low environmental effects and the lack of skilled labour required [25]. In the case of systems which are only functionally obsolescent, reuse requires very little investment of time or labour; however PC reuse as an industry has not developed in Ireland or internationally owing to several barriers which will be examined in the next section.

The overheads associated with direct reuse of PCs would only be the removal of personal information from the used system, and the installation of a new and officially licenced operating system. As such, the environmental impact of this process is low, and by extending the life of a computer it reduces the overall impact of the ICT sector, compared to the purchase of a new system.

3.7 Conclusion

The social and economic benefits which may only be realised through continued proliferation of ICT mean that it is unlikely that the digital revolution will slow down in the foreseeable future. While certain environmental benefits are realised through adoption of ICT technologies, the effects of the manufacturing and disposal phases, combined with the incremental increase in normal operating energy use for desktop PCs suggest that action is needed to mitigate to some extent the scale of the burden which ICT places on our environment.

Industry groups, legislative instruments and voluntary organisations are serving to mitigate the impact of ICT reliance, with the result that significant advances have been made in the areas of regulation and of manufacturing processes; however even where end of life processes are included in consideration of overall environmental impact, it is difficult to ensure that manufacturers are considering life extension in the design of their products, and as a result this cannot be prioritised through the use of rewards and incentives.

At the same time, with increases in reliability and decreases observed in the operating life of PCs, a greater number of functional PCs than ever are being retired and stored or disposed of. In order to tap into this resource of used but functional systems

and components, a framework for the redistribution of used PCs is required. A potential solution for this function is outlined in the following chapter.

3.8 References

- [1] H. S. Matthews and E. Williams, "Telework Adoption and Energy Use in Building and Transport Sectors in the United States and Japan," *Journal of Infrastructure Systems*, vol. 11, pp. 21-30, 2005.
- [2] R. J. Meyers, E. D. Williams, and H. S. Matthews, "Scoping the potential of monitoring and control technologies to reduce energy use in homes," *Energy and Buildings*, vol. 42, pp. 563-569, 2010.
- [3] B. Sparrow, J. Liu, and D. M. Wegner, "Google Effects on Memory: Cognitive Consequences of Having Information at Our Fingertips," *Science*, vol. 333, pp. 776-778, August 5, 2011 2011.
- [4] (20/02/2009). *One Laptop per Child*. Available: <http://laptop.org>
- [5] A. Unwin, "The professionalism of the higher education teacher: what's ICT got to do with it?," *Teaching in Higher Education*, vol. 12, pp. 295-308, 2007.
- [6] D. P. McDaid, A-La, "Bupa Health Pulse 2010 - Online Health: Untangling the Web," The London School of Economics and Political Science 2010.
- [7] S. Lindsay, P. Bellaby, S. Smith, and R. Baker, "Enabling healthy choices: is ICT the highway to health improvement?," *Health*, vol. 12, pp. 313-331, July 1, 2008 2008.
- [8] S. A. Shahrestani, "ICT healthcare: overcoming complex barriers for successful deployment," presented at the Proceedings of the 5th WSEAS international conference on Dynamical systems and control, La Laguna, Spain, 2009.
- [9] (13th May). *Louvre Museum Official Website*. Available: <http://www.louvre.fr>
- [10] "EU Eco-Design of Energy-related Products Directive," in *2009/125/EC*, ed. European Commission, 2010.
- [11] (4th May 2012). *Energy Star website*. Available: <http://www.energystar.gov>
- [12] (8th July 2009). *EPEAT*. Available: <http://www.epeat.net>
- [13] "Federal Acquisition Regulation; FAR Case 2006–030, Electronic Products Environmental Assessment Tool (EPEAT)," 2009.
- [14] L. B. Hsu, Scott "Redefining Software and Electronic Hardware Packaging within an evolving technology environment," presented at the CARE Innovation 2010, Vienna, Austria, 2010.
- [15] R. Paul. (2009). *Ubuntu 9.10 boot optimizations: 5 second startup with an SSD*. Available: <http://arstechnica.com/information-technology/2009/09/ubuntu-910-alpha-6-released-boot-optimizations-arrive/>

- [16] S. Okumura, T. Morikuni, and N. Okino, "Environmental effects of physical life span of a reusable unit following functional and physical failures in a remanufacturing system," *International Journal of Production Research*, vol. 41, pp. 3667-3687, 2003/01/01 2003.
- [17] K. L. Wong, "What is wrong with the existing reliability prediction methods?," *Quality and Reliability Engineering International*, vol. 6, pp. 251-257, 1990.
- [18] Microsoft. (5th May 2012). *Windows 7 System Requirements*. Available: <http://windows.microsoft.com/systemrequirements>
- [19] Microsoft. (5th May 2012). *Running the Windows 8 Consumer Preview: System requirements*. Available: <http://blogs.msdn.com/b/b8/archive/2012/02/29/running-the-consumer-preview-system-recommendations.aspx>
- [20] Microsoft. (5th May 2012). *Windows Vista System Requirements*. Available: <http://windows.microsoft.com/en-us/windows-vista/products/system-requirements>
- [21] C. W. Babbitt, R. Kahhat, E. Williams, and G. A. Babbitt, "Evolution of Product Lifespan and Implications for Environmental Assessment and Management: A Case Study of Personal Computers in Higher Education," *Environmental Science & Technology*, vol. 43, pp. 5106-5112, Jul 1 2009.
- [22] S. W. Hickey and C. Fitzpatrick, "Combating adverse selection in secondary PC markets," *Environmental Science & Technology*, vol. 42, pp. 3047-3052, Apr 15 2008.
- [23] R. F. Kahhat Jr and E. Kavazanjian, "Preliminary feasibility study on the use of mono-disposal landfills for e-waste as temporary storage for future mining," *Proceedings of the 2010 IEEE International Symposium on Sustainable Systems and Technology, ISSST 2010*, 2010.
- [24] Intel. (5th May 2012). *Intel Processor Quick-Reference Guide*. Available: <http://www.intel.com/pressroom/kits/quickreffam.htm>
- [25] E. D. Williams and Y. Sasaki, "Energy analysis of end-of-life options for personal computers: resell, upgrade, recycle," in *Electronics and the Environment, 2003. IEEE International Symposium on*, 2003, pp. 187-192.

Chapter 4

PROPOSED SOLUTION: RFID AND LCM

4.1 Introduction

In the previous sections, the current state of the computer industry and the probable directions of its future development have been discussed. Lifetime extension through reuse has emerged as a solution which has strong potential to alleviate the predicted future impacts of the sector. As a formal or strong informal market for these products has not developed, the barriers to the development of a secondary market for computers have been examined. Studies have shown that information asymmetry hinders the agreement of fair market value in the secondary market.

In this section, the use of signals in the used car market is presented as an example of a system which overcomes the problems of information asymmetry. The requirements for the development of a similar type of signalling for PCs are discussed, considering the essential differences between the two products, and the factors affecting reliability. The processes and signals currently in use by formal refurbishing centres for the evaluation of the viability of end of life PCs are examined, and the shortcomings of these processes considered in the context of ensuring the maximum possible proportion of viable used PCs are reused. The requirements of a process to allow the evaluation of viability while maintaining low cost to the remanufacturer, and consequently to the secondary user are also examined.

An overview of the solution proposed in this project is presented, in which LCM data collected throughout the lifetime of PCs are used to estimate the probable

reliability for the foreseeable future. This could potentially overcome many of the barriers to the determination of value based on the lifetime usage pattern and environmental factors. RFID is discussed as a technology which can be used for the extraction of this information at EoU, and the limitations of existing RFID technologies for this purpose are examined. Modifications to RFID technologies which would be necessary to overcome these limitations are also considered.

If the amount of LCM data stored in a device increases linearly in the course of the lifetime of a PC, the volume of data to be extracted at EoU could vary depending on the age of the system. With increasing data to be extracted, the probability of an error occurring in the extraction process increases. The amount of data storage required for a system like this should also be predictable to allow hardware specifications to be customised to the requirements of the function. As a result, it is important to attempt to record the LCM data in a way which preserves information about the depreciation of systems, while ensuring that the total amount of data remains within acceptable limits. Methods used in stress analysis of mechanical structures are presented, which have the potential to achieve this.

4.2 Signalling

Information asymmetry is considered to be one of the principal barriers to the development of a secondary market for used computers between consumers. The inability to reliably share information about the conditions in which a computer has been operating impedes the ability of parties to find balance points in the market, where buyers and sellers agree on a fair price. In order to solve the problems posed by information asymmetry, it is necessary to increase the confidence that buyers have in used computers. This is difficult to achieve while there is still a general lack of

knowledge among users regarding the factors which can lead to failures in PCs, and what the indicators of the likelihood of failure are.

The economist Michael Spence identified signalling as a method of improving sellers' chances in markets characterised by information asymmetry, such as the job market [1]. Signalling is a method by which a seller is capable of sharing private information about the condition of the product with potential buyers. This information must be trustworthy and presented in a format which is easily understandable to the average consumer as a metric of lifetime consumption. An example of a market in which signalling has gained widespread acceptance, and thereby overcome information asymmetry is the used car market [2], where a majority of purchasers are aware of the signs, such as the odometer and the service history, indicating a vehicle which has been well taken care of. By examining these simple signals, it is not necessary for a buyer to have a full understanding of the functioning processes of a car's engine to be able to determine to a reasonable degree of accuracy whether it can be expected to reliably operate for many years.

In order for a system of signalling of this type to become adopted on a wide scale, it is necessary for its adoption to be low cost to both manufacturers and secondary purchasers, particularly in a market like computers, where costs of new units are constantly decreasing, driving the value of used products still lower. The metric in use must reflect not only the initial value of the system and its age, but also the depreciation that has been caused in the system throughout its lifetime. This information must be presented in such a way as to be easily comprehensible to buyers and sellers of reasonable technological capability.

LCM and signalling are naturally compatible, as the information retrieved using LCM is designed to be a factor in determining MTTF [3]. While this metric doesn't necessarily give an accurate prediction of the time when a system will experience failure, it does provide a degree of certainty in predicting the likelihood of a failure in the immediate future. This would provide quantification of depreciation which has occurred as input to buyers and sellers in trying to agree upon a reasonable price for a product.

4.3 Triage

A large part of the cost associated with the formal processes for reuse of computers at present is related to the complex process of ascertaining whether an individual system is suitable for reuse. The process in one Irish refurbishment centre is to carry out a visual inspection, followed by a diagnostic process performed on the computer while it is operational. Computers which pass the diagnostics processes are then erased and a new operating system is installed, however no information is known about the condition of the hardware, beyond basic functionality. Additionally, the process of diagnostics takes time, regardless of whether or not the PC will eventually be suitable for resale, and each unviable PC which goes through this process increases the cost which must be recouped through the sale of viable systems, in turn increasing the cost of used systems which do eventually get resold. This in effect raises the bar of the minimum acceptable value of a computer to be resold, decreasing the total number which make the cut [4].

If a refurbisher or reseller of used computers had access to LCM data, and the capability of analysing it in the context of depreciation, the overhead associated with evaluating the potential of unviable computers could be reduced and an additional

metric of reliability could be introduced, increasing the confidence of the reseller or refurbisher in the quality of the goods offered. Triage in this context is named for the process in hospitals of grouping patients based on the varying degrees of severity of their conditions. In the context of PCs in a refurbishing centre, the process would involve grouping PCs based on the evaluation of their potential for reuse and remanufacturing. By performing a rapid initial triage process, whereby the systems with the least potential are separated, attention can be focused on those which are most likely to be suitable for reuse, and will result in the greatest profit. By thus reducing the costs associated with ultimately valueless systems, the profit to refurbishing centres can be increased, resulting in turn in an increase in the total number of computers which can viably be reused.

A key factor of the development of a signalling methodology for a triage purpose is the ease of extraction of data when a product reaches EoU. In a consumer to consumer market, information should be easily retrieved so that informal sales can be arranged without expensive or unnecessarily complicated data extraction procedures. In cars, retrieving data is a simple matter of observing the odometer and the vehicle service history; however in the case of computers the process must necessarily be more complicated, as it is difficult to imagine a method for externally presenting information about the manner in which a computer has been used throughout its lifetime, and the various factors which can impact on its reliability. In an industrialised setting such as a WEEE centre or a refurbishing plant, it is desirable to perform analyses on a large number of systems in a short period of time, so it is necessary to minimise the time overhead associated with the recovery of LCM data.

4.4 Proposed Solution

As a solution to the problems outlined, a system to perform a triage-type analysis of used computers is proposed. Because RFID is a wireless interface, in which the energy for the transfer of information is provided solely by the reader, it is an appropriate choice for the purpose of extracting this type of information as it is possible to retrieve information while a computer is powered off. The wireless features of the technology make it possible to perform analysis of computers easily on a large scale, such as would be necessary for an industrial scale refurbishing plant evaluating systems on a conveyor belt, or in small amounts using handheld readers.

Throughout the use phase of a PC the process of acquiring LCM data is performed while the computer is operational. Thermal sensors, located as standard within the processors of all computers, can report the current thermal conditions within a PC when queried. By using sensors which are already in place, it is possible to reduce the cost of the inclusion of new components in PCs, reducing the initial cost of implementation of this system for manufacturers. The information from these sensors is stored in a memory device located on a PCI card throughout the lifetime of a system, developing a thorough overview of the thermal environment in which the components have been operating throughout the life of a system. While further sensors could be integrated in order to improve the accuracy of LCM methodologies, these would represent additional cost for manufacturers, so would have to be evaluated in terms of the potential benefit in evaluating reliability. Further thermal sensors located around the motherboard could be used to develop a more detailed profile of the internal thermal layout of a case. As some notebooks integrate accelerometers into their hardware, these sensors could be used to detect shocks and drops which could affect the reliability of a system.

Using RFID, the memory device on which the LCM data are stored can be accessed at EoU, and the data can be retrieved. These data allow a profile of the usage patterns of an individual computer to be built up and used to evaluate the expected continuing reliability for that computer. Cross-referencing the information regarding the probability of imminent failure and the value of the hardware in the computer will provide a more accurate estimate of the true value of the used system. Sellers with access to the analysis of a used computer are therefore better informed about the quality of a system before committing to a price. An overview of the operation of the system is shown in Figure 1.

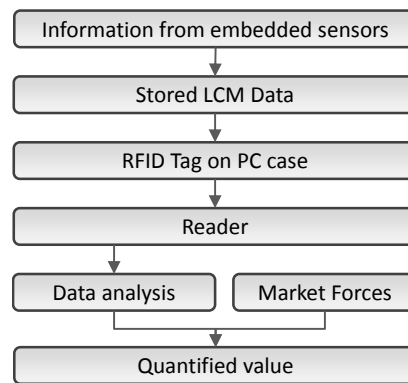


Figure 1: Overview of data extraction system operation

Because RFID cannot normally propagate signals through metallic surfaces, it is necessary for the RFID tag antenna to be located on the outside of the case of a PC. The motherboard of a PC offers a fascia which is open to the outside of the system, normally at the back. An RFID antenna can be mounted on the outside of the PC, and connected through the external-access fascia to the hardware located on the inside of the system. The hardware component of the system performs all of the necessary encoding of data and communication with a reader, as well as the analysis of data from the sensors, and performs data reduction on incoming environmental sensor information. A broad overview of the hardware located on the PCI card is shown in Figure 2.

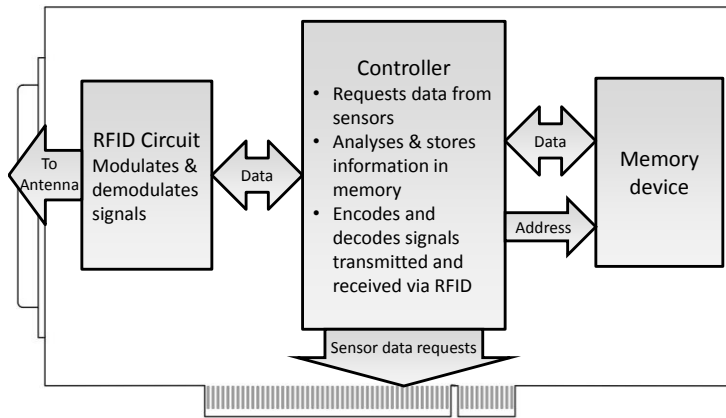


Figure 2: PCI hardware interactions

From the reader, data is transmitted to “middleware”, software which processes the information retrieved and analyses it to determine the patterns present and their impact on the reliability of a PC. This information can then be cross-referenced with current market values, based on trends in technology and the expected continued life of the used PC to give a value for the used system which accurately reflects its reliability, its functionality and the competing value offered by newer systems. Various potential reader configurations and the purposes for which they are suitable are discussed in detail in Chapter 7.

4.4.1 Prototype development

In order to demonstrate the potential for the development of this type of technology, a proof of concept has been developed for the technologically innovative section; the communications of large volumes of stored information from a RFID tag to a reader, and the streaming of this information to a computer for analysis. A volume of information can be transmitted in this way which would not be possible using currently available RFID technology.

In making decisions about which frequencies and types of RFID technology would be most suitable for the function of life consumption data extraction, the range, data rate, ease of development and tolerance of barriers were taken into consideration.

Low frequency technologies were considered to be too limited with regard to data rate, while the complexity of the antenna design required for the functioning of the far-field magnetic coupling made the UHF and microwave frequency ranges unsuitable as a prototype. Following adoption of the technology, it is likely that the advantages offered by far-field coupling would mean the eventual adoption of UHF technologies; however technological limitations in the process of creating and testing a prototype device precluded their immediate use.

High Frequency technology, operating in the 13.56MHz frequency range, is demonstrable and testable in a lab environment using standard oscilloscopes. Furthermore, microcontrollers operating at frequencies of up to 16MHz are relatively common and inexpensive. The design of simple RFID circuits by hobbyists is common at this frequency, meaning there is readily available literature about the process of coupling and circuit design.

Following the choice of HF technologies, it was necessary to decide on a protocol upon which to base the operation of a prototype device. Although the protocols currently in use do not allow for the transmission of the volume of data required, by basing the fundamental operation of the prototype on an existing protocol a large amount of the technologically complex process of initiating communication could be bypassed. Basing the prototype on existing protocols also allowed for the use of existing RFID reader technology without the need for excessive modification of the fundamental operating principles of the technology, so reducing the risk of the fundamental functionality of the interface being inadvertently affected.

The MIFARE protocol was selected the most appropriate protocol for designing a prototype tag. This was primarily owing to the availability of a user programmable

RFID reader, which would enable the modification of the firmware on the reader without affecting the low-level functionality. The MIFARE protocol features methods for the identification of a single tag type, multiple tag handling, and error handling by means of both the use of byte level parity bits and frame level cyclic redundancy checks. The error checking functionality is necessary in order to deal with the amount of data which would be transmitted using the prototype tag, as an increased volume of data increases the probability that an error will occur at some point during transmission. Tags normally associated with this standard are used as identification cards for such purposes as building access control. These are passive tags, and are used only within a short range of the reader.

The goal of reprogramming the reader is to retain all of the functionality associated with the protocol, allowing it to continue to recognise the common RFID tags and perform access control functions, while at the same time allowing the reader to communicate with and identify the newly designed tag prototype, and to modify the way in which data is handled so that the increased volume of information could be processed using the existing hardware. Information retrieved by the reader is then transmitted to a PC, which acts as the middleware in the system, analysing the thermal data for the relevant characteristics which would determine the amount of degradation which the system has undergone.

4.5 Advantages of Data Availability

Owing to the rate of technological improvement for PCs, and to the cost associated with such studies, there is not a large volume of data in literature relating to actual computer failures and their causes. A secondary advantage of the wide-scale development of a system for LCM of PCs is that through widespread adoption a broader

data set will become available, and it will be possible to develop a clearer understanding of the effects of different factors on the continued reliability of a computer, further improving the accuracy of any estimate of remaining life. This information could feed back into subsequent estimations of secondary market value, allowing a continually updated measure of the projected life span of a PC. In this way, the adoption of a system for widespread LCM of ICT would improve not only the number of PCs which could be reused, but would function as a large scale experiment into the complicated interactions between various environmental conditions and their effects on PCs. The prevalence of PCs and the rate at which they're replaced would provide a wealth of information regarding the means by which failure occurs.

In the case of some of the factors which would devalue a computer at EoU, their reduction during the first use phase would also reduce the environmental impact of the PC during the use phase. For example, a computer which has its air intakes cleaned occasionally requires lower fan speeds, therefore using less power throughout its life as well as improving the lifetime thermal profile of the PC and increasing its potential for reuse.

A clearer understanding among manufacturers of the usage and environmental patterns which tend to cause weakening and failure in PCs would result in the possibility of use phase feedback to users, creating behavioural change to increase the value of a PC in a secondary market and also possibly extending the first life of a system, before the occurrence of initial functional obsolescence. A potential result of implementing LCM in PCs is in the integration of a system of LCM with a user interface in the software of a computer, which could afford the opportunity to directly inform users of the estimated current value of their system, which environmental and usage patterns are currently determining this value, how these patterns can be affected,

and the projected effects of changing these patterns throughout the lifetime of the PC on its eventual value.

Such feedback to users would tend to improve the prevalent comprehension of the failure mechanisms of PCs, as well as introducing direct and tangible incentives to use a PC in a way which extends its life and increases its value in a secondary market. Introducing a popular understanding of the factors which govern secondary market value would mean the necessary motivation could be provided for PC owners and companies to ensure that they extend the lives of their PCs. In this way they ensure that they will receive the best possible price for their used PC at end of life, and also potentially that the use phase impacts are decreased throughout the lifetime.

Finally, increasing the amount of knowledge available about the causes of failure presents manufacturers with the opportunity to increase the reliability of their products through changes to manufacturing processes which directly target the sources of failure which occur most often in real-world situations. This could have the result of further increasing the reliability of computers, and increasing the potential for secondary market growth.

4.6 Data Reduction

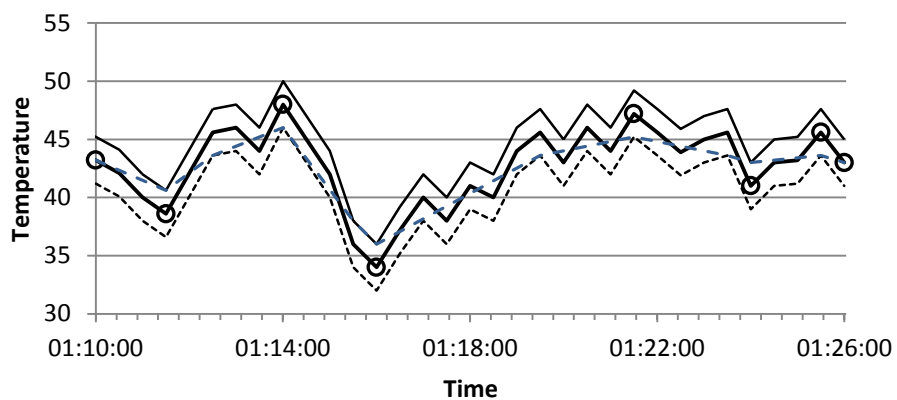
Throughout the lifetime of a computer, the amount of information provided by environmental sensors would increase in a linear fashion. This data must be stored in a memory device located within the PC. In order to reduce the size of memory required, the length of time to transmit data recovered, and the subsequent analysis which is performed on the data, it is possible to perform some analysis and processing of data prior to its storage on the memory device. A potential method for reducing the amount

of data required while preserving the record of depreciation which has occurred is to adapt methods in use in other areas for similar load characterisation and quantification.

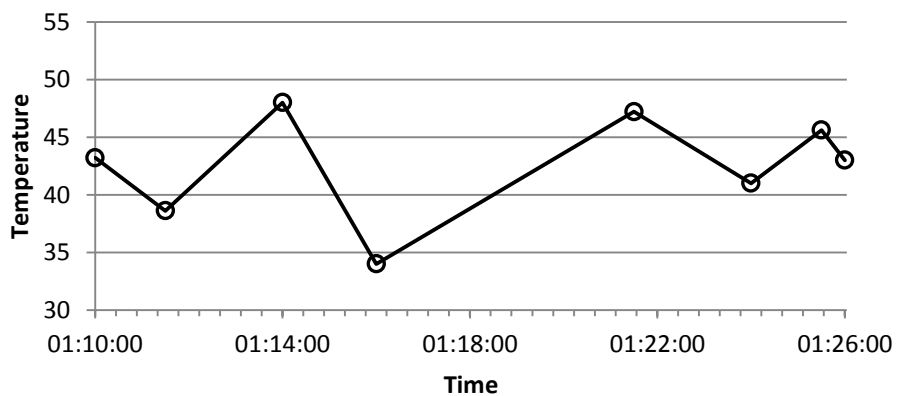
4.6.1 Ordered Overall Range

In order to reduce the volume of data for which processing is necessary, it is possible to express thermal data as a series of time dependent peaks and troughs, by eliminating trends which are not sustained or too small to take into consideration [5].

This is performed by specifying a range of values around the data in use. As long as a line can pass within this range without reversing direction; top to bottom or bottom to top, the values can be ignored. Whenever a trend reverses direction, the point is taken into consideration. The results of this form of data reduction are shown in Figure 3.



(a)



(b)

Figure 3(a): Sample data showing range of temperature samples, and (b): retrieved trend data

In this way, the trends of temperature may be expressed as only the larger overall peaks and troughs which have a significant effect on the reliability of a system. This essentially adds hysteresis to the system, ignoring temperature changes which fall within a selected range of the previous value, and recording only significant changes in temperature. With additional information on the effects of different periods and amplitudes of thermal cycles, the optimal range of the reduction can be found.

4.6.2 Rain flow Cycle Counting

For stresses such as temperature, the majority of the degradation effects are owing to changes in temperature over time, and to temperature cycles. As such, by counting the cycles which occur, and classifying their unique characteristics, it is possible to reduce the amount of information to be stored and transmitted. Rather than storing all of the sequential temperature data acquired from sensors, it is more efficient to store information in bins, with each bin representing a cycle of one specific mean, duration and amplitude [6].

Among mechanical engineers, the characterisation of stress loads is done in this way. Rain flow cycle counting operates on the basis of observing the cycles in stress and plotting the load versus time such that the time axis points downward and the load axis is horizontal. Then, by imagining rain dripping from the inside of each reversal in load, and observing the flow of the water, it is possible to characterise each cycle as it occurs. There are several rules which then determine whether or not a half-cycle is counted:

- i. If it starts inside a peak, and passes another peak of equal or greater amplitude, it terminates, and its duration and amplitude are recorded as a half cycle.
- ii. If a cycle initiates inside a trough, and passes another trough of equal or lesser amplitude, the same happens.

- iii. If a flow encounters water from a “higher” reversal, it terminates, and the flow from the higher point continues.
- iv. Half cycles based on peaks can be paired with half cycles based on troughs with similar amplitude to be recorded as complete stress cycles [6].

As these rules are not entirely intuitive, a demonstration of the methods is given in Figure 4, with each type of termination shown.

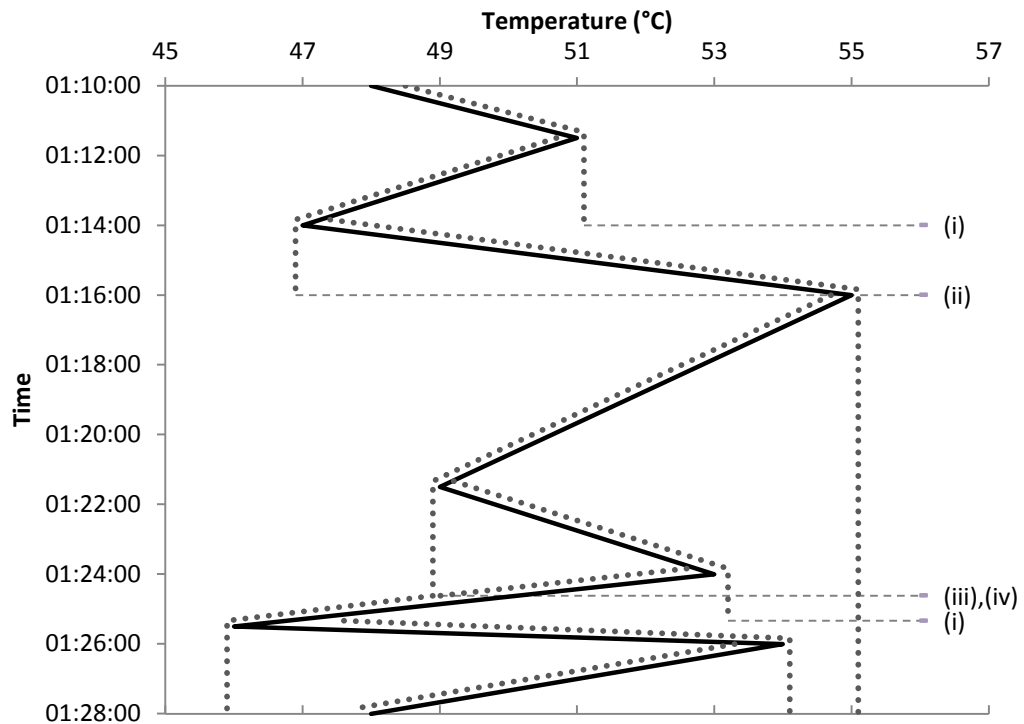


Figure 4: Rain flow cycle counting, demonstrating various half-cycle terminations

This approach is sufficient for mechanical processes, where the measurement of tension and compressions means that the average displacement will normally be zero. In the case of computers, however, it is important to also record the mean temperature, and the duration of the half cycles. As a result, a system for was developed which records three characteristics; mean load, load amplitude and duration of a cycle [7]. Each unique value may be assigned a discrete bin in memory, counting half-cycles of this specific mean, amplitude and duration.

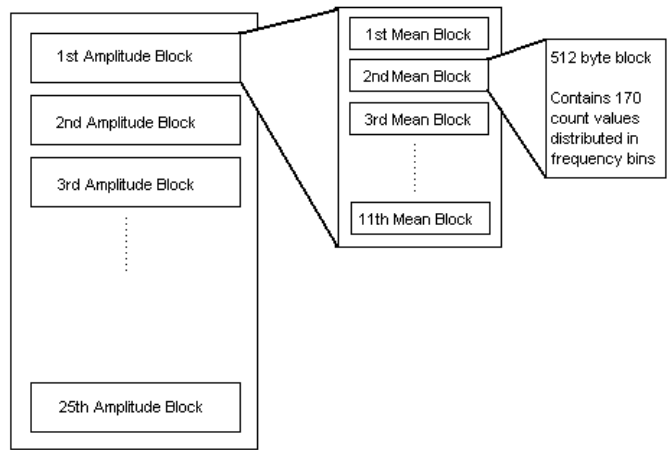


Figure 5: Data bin structure in memory

By storing this information in bins on a memory device, it is ensured that the amount of storage required for data will be a constant, equal to the total number of bins in use, multiplied by the size of each bin. The amount of storage required is then independent of the amount of time for which a computer has been operating, and of the size or frequency of the thermal cycles which it undergoes.

4.6.3 Implementation

The initial solution for rain flow cycle counting required 142 kB of memory to store the requisite binning structure of cycle amplitude, based on the following distribution.

- 25 amplitude bins, between 1°C and 25°C, with resolution 1°C.
- 11 mean bins, between 45°C and 75°C with resolution 3°C.
- 170 duration bins, between 30 seconds and 7000 seconds with resolution 10 seconds between 30 seconds and 400 seconds, and resolution 50 seconds between 400 and 7000 seconds.

Each duration bin consists of 3 bytes of data. The data structure is shown in Figure 6.

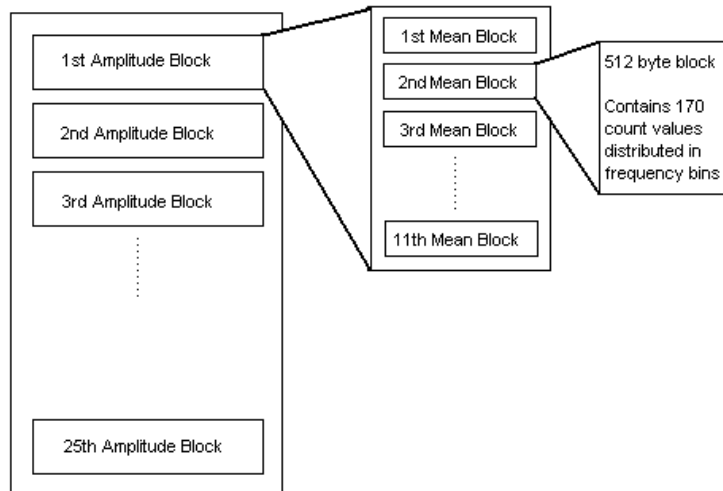


Figure 6: Initial structure of data bins in memory

Based on a binning structure requiring 142 kB compared to one byte of thermal data stored at thirty second intervals, the data required for linear storage would exceed that for processed data after 1183 hours of use, or approximately 50 days of eight hours per day. This efficiency is illustrated in Figure 7. Compared to the estimated lifetimes of PCs of 4380 hours [8] and 6000 hours [9], this value offers significant efficiency.

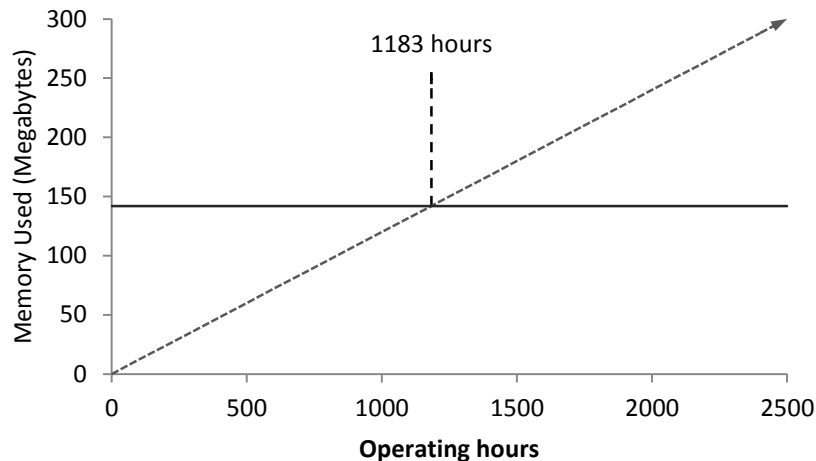


Figure 7: Data storage requirements: Rain flow Cycle Counting compared to Linear Storage

Analysis of real world temperature data led to the need to revise the bin structure in memory. Temperature was found to cycle about a wider range of means than were accounted for in the existing structure, and cycles were also found to have a wider range

of amplitude values. On examining the duration of real world cycles, some cycles were found to have much shorter duration than the 30 second minimum catered for in the above structure, and as these fast cycles could potentially have a significant impact on the reliability of a computer, it was important to record their occurrence.

The revised process for the storage of data takes account of the relative importance and higher incidence of the resolution of short slopes in temperature by spacing the bins across logarithmically increasing durations.

The resultant revised structure has the following distribution of values:

- Between 1 and 19 amplitude blocks depending on mean value, between 4°C and 76°C, with resolution 4°C.
- 37 mean blocks, between 27°C and 99°C with resolution 2°C.
- 30 duration blocks, between 5 seconds and 46198 seconds with the spacing of the blocks as shown in Table 1.

#	Range (s)	#	Range (s)	#	Range (s)	#	Range (s)	#	Range (s)
0	0-5	6	62-82	12	346-462	18	1948-2598	24	10955-14609
1	5-15	7	82-110	13	462-616	19	2598-3464	25	14609-19481
2	15-25	8	110-146	14	616-822	20	3464-4620	26	19481-25979
3	25-35	9	146-195	15	822-1096	21	4620-6161	27	25979-34643
4	35-46	10	195-260	16	1096-1461	22	6161-8215	28	34643-46198
5	46-62	11	260-346	17	1461-1948	23	8215-10955	29	46198+

Table 1: Distribution of duration values in duration blocks

This structure of data storage both expanded the range of cycles which were being stored and reduced the volume of memory required. The reduced memory requirements are equivalent to only 266 hours of continuous storage of temperature data, compared to the 1183 hours required for the previous implementation. The process for implementing the data reduction on hardware is detailed in Section 5.6.

4.7 Conclusion

In this chapter, a process for overcoming the barriers presented by adverse selection in the secondary PC market is presented. The goal of the system is to stimulate the growth of the secondary PC market by reducing the financial and labour-related cost of the evaluation of the viability of used PCs for sale in a secondary market.

In order to classify PCs based on their potential for reuse, the factors determining depreciation of PCs must be taken into account. Life consumption monitoring techniques offer the potential to quantify the damage which occurs in a PC owing to environmental factors throughout its lifetime. Prognostics can then be performed on the basis of data relating to environmental and operating conditions for the PC. Critically, for temperature, which is the factor most likely to affect the reliability of the PC, sensors are already in place. Sensors which could monitor additional environmental loads are readily available, and simple and cost effective to integrate into current PC architectures. In order to perform prognostic health monitoring of PCs it is only necessary to monitor and record the information from these sensors throughout the lifetime of the PC.

A modification of RFID technology offers the capability of recovering LCM data from PCs at EoU without the need to have the systems powered on, thereby enabling the large scale separation, or triage, of PCs based on their potential for reuse. Among the various frequencies and types of RFID, high frequency was selected as the candidate with the best all-round performance for this application, although because of the relatively low data rate facilitated by HF RFID, the implementation of the solution using this technology would require simplification of the lifetime data before storage, and modification of existing RFID protocols.

A system implementing this type of monitoring would involve the sensors located in or around the components of a PC communicating with a device on the motherboard. The function of this device is the aggregation and simplification of data throughout the lifetime of a PC, and its storage in a memory device. A microcontroller connected to the memory device will, at EoL, communicate the stored data to a remote RFID reader through an antenna located on the outside of the PC by implementation of the necessary protocols. From the reader, data can be transmitted to middleware for analysis and a value can be allotted to the PC based on the various attributes monitored in combination with market-related inputs.

To demonstrate the viability of this process, in particular the RFID aspect, a prototype has been developed which implements the necessary storage and transmission of thermal data using a PCI card connected to the motherboard of a PC. The following chapters detail the development of the process for RFID communication, allowing the extraction of data and its transmission to middleware.

4.8 References

- [1] M. Spence, "Job Market Signaling," *Quarterly Journal of Economics*, vol. 87, pp. 355-374, 1973.
- [2] G. A. Akerlof, "Market for Lemons - Quality Uncertainty and Market Mechanism," *Quarterly Journal of Economics*, vol. 84, pp. 488-500, 1970.
- [3] P. Lall, M. Hande, C. Bhat, J. Suhling, and N. Islam, "Prognostics and health management of electronics," *IEEE CPMT: International Symposium and Exhibition on Advanced Packaging Materials: Processes, Properties and Interfaces*, pp. 148-148, 199, 2006.
- [4] "Electronics Re-Use and Recycling Infrastructure Development in Massachusetts," United States Environmental Protection Agency 2000.
- [5] C. Lalanne, *Mechanical Vibration & Shock: Fatigue Damage* vol. IV, 1999.
- [6] S. D. Downing and D. F. Socie, "Simple Rainflow Counting Algorithms," *International Journal of Fatigue*, vol. 4, pp. 31-40, 1982.

- [7] A. Ramakrishnan and M. G. Pecht, "A life consumption monitoring methodology for electronic systems," *Ieee Transactions on Components and Packaging Technologies*, vol. 26, pp. 625-634, Sep 2003.
- [8] E. D. Williams, R. U. Ayres, and M. Heller, "The 1.7 kilogram microchip: Energy and material use in the production of semiconductor devices," *Environmental Science & Technology*, vol. 36, pp. 5504-5510, Dec 15 2002.
- [9] S. B. Boyd, A. Horvath, and D. Dornfeld, "Life-Cycle Energy Demand and Global Warming Potential of Computational Logic," *Environmental Science & Technology*, vol. 43, pp. 7303-7309, Oct 1 2009.

Chapter 5

METHODOLOGY: SOFTWARE DEVELOPMENT

5.1 Introduction

In the design of the working prototype, the most challenging aspect was the RFID system allowing the wireless transmission of life consumption information from a PC to a remote communications system. This required implementing the existing ISO/IEC 14443 standard for contactless integrated circuit cards, and the modification of a commercially available MIFARE RFID reader to allow the transmission of greater volumes of data than would normally be possible.

In order to implement these requirements, it was necessary to first design a circuit which would behave exactly in the same manner as a proximity integrated circuit card (PICC) from the perspective of the reader (proximity control device, or PCD), but also be capable of transmitting a much greater volume of data.

As the circuit utilises existing communications protocols, it was necessary to be familiar with the specifications governing the communications between the devices. These are laid out in the ISO 14443 standard, which defines the physical characteristics, the radio frequency power and signal interface, initialisation and anti-collision processes and the transmission protocol. The goal of the design was to conform as closely as possible to this standard in order to demonstrate that the transmission of large volumes of data is possible to existing RFID technology, and only require minor alterations to the current operating parameters.

5.2 Modulation

The ISO/IEC 14443 standard specifies a different type of modulation and encoding to be used in communications in each direction between the PICC and PCD. As the technology is based around the assumption that power for the tag will be provided by the reader, the format of modulation ensures that the reader is transmitting power for a maximum possible proportion of the time during communication. This allows the PICC to gather enough of the power through inductive coupling to be capable of responding.

5.2.1 PCD to PICC

In communications from reader to tag, the modulation in use is 100% amplitude modulation (AM). However it is necessary for the PCD to provide enough energy to allow the PICC to respond, so the communication is encoded so that information may be transmitted with minimal disruption to the carrier wave. In this manner, the energy required to enable PICC circuitry operation may be transmitted entirely from the PCD, and the pauses in the carrier signal, which are required to communicate data, are sufficiently short to ensure that a PICC remains energised throughout the communications. The bit representation sequences as defined in the ISO 14443 standard are described in Table 2, and the standard definition for the “pause” transmitted is shown in Figure 8.

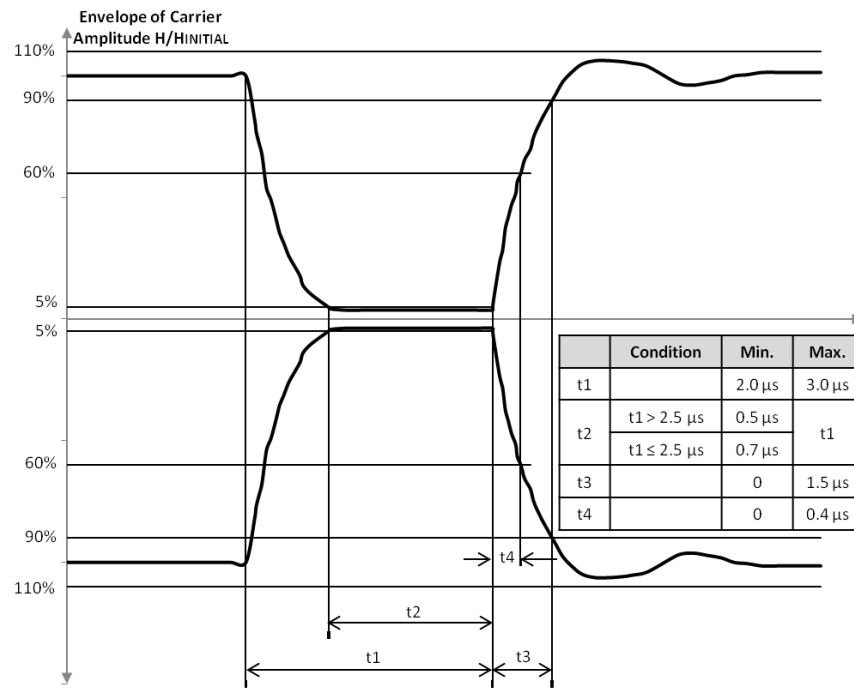


Figure 8: ISO 14443 definition for "pause" in communication

Sequence X	After a time of $64/f_c$ a "pause" shall occur
Sequence Y	For the full bit duration ($128/f_c$) no modulation shall occur
Sequence Z	At the beginning of the bit duration a "pause" shall occur

Table 2: Signal sequences for PCD to PICC communication defined in ISO 14443 standard

In order to differentiate between the transmission of signals, the provision of energy for response from the PICC and the absence of a PCD entirely, the sequences from Table 2 are used as described in Table 3. Specific combinations of sequences are also used to identify the start and ends of communications. In this way it is ensured that the absence of data can easily be distinguished from the transmission of a continuous sequence of '0's.

Logic “1”	Sequence X
Logic “0”	Sequence Y with the following two exceptions <ul style="list-style-type: none"> • If there are two or more contiguous “0”s, sequence Z shall be used from the second “0” on • If the first bit after a start of frame is “0”, sequence Z shall be used to represent this and any “0”s which follow directly thereafter
Start of communication	Sequence Z
End of communication	Logic “0” followed by sequence Y
No information	At least two sequences Y

Table 3: Information coding for PCD to PICC communications

5.2.2 PICC to PCD

Communications between the PICC and the PCD are performed by means of increasing or decreasing the load applied to the RF field generated by the PCD. This in turn affects the magnetic field intensity, which is detected at the PCD, and decoded to recover the transmitted information.

These communications are also encoded, using a form of Manchester encoding whereby modulation is applied during the first half of a bit period if the bit represented is a ‘1’, or in the second half if the bit is a ‘0’, as shown in Table 4. Communication is always preceded by a ‘1’, and the modulated portion of each communicated bit starts with the load applied. Table 5 shows how information is communicated using the sequences defined in Table 4.

Sequence D	the carrier shall be modulated with the subcarrier for the first half (50%) of the bit duration
Sequence E	the carrier shall be modulated with the subcarrier for the second half (50%) of the bit duration
Sequence F	the carrier is not modulated with the subcarrier for one bit duration

Table 4: Signal sequences for PICC to PCD communication defined in ISO 14443 standard

Logic “1”	Sequence D
Logic “0”	Sequence E
Start of communication	Sequence D
End of communication	Sequence F
No information	No subcarrier

Table 5: Information coding for PICC to PCD communications

5.2.3 Error Checking

The ISO 14443 standard defines error checking protocols which ensure the validity of information communicated in both directions. Odd parity bits are appended to each byte of all communications with greater than eight bits length, and for standard communications two cyclic redundancy check (CRC) bytes are appended to each frame. The polynomial for the calculation of the CRC bytes is $x^{16} + x^{12} + x^5 + 1$, and the initial value is defined as 6363 hexadecimal.

5.3 Physical Communications Interface

At the physical level, it is not necessary to be concerned about the function of the data which is being transmitted, and only the methods of communication need to be considered. For the purpose of the RFID tag and reader, the physical layer is implemented in the hardware which performs the modulation and demodulation of the transmitted signals, and by the software on the PICC which encodes and generates the outgoing signals, and decodes those incoming signals into simple digital data, which may then be passed to the higher levels of the code, where the resultant data is analysed and appropriate responses generated. This section examines the process for the development of software which implements the requirements of the physical interface.

Owing to the requirements of the standard governing RFID communications, as explained in section 5.2, it was necessary for the hardware code which affected the physical transmission of data to be written in assembly language. This was preferable to

writing it in C because, while C may simplify the programming process, the compiler performs optimisation of any C code entered, which makes it impossible to predict exactly the timing of the final implementation which is programmed to the microcontroller.

Given a clock at the carrier frequency, 13.56MHz, transmissions from the PICC during the application of modulation change every eight system clock periods, which is a narrow time frame in which to execute the intervening commands. The advantage of programming in assembly language is that each command in assembly has a specific execution time, defined in numbers of clock periods, which is described in the datasheet of the device in question. This allows a code writer to determine, by counting commands and considering the clock frequency, exactly the length of time which will pass between the alterations of the voltages on the output pins, guaranteeing that the signals from the device will conform as closely as possible to those laid out in the communications protocol.

5.3.1 Software Solution

Initially, the possibility of performing encoding, decoding and error checking using assembly code was considered. This solution was originally dismissed as it was thought to require too much assembly coding. An initial software solution was therefore developed in which the transmission of data was performed with assembly language. This solution is included in Appendix A, along with an analysis of the results of initial simulations which were carried out on the code. After examining the cost in processing time and memory of using C code to perform these functions, it was found that the time required to prepare data for transmission exceeded the maximum permitted delay between communications from the reader and reply from the tag. As a result it was decided to revisit the possibility of performing basic error checking, encoding and

decoding using assembly code functions owing to the superior timing efficiency of assembly code. The full solution is given in Appendix A including analysis of the relative efficiencies of the different implementations.

There were several advantages to the ability to encode the data as it was transmitted and to decode data as it was received over the previous solution - which stored samples sequentially, and subsequently analysed the resultant data to recover the information which was transmitted. Primarily, the amount of time required to analyse data is reduced drastically, as incoming data is analysed in real time as the bits arrive at the input pin. This allows the program to operate within the requirements laid out in the standard outlined.

5.3.2 Encoding

Encoding of data in the program was performed by the `asm_TX` function. The purpose of the encoding section of the assembly code was to transmit information in the manner required by the protocol, so that it could be received and decoded by a MIFARE RFID reader. The code for transmission of data operates by taking information which is stored in the `IObuffer` array. In `IObuffer`, the total number of bits to be transmitted is stored in the first address in the array, and all sequential bytes contain the information to be transmitted.

Owing to the fact that the modulation frequency for transmitting responses to the reader was one sixteenth of the carrier frequency, it is necessary to switch the load on the input waveform once every eight system clocks. Given that the assembly command to alter the signal on an output pin requires duration of two system clocks to operate, this only allows six system clocks of free processing time between each event during the modulated half of each transmitted bit. As such, all calculations relating to

subsequent bit transmissions are performed during the unmodulated half of each bit, which allows 64 clocks in which to perform the necessary operations.

During the development of the code for transmission of information further issues arose. Any viable code for this purpose has to follow several ‘branches’ depending on what is required. These include two normal branches, appending a zero or a one, and several more complicated branches, such as appending a parity bit, or accessing the next byte for transmission. While none of these processes are necessarily impossible, it is vital that each branch within the code takes exactly the same length of time, which makes the process very exacting, as changing any branch has a knock-on effect on any successive code, and it is necessary to ensure the cumulative time taken between the output events is consistent.

It is possible to describe the three main operations to be performed each time the PICC transmits a bit in normal operation. These are best described by observing Figure 9. If the next bit to be transmitted is the same as the current bit, as shown in Figure 9(a) and (d), then one modulated half bit must occur before the subsequent unmodulated half-bit. If the sequence is “01”, then two modulated half bits occur with no unmodulated time in between. Finally, if the sequence is “10”, nothing needs to be done before the next bit, as there are two consecutive unmodulated half bits, as shown in Figure 9(c). While the signal is being modulated, a counter monitors the number of modulations which have occurred. This counter is loaded with the required number of modulations by considering the current bit and the next bit as described above.

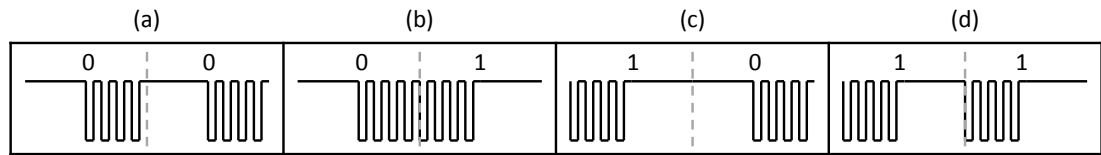


Figure 9: Possible bit sequences at output of microcontroller

Each of these processes is relatively quick. Adding a zero was found to take only six clock periods, while adding a one takes eight. By putting these processes towards the end of the unmodulated half bit, it is possible to add the remaining required processes before them, and to ensure that each process is started at the same time regardless of which branch it is being accessed from.

Figure 10 is a detailed examination of the operation of the section of code governing transmission of data. As described in section 5.2.2, each communication from the PICC is required to start with a ‘1’, so the initial transmission of data is four sequences of modulation. Following this, there are three paths, depending on which of the following actions is being performed:

- a) Transmitting normally
- b) Transmission has reached the end of a byte and needs to append parity
- c) Transmission has appended the parity bit and needs to fetch the next byte from the IObuffer

The decision is made by observing the “bits remaining in byte” counter, and if it is 0, then a parity bit must be added. If the counter is less than 0, an additional byte should be fetched from memory, and the counter should be reset to eight bits.

Concurrently, each time a bit which is not a parity bit is transmitted, the “overall bits” counter is decremented. This is initially loaded from the first address in IObuffer, so when it reaches 0, all of the required information has been transmitted by the PICC.

At this point, if the final bit was a '1', the function can exit immediately, as the modulated half bit has already been transmitted, while if it is a '0', a final modulated half-bit needs to be transmitted before exiting. Figure 10 details the decision process for the transmission code.

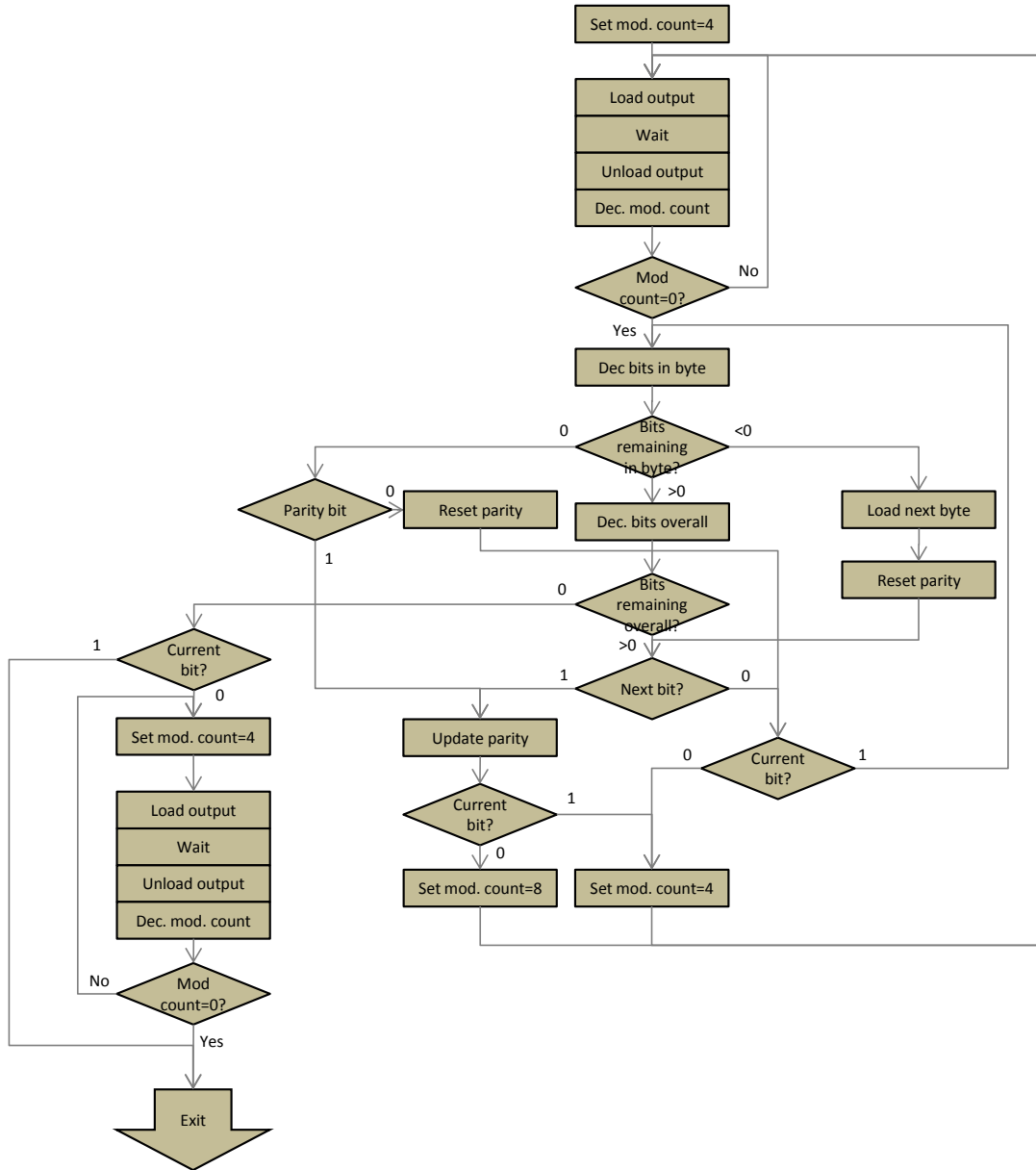


Figure 10: Flow diagram outlining operation of transmission section of assembly code

5.3.3 Decoding

As with the encoding section of the solution, the decoding function, `asm_RX`, performs the functions of receiving the information being transmitted by the PCD, decoding the signals and storing the information received in the `IObuffer`.

Because the number of samples in a bit is known, it is possible to write code which, at the end of eight samples, decodes the bit which has been received. This is done by counting logic high signals. When a transition from low to high is registered, the count is reset. As long as the input remains high, the count is incremented. The hardware involved is examined in detail in section 6.3.1, and performs the function of converting the carrier frequency, when transmitted, to a 'high' logic level, and the pauses to a 'low' logic level.

By examining the definition of a "pause" as shown in Figure 8, it can be seen that between the initial deviation from 100% of the carrier amplitude and the return to 60% of the maximum amplitude after the "pause", a minimum of $2\mu\text{s}$ and a maximum of $3.4\mu\text{s}$ passes. Allowing for the best and worst cases and taking into account a sampling period of $1.18\mu\text{s}$, a minimum of one and a maximum of three low samples must be recorded during the pause. From Table 2, a pause will begin either at the start of a bit period or half way through the bit. Allowing for the best and worst cases above:

- a) If three or less consecutive high samples are recorded before the end of a bit, the sequence transmitted corresponds to a sequence X, meaning a logic '1' was transmitted
- b) If between nine and eleven consecutive high samples occur before the end of a bit, the sequence was the last one to three samples in the preceding sequence X, and the eight high samples of a sequence Y, representing a logic '0'

- c) If between five and seven consecutive high samples are recorded before the end of a bit, sequence Z was transmitted, meaning either the start of communication or second or subsequent contiguous logic '0'
- d) If greater than 12 high signals are received, the signal is either a sequence Z followed by a sequence Y or two consecutive sequences Y, in either case denoting the end of the data communicated.

In the implementation of these rules in the code if one, two or three consecutive samples are found to have been high at the end of a bit, then that bit is stored as a '1'. If greater than four successive samples were high, it is recorded as a '0', unless greater than 12 sequential samples were high, as this is the indication of the end of transmission.

While the function sequentially receives and decodes bits, it rotates them into a temporary memory address, `temp`, and calculates the parity for the current byte. On receiving the ninth bit since the start of a byte it tests the parity bit to ensure that it is correct, and stores the contents of `temp` in the `IObuffer` at the next location. If the parity bit is not correct for any received byte, the function terminates. A flow diagram of the code execution is shown in Figure 11.

At the end of transmission, the last two bits received are discounted as they are the two end bits, and the total number of bits received is stored in the first address in `IObuffer`. At this point the amount of information referred to by the first address in `IObuffer` should be stored in the following addresses.

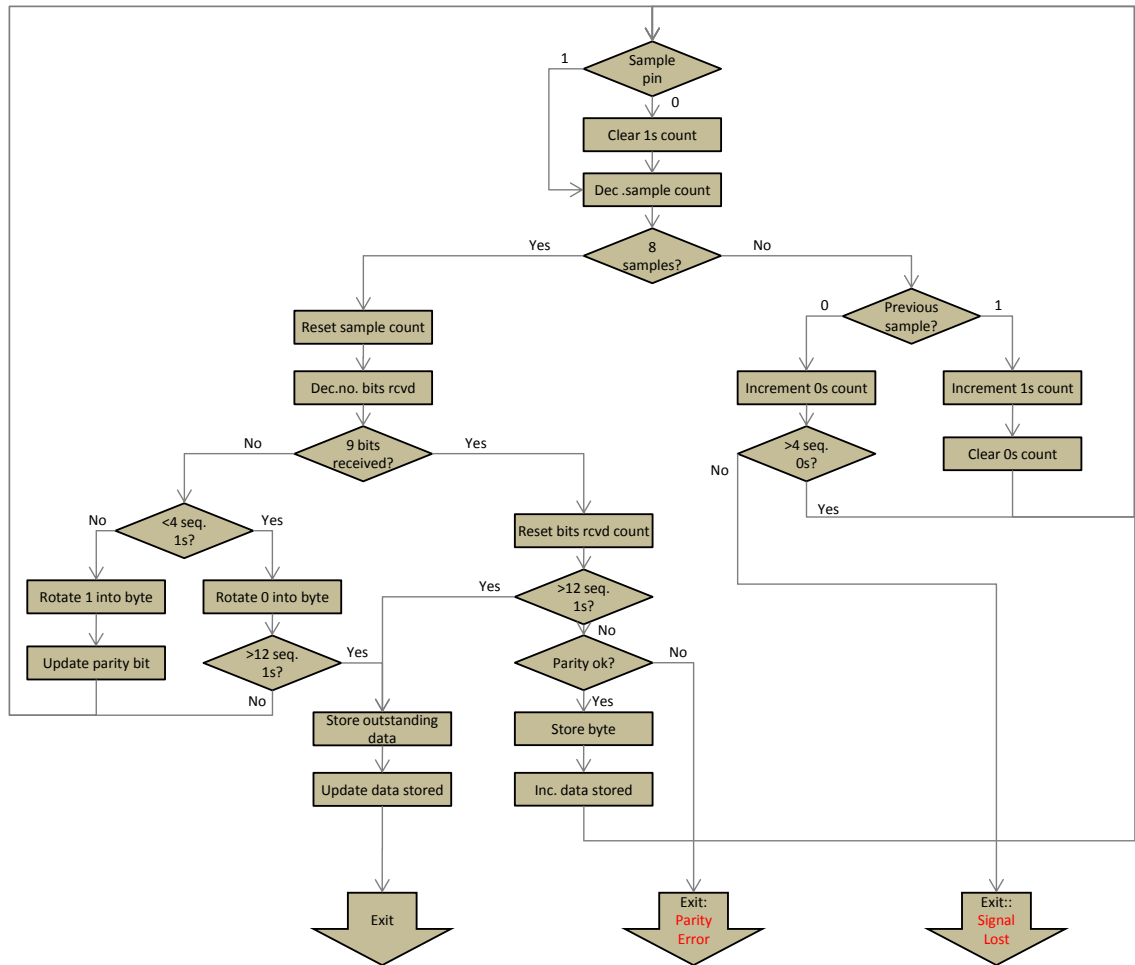


Figure 11: Flow chart for receiving and decoding of data

5.4 State Machine Implementation

The ISO 14443 standard describes the higher operations of the PICC and reader in the form of a state machine. This model significantly simplifies the protocol from a design perspective, as the prototype may be programmed to remain in any individual state until such time as it receives specific commands. Once these commands are decoded, the PICC can move onto the next state in the sequence, depending on the exact command received.

It was decided to use the state machine as the model for the functionality of the main part of the PICC program. In this way, as the PICC receives commands from the PCD it is only necessary to compare them to a predefined selection of commands. For

each state, only certain specific commands will allow progression to the sequentially following state, while a different selection of commands will cause the PICC to either reset or to go into a ‘Halt’ state. A synopsis of those commands which are defined in the program is presented in Table 6. Any command not catered for in a particular state will be ignored by the PICC.

Abbreviation	Command	bits	Value
REQA	Request Command, Type A	7	26 _h
WAKE-UP/WUP	Wake Up	7	52 _h
ATQA	Answer to Request, Type A	16	Proprietary
SEL	Select Command	8	93 _h : Select cascade level 1 95 _h : Select cascade level 2 97 _h : Select cascade level 3
NVB	Number of valid bits	8	
HALT	Halt	32	50 _h +00 _h +CRC

Table 6: Synopsis of commands used in state machine

For these more complex commands, governing data recovery and analysis, the higher programming language of C was used. This was possible as the ISO 14443 standard allows generous maximum delays between transmission and receipt of blocks of data. Between the initial activation command for a device and its response, exactly 960 clock periods are required, or a total of 70.8µs. Between all other PCD transmissions and PICC responses, 960 clock periods is defined as the minimum response delay, while the delay is permitted to be anything up to almost half a second, or over 6.5 million clock periods. This allows ample time for complex operations to be performed in the code without concern for the exact timing of responses.

Responses are only required to occur at a time equal to an integer multiple of bit periods since the last bit transmitted by the PCD. During inter-transmission delays it is possible to perform analysis on the data received and prepare the appropriate next data blocks for transmission without exact knowledge of the number of clock periods

required. While it was not necessarily certain to what degree the optimisation performed by the compiler would affect the timing parameters between input or output events, once the program was implemented on hardware it was possible to add and modify delays to the code so that the requirements of the protocol were satisfied.

The following is an overview of the states outlined in the ISO 14443 standard and the signals from the reader which result in the progression to subsequent states. In order for communication of stored information to occur, the PICC must enter the active state by means of progression through all of the preceding states.

5.4.1.1 State 1: Idle

In this state the PICC is inactive until it encounters the field of a PCD, which gives it the power necessary to process commands. On detecting a PCD in range, the PICC enters receiving mode, and will remain there until such time as a transmission is received from the PCD or the PCD goes out of range.

In order to exit the idle state, the PICC must receive either the seven bit request acknowledge (REQA), or wake up (WUP) command. On receiving any communications from a PCD, the PICC compares the information received with these values. If the value is as expected, the PICC automatically transmits the response, ATQA, and transitions to the next state. Should any other command be received, the comparison will return a 'false', and the PICC will return to receiving mode.

5.4.1.2 State 2: Ready

After receipt of a REQA or WUP command, and transmission of the ATQA, the PICC enters an anti-collision state, in which the PCD selects a unique PICC if there is more than one in its range. For this reason, the ATQA must be transmitted at a specific time by the PICC, so that all of the returned ATQAs in a field occur at the same time. A

collision occurs when two PICCs transmit different bits at the same time. Since a '1' and a '0' are encoded as the modulation of opposite halves of a bit period, the simultaneous transmission of two opposing bits is interpreted at the reader as a single bit period of constant modulation. While this application does not take into account the possibility of multiple tags implemented in a single reader field, the anti-collision protocols were implemented in code, so that at a later stage the functionality could potentially be added if deemed useful.

The PICC expects to receive a select (SEL) and number of valid bits (NVB) command, consisting of two bytes, which will vary through the anti-collision loop. Following successful receipt of this command, as part of the anti-collision loop, the PICC will transmit its complete 32 bit unique identification number (UID). In normal operation of the circuit, where no collision occurs during this communication owing to multiple PICCs in the reader field, the reader will respond with a modified SEL and NVB command, acknowledging the number of bits correctly received. Depending on the length of the UID of the PICC there can be several rounds of communication to transmit one complete serial number. The minimum length of a PICC UID is four bytes, and the maximum is 10. For the purposes of prototype development, it was decided that the minimum UID size should be used, so only a single frame of communication in each direction is necessary.

Following accurate receipt of the complete PICC UID, the PCD transmits a response containing the complete UID back to the PICC, combined with the SEL command. If the UID and the SEL command are correct at this point, the PICC responds with the select acknowledge (SAK) command, which is a single byte followed by two CRC bytes. At this point, the PICC enters the active state, and is prepared to deal with the duplex communication of data.

At any point during the ready state, if the PICC receives the REQA, WAKE-UP or HALT commands, it will revert to the idle state, and must go through the selection process again.

5.4.1.3 State 3: Active

In the active state, the PICC is capable of both receiving and transmitting data. This is the only state in which requests may be made for the PICC to perform a variety of operations. In normal operation of a PICC, these operations may include reading the data or writing information to any writeable memory on the card. This is where the implementation of the extraction of the thermal data stored on the prototype takes place.

The information extraction is requested by transmission of a proprietary command from the PCD. As this is additional to the normal function of both the PCD and the PICC, it required the modification of the PCD software, as well as its implementation on the PICC. The command was defined as CA_h , and on receiving this command byte from a PCD, the PICC begins the continuous transmission of data stored in the 512 byte on-board EEPROM. After each frame of one protocol byte and 15 data bytes, the PICC awaits confirmation from the PCD, and if confirmation is received correctly the subsequent byte is transmitted.

In order to verify that the amount of data specified in section 4.6.2 can conceivably be transferred over the PICC to PCD interface, the PICC is designed to cycle through the memory addresses repeatedly, transferring the complete contents of the EEPROM continuously over the course of 1000 package transmissions. This represents 15kB of data; about one tenth of the amount required given the data reduction processes used, and was considered to be an appropriate starting point to evaluate the performance of the circuit.

5.5 Simulation

Early in the development process of the PICC, simulations were used to monitor the effectiveness of the program. By simulating the execution of the code, it is possible to step through the operation of the device and examine how it responds to real world situations. The Atmel AVR Studio 5 software allows operation of a virtual ATMega16 to monitor all of its input and output (I/O) pins and to simulate inputs to ensure that the performance of the device is as expected. In simulation mode it is possible to halt the operation of the virtual chip at any command and observe the state of any I/O pin, as well as the state of any variable in the code. This makes it possible to see that the timing of the initial implementation of the code did not conform to the requirements of the protocol, and also facilitated debugging early code to ensure correct operation.

Figure 12 shows the interface for debugging the simulated microcontroller. In the top left is the code currently executing, while the top right shows the input and output pins, and allows the user to simulate inputs. At the bottom left is the watch list, containing the current value of the variables in the code, and in the bottom right is the current clock and the register values, important in the execution of the assembly code.

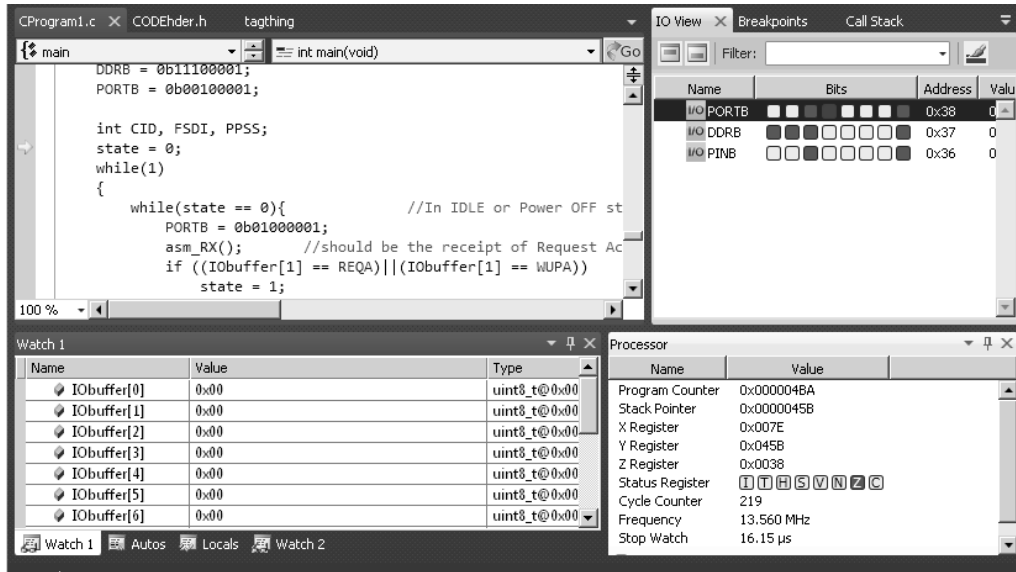


Figure 12: AVR Studio 5 debugging interface

Although simulation is vital to the design process, it became very time intensive, as it is necessary to enter each command in sequence as it is expected by the PICC in order to ensure that each response was correctly handled. This requires inserting break points in the code when sampling is taking place, and setting the input pins to the relevant values for each sample. This is time consuming, as execution necessarily stops at each sample regardless of whether or not the input is changing, and any deviation in the manual input of the sequence of samples from what is expected results in the receiver recording an incorrect value, and requires resetting the simulation. As a result, the sections of code which occur later in the operation of the program can take days of debugging to reach by this method.

Similarly, the output can be monitored to ensure the correct signals are being transmitted from the PICC, but this is also time intensive, as the output is modulated, and observing changes in the output requires pausing operation of the program at each change, and observing the time which has elapsed since the previous change. By stopping the program and noting the sequence of modulated and unmodulated half-bits which are being produced, it is possible to reconstruct the information being transmitted

to ensure the PICC is operating correctly, but because the signal at the output changes a total of eight times for each bit which is transmitted it is necessary to keep track of a huge number of events in order to reconstruct the exact data which is transmitted by the microcontroller.

During the design of the assembly code, this method of code debugging is desirable, as it allows observation of the effects of each operation on the various registers and flags of the ATmega16. After the assembly language had been completed, however, and it had been ascertained that information could be transferred successfully, debugging the timing features of the program required automation in order to avoid highly repetitive analysis.

In order to overcome this problem, it was necessary to set up stimuli in advance of running the simulation. This automatically alters the inputs and outputs at the relevant times in order to facilitate much faster simulation of the program. Unfortunately, AVR Studio 5 does not support preloading of stimuli or logging of outputs. Further investigation revealed that AVR Studio 4, the previous version of the compiler program, does support stimuli and logging.

5.5.1 Stimuli

In the AVR Studio 4 software, inputs to any port can be pre-set to occur at specific times through the use of a stimulus file. Each file relates to only one eight bit port on a device, and a specific format of data is required in the file to ensure that it is interpreted correctly.

Figure 13 shows the layout of a stimulus file. The first set of numbers is the clock number at which the input should change. The number after the colon is the hexadecimal representation of the eight pins of the port in question. These are numbered

from seven down to zero. Since the only pin in use as an input is Pin 4 of Port B, the input alternates between '10' when Pin 4 is set and '00' when all pins are cleared.

1	000000196:10
2	000067800:00
3	000067827:10
4	000067928:00
5	000067955:10
6	000068120:00
7	000068147:10
8	000068312:00
9	000068339:10
10	000068504:00
11	000068531:10
12	000068632:00
13	000068659:10
14	000068824:00
15	000068851:10
16	000072916:00
17	000072943:10
18	000073108:00
19	000073135:10
20	000073300:00
21	000073327:10

Figure 13: Layout of stimulus file for AVR Studio 4

While this significantly improves the speed at which repetitive processes can be performed, it also requires the timing for each command to be entered manually. This is quite time consuming, and changing the starting time of one command requires recalculating and re-entering the time for every subsequent change in the inputs. Added to this, as the inputs to be included represent the modulated data incoming from the PCD, rather than simple digital commands, it is necessary to calculate the pattern of modulated bits for each input to the PICC. Parity bits and CRC bits must also be calculated and included. The process is therefore highly calculation-intensive, and as the timing for each value determines every subsequent one, it is highly susceptible to human error.

In light of this fact, a spread sheet was developed in Microsoft Excel which is capable of determining the sequence of stimuli for any given command. It is also capable of varying the length of a 'pause' to be any value in the permissible range defined in ISO 14443, and of starting a command at any given clock.

The spread sheet functions by alternating the commands stored in vertical cells. As each 'pause' can reasonably be expected to be of more or less uniform length, every second timing line would simply be the previous time with the length of the pause added. The entire sheet works by starting from the least significant bit, calculating the pause to the next change in input, and then erasing the bit already handled. In this way, the same formula can be used in each calculating cell, and the sheet can be extended to calculate the timings for a command of any length.

As the first bit transmitted is a 'Z' signal for any command, the first bit is always predefined. Subsequent timings are decided on the basis of the following formula, where A8 is the cell currently under examination, and B9 is the cell containing the start time of the previous pause:

```
=IF(LEN(A8) = 1, "", IF(LEFT(A8, 3) = "101", B9 + 256,
IF(LEFT(A8, 3) = "100", B9 + 192, IF(LEFT(A8, 2) = "01", B9 + 192,
IF(LEN(A8) = 2, IF(A8 = "10", "", B9 + 128), B9 + 128))))))
```

This operates on the basis of considering three possible conditions for the data, and depends on the length of data remaining to be processed. If the first two bits are '10' then (if there are three or greater bits of data for processing) the first three bits are taken into account, as the '0' will not require a change in the signal. As can be seen from Figure 14, if the third bit is a '1', then the signal is '101' (XYX), and the input should change 256 clocks after the start of the previous pause. If the third bit is a '0', then the sequence is '100' (XYZ). In this case the initial change for the '1' took place half way through the bit interval, so the next change occurs 192 clocks later.

Sequence	Pattern
100	
101	
00	
11	
01	

Figure 14: PICC output sequences with delays highlighted for stimulus file generation

If the first two bits are not ‘10’, then only the first two bits are taken into account, as the successive change in polarity occurs in the second bit. The possible conditions are ‘11’, ‘00’ or ‘01’. In the case of ‘00’ (ZZ) and ‘11’ (XX), as the same bit is in use, the pause will start 128 clocks after the previous one. For the case ‘01’ (ZY), the pause starts 192 clocks after the previous one.

The final condition is again when the first two bits are ‘10’ (XY), but when the length of the remaining information is two bits. This means that the ‘0’ in this case is the end bit, appended to the signal at the start of computation of the excel sheet. Since the initial ‘1’ would be handled by the spread sheet in the previous line of the sheet, no further action is required, as the ‘0’ will not require a pause to be inserted, as described in Table 2 and Table 3.

Once each calculation has been completed, all of the bits taken into account except the rightmost are removed, so that the remainder of the data may be handled. It’s necessary to preserve one of the bits already handled, as it is the specific sequence of bits, rather than the bits themselves that determine the starting point for the next “pause”.

Initially, this program was used to code signals such as REQA, which is seven bits in length, and ATQA, which consists of 16 bits. While it functions well for short sequences, it became quite time consuming to convert the longer sequences of bits such as the select command, which consists of seven bytes of data, plus parity bits, requiring entering 63 bits manually, and again resulting in the potential for human error to become a factor. As such, the spread sheet was revised to allow entering data as hexadecimal notation, and to automatically calculate parity bits.

One of the challenges of converting from hexadecimal notation to binary was the requirement of the protocol that bits be transmitted LSB first, and with a parity bit appended to the end. As such, a macro had to be written to reverse the order of bits in the binary representation of the hexadecimal input, and the parity bit was calculated using the following code, where the reversed binary number is in cell G4.

```
= CONCATENATE (G4, IF (MOD (LEN (SUBSTITUTE (G4, 0, "")), 2), 0, 1))
```

This simply counts the number of '1's in the result using `LEN (SUBSTITUTE (G4, 0, ""))`, which removes all '0's, and returns the remaining length. If this number is even a '1' is appended on the binary code, and if the number is odd a '0' is added. Using this spread sheet, the beginning of which is shown in Figure 15, creating the stimuli files – which for some commands were over one hundred lines in length – was much simplified.

	D	E	F	G	H	I	J	K
1	Start Time	Delay Length		Byte 1	Byte 2	Byte 3	Byte 4	Byte
2	110208	27	Hex:	02	ca	46	44	
3			Binary	00000010	11001010	01000110	01000100	
4			Binary rev	01000000	01010011	01100010	00100010	
5			w/CRC	010000000	010100111	011000100	001000101	
6	OUTPUT:							
7	000110208:00							
8	000110235:10							
9	000110336:00							
10	000110363:10							
11	000110528:00							
12	000110555:10							
13	000110720:00							
14	000110747:10							
15	000110848:00							
16	000110875:10							
17	000110976:00							
18	000111003:10							
19	000111104:00							

Figure 15: Example of input and output calculations for stimulus files

A simple program, whose output is shown in Figure 16, was also designed which made it possible to calculate the correct CRC bytes for any transmission. In this way it was possible to verify that the CRC bytes received and transmitted conformed to the format specified in the ISO 14443 protocol, and also that the microcontroller would be capable of decoding these bytes correctly on receiving them. This program allowed the calculation of CRC bytes for any length of input.

```

C:\MinGW\Files>CRC
Number of input bytes: 2

Enter byte #1: 02
Enter byte #2: CA

Step 1: ch = 6361, CRCreg = 6363
Step 2: ch = 65571, CRCreg = 6363
Step 3: ch = 71, CRCreg = 6363
Step 4: ch = 71, CRCreg = 72ec

Step 1: ch = 7226, CRCreg = 72ec
Step 2: ch = 75046, CRCreg = 72ec
Step 3: ch = 46, CRCreg = 72ec
Step 4: ch = 46, CRCreg = 4446

CRCreg = 0x4446
1: 0x2
2: 0xca
3: 0x46
4: 0x44
    
```

Figure 16: CRC generation and testing software

5.5.2 Logging

As well as the potential to simulate the commands from a reader, AVR Studio 4 offers the capability of logging the outputs of the simulated PICC as they change. The log file takes the same form as the stimulus, with first a set of numbers indicating the

transmission of thermal data from a system simply from the PC to the hardware; however it did require the use of PC software to perform the transfer of thermal data.

Speedfan is free software which monitors the temperatures reported by the various temperature sensors located on the motherboard of a PC, including graphics processing unit and (GPU) and CPU temperatures. The program also offers the capability of logging recorded temperatures in a comma-separated values file for analysis. The primary advantage of using Speedfan as monitoring software was the fact that it utilises a shared memory space for storage of temperature values while the program is running. This allows other programs to utilise this data, giving access to temperature readings from a large number of hardware configurations without the need to write individual programs for access to thermal sensors in different processors.

Operating in conjunction with the shared memory created by Speedfan, a program was written in C which periodically transmits the temperatures of the cores. This software used the free libusb USB library to connect and transmit data to the connected hardware through the computer's USB bus. The software simply searches the connected USB devices for the one with the expected device ID, and transmits the current temperature of the processor core as part of a USB control message approximately every three seconds. The user interface is a simple display of the last temperature which was transmitted over the USB interface.

An Atmel ATMega16 is connected to the USB port of the PC, and this receives thermal data from the PC software in the form of a USB control message. A counter on the microcontroller monitors the duration between the receipt of control messages, and this count is used to determine the duration of temperature cycles as they are stored. The

microcontroller then performs data analysis and reduction based on ordered overall range and rain flow cycle counting methods.

The process of analysis involves monitoring the general trend of temperature and recording the temperature values about which the general upward or downward trend reverses direction and exceeds the pre-determined hysteresis. In order to achieve this, a running record of the local maximum temperature in an upward trend or the local minimum temperature in a downward trend is kept, along with the duration of time since the previous reversal. Taking the case of upwards temperature trends only, if a subsequent temperature is higher than the local maximum then the new value replaces the old. The duration is also updated by adding the time measured since the previous local maximum.

On encountering a temperature value which is less than the previous local maximum by double the hysteresis value selected or more, the previous local maximum is considered a new reversal, and the newest temperature is recorded as the new local minimum, and the process begins again on the downward trend. The half-cycle is then added to the end of an array of slopes. Each slope structure stores the maximum, minimum and duration of the slope.

Each time a new slope is added to the array, the four rules governing cycle harvesting are applied to the new slope in order to record any half-cycles which may be recorded and removed from the array. The rules, as outlined in Section 4.6.2, are as follows.

- i. If a cycle starts inside a peak, and passes another peak of equal or greater amplitude, it terminates, and its duration and amplitude are recorded as a half cycle.

- ii. If a cycle initiates inside a trough, and passes another trough of equal or lesser amplitude, the same happens.
- iii. If a flow encounters water from a “higher” reversal, it terminates, and the flow from the higher point continues.
- iv. Half cycles based on peaks can be paired with half cycles based on troughs with similar amplitude to be recorded as complete stress cycles

For example, when adding an upward trend in temperature to the array, the array is traversed in reverse from the last value seeking a trend with an equal or higher maximum temperature. Until this temperature is encountered, any slope which has a lower maximum temperature may be recorded as a full cycle, under rules (i) and (ii). The newest slope is then updated to reflect the new duration and minimum value under rule (iii), and the next slope is considered. If the beginning of the array of slopes is encountered in this way without encountering a slope with greater maximum value, then rules (i) and (ii) can be applied, allowing the recording of the first slope as a half cycle.

The process of reducing the measured mean amplitude and duration of a temperature slope into the index of the slope in the array in memory is simple. For the index of the mean value, the following formula is applied, where t_{max} is the maximum value of the slope, t_{min} is the minimum value, r_m is the resolution of mean temperature values (2°C in this case), and T_{min} is the minimum allowed mean (27°C in this case):

$$m = \frac{t_{max} + t_{min}}{2} - (T_{min} - \frac{r_m}{2}) \quad r_m$$

By typecasting the resultant fraction to an integer, any decimal places are truncated, so the mean value is automatically rounded into the correct bin. Mean values which lie outside the range of expected values (27°C to 99°C in this case) are automatically placed into either the first index (for means lower than 27°C) or the last

(for means higher than 99°C). The result is that all mean values are stored in 37 blocks, numbered between 0 and 36.

Similarly for amplitude measurements, the following equation is applied to calculate the index of the amplitude block, where t_{max} is the maximum value of the slope, t_{min} is the minimum value and r_a is the resolution of temperature values (4°C in this case):

$$a = \frac{t_{max} - t_{min} + r_a}{r_a} - 1$$

Again, typecasting the result truncates decimal places, correctly rounding the resultant value into a block. A total of 19 blocks were used, allowing amplitude values between 4°C and 76°C. The blocks were numbered between 0 and 18.

As durations were not stored linearly in blocks depending on the length of the slope, it was difficult to develop a mathematical equation to calculate the correct bin depending upon duration. As a result, the duration blocks were assigned on a case by case basis using a 19 case if/else statement.

Previously, as the number of mean blocks in each amplitude block was equal, it was a simple matter to calculate the exact bin for any individual value of amplitude, duration and mean. The following formula would find the memory location of the least significant byte for a cycle of mean index m , duration index d and amplitude index a :

$$Address = 3(a D_t \times M_t + m D_t + d)$$

In this case, D_t and M_t refer to the total number of duration blocks (170) and the total number of mean blocks (11) respectively. The result of the equation is multiplied

by three to allow for the fact that each duration bin requires three memory addresses with size of one byte each.

However, in the revised version of the memory map, the number of mean blocks is not identical across different amplitude blocks, so the equation to find the location of a specific bin in memory becomes slightly more complex. The equation in this case, using the same conventions, was as follows - where D_t and M_t represent the total number of duration blocks (30) and mean blocks (37) respectively:

$$Address = 3(D_t M_t + 1 a - a^2 + D_t m - a + d)$$

The process of incrementing the relevant bin in memory requires reading up to three bytes of memory space to increment the relevant value. The bytes are stored with the lowest memory address containing the least significant byte. If the lowest byte is not FF hexadecimal, the value stored is simply incremented and restored. However if it is FF, the value stored in the memory space is reset to 0, and the same process is carried out on the subsequent address.

The end of time history occurs when a computer generates a suspend or shut-down request, as temperature cycles would be discontinuous between this point and the time when the computer starts up again. When a notification of suspend or shut-down is received by the hardware, the response is to propagate through the remaining half-cycles in the array, recording each cycle to memory, as outlined in rule (iv). In this way, assuming a PC is shut down in the normal manner, no information is lost. If the computer loses power or does not go through a normal shut-down process, any cycles which have not yet been logged to memory are lost.

5.7 Data Reduction - PC Software

Alongside the software transmitting thermal data over USB, a program was developed which executed the ordered overall range and rain flow cycle counting methodologies using the logs generated by the Speedfan program as input data. This allowed the analysis of a large volume of thermal data from logs generated by operational computers. This program generated output files in CSV format which could be used to create graphs of the type of temperature cycles. The operation of the software is the same as the Atmel implementation described in Section 5.6, however it utilises the time recorded by the Speedfan program to determine the duration of temperature cycles in place of the on-board microcontroller timer.

5.8 Conclusion

While development of a software solution to implement the MIFARE protocol represented probably the most significant time investment associated with this work, accurate adherence to the protocol is central to the goal of demonstrating that communication of LCM data can be achieved using modifications to existing hardware and supplemental features which it is possible to add to existing protocols at relatively low investment.

Through proper implementation of the MIFARE power and interface guidelines, a widely available microcontroller chip can be enabled to communicate as a MIFARE tag, communicating any data to which it has access. When queried as a standard RFID tag, the microcontroller responds in the manner expected from a MIFARE PICC. Additional features have been implemented in the hardware, however, which allow the

tag to relay a continuous stream of data until a “stop” command is received from the reader, or until the end of available data is reached.

Development of the software continued throughout the hardware development process, as various bugs and inconsistencies between simulation and reality arose in the hardware debugging process. This process is further examined in the following chapter.

Chapter 6

METHODOLOGY: HARDWARE DEVELOPMENT

6.1 Introduction

In this chapter, the hardware aspects of the project are presented and the processes for design of the various circuits are explained. The work necessitated the design of a number of hardware modules at different stages in the development of the prototype RFID PICC. The layout and purpose of each of these circuits is examined in detail.

Performance of the hardware and software in communicating with an ISO 14443 reader is evaluated, and the features extending this functionality are tested. Data rate and range of the prototype circuit are measured, and the data rate achieved is considered in the context of the optimal data rate for communications of this type.

The RFID reader used for the purpose of prototype development is the Elektor Electronics PCB reader shown in Figure 18. This is a relatively light-weight reader, which offers the advantage of a USB connection for extraction and evaluation of returned data. It also has the benefit of freely-available program code, and the capability of programming the reader circuit via USB, allowing the modification of the operating parameters which was essential in order to supplement the basic RFID functionality with the features specified in the project provisions.

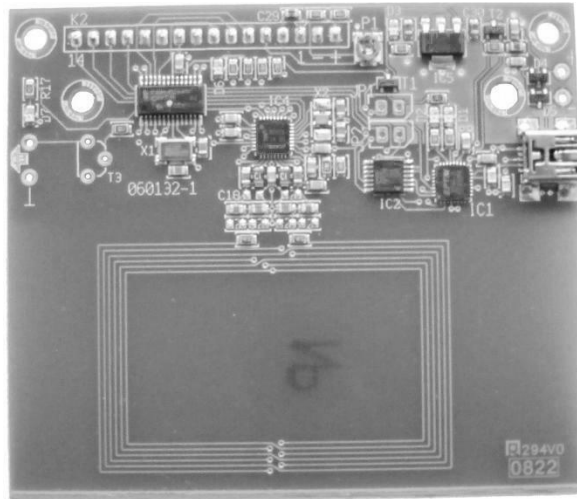
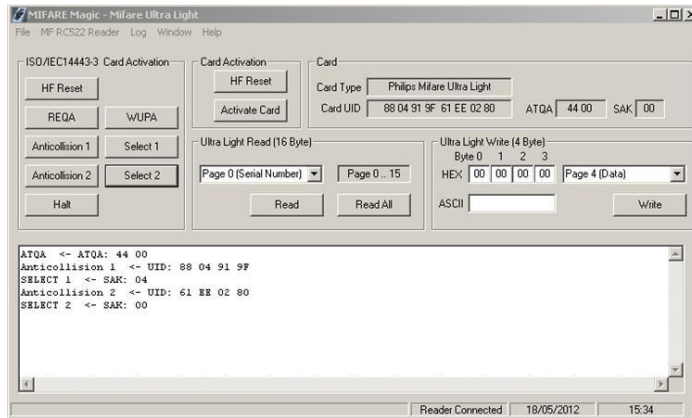


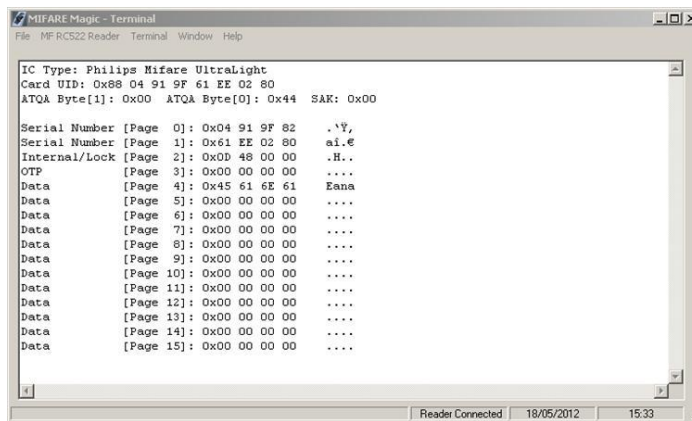
Figure 18: Elektor Electronics MIFARE RFID reader

6.2 Programming Interface

The Elektor reader's software interface is the MIFARE Magic program, which features a simple user interface (UI) for the purpose of testing the operation of MIFARE RFID cards, and allows the transmission of individual commands from the activation protocol, as shown in Figure 19(a). This was a useful UI for testing the prototype response to various stimuli from the standard protocol, as the prototype is expected to undergo activation in the same way as other MIFARE tags, and only when identified as a LCM tag is it necessary to change the way in which the protocol interacts with the tag. The MIFARE magic software also features a basic terminal mode which constantly polls for tags and prints out the contents of their memory. This is shown in Figure 19(b).



(a)



(b)

Figure 19(a): MIFARE UltraLight Interface, (b) MIFARE Terminal Interface

Following the development and simulation of the code as explained in the previous chapter, it was necessary to program the Atmel ATmega16 with the compiled software. The tool used is the Atmel AVR ISP MkII, which integrates directly with the AVR Studio software through a USB interface for simple programming of the IC. The AVR ISP Mk II has a six pin header interface, shown Figure 20(a). These pins are simply connected to the respective pins of the ATmega16 using the programming circuit shown in Figure 20(b). The programming interface automatically detects the correct microcontroller presence, and downloads the hex file created by AVR Studio.

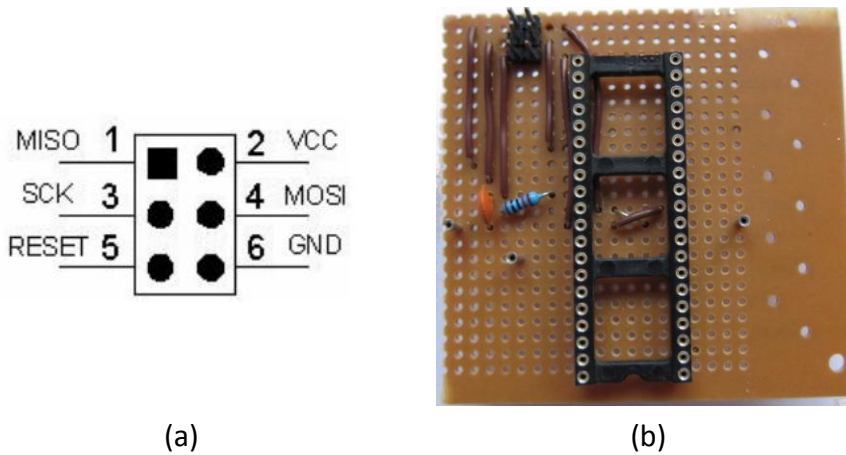


Figure 20(a): Six-pin header of AVR ISP; (b) ISP Programming circuit

Although it is possible to program the microcontroller while it is in its normal operating circuit, the locations of the pins which were being used as outputs and inputs of the circuit in operation made the relevant portion of the PCB particularly crowded, and it was decided that a separate programming circuit would be preferable to the addition of further electronic components to the PICC circuit.

6.3 RFID hardware

6.3.1 Receiver

As explained in the previous chapter, the software can decode the signals from the reader into the commands contained in the ISO 14443 protocol; however it is necessary for the signal to be demodulated before it may be decoded. The purpose of the electronic hardware aspects of the RFID circuitry is to perform this physical coupling and demodulation of the signals from the reader to the prototype. The signals from the reader are transmitted by means of a magnetic field oscillating at the carrier frequency of 13.56 MHz and modulated with the encoded information as shown in section 5.2.1, by means of 100% AM.

In receiving the incoming signal, a circuit with a resonant frequency close to the carrier frequency of the signal is used to maximise the power which is transferred between the reader and the PICC. This is achieved using a capacitor and inductor in parallel with a resonant frequency close to that of the carrier.

The resonant frequency of this configuration is given by the formula:

$$f = \frac{1}{2\pi \sqrt{LC}}$$

For a resonant frequency of 13.56 MHz, this gives the following combination of inductance and capacitance:

$$LC = 13.776 \times 10^{-15}$$

This requires small values of both inductance and capacitance to achieve. Using a 50pF capacitor, resonance is achieved using a coil with inductance of 2.755µH.

The formula for calculating the inductance of a coil of wire is as follows:

$$L = \frac{\mu_0 DN^{1.9}}{2} \ln\left(\frac{D}{d}\right)$$

Where L is the inductance of the coil, D is the diameter of the coil, d is the cross-sectional diameter of the wire and μ_0 is the magnetic field constant; $4\pi \times 10^{-7}$. Based on this formula, a coil of five turns, with coil diameter 5cm and wire cross-sectional area of 0.5mm was chosen, as it was expected to have a characteristic inductance of 3.079µH. Measuring with a RLC bridge shows inductance of between 2.9µH and 3µH. Using the value of 3µH in parallel with a 50pF capacitor gives a resonant frequency of:

$$\frac{1}{2\pi \cdot 50 \times 10^{-12} \times 3 \times 10^{-6}} = 12.99\text{MHz}$$

This is a deviation of only 4.2% from the desired frequency of 13.56MHz, which is sufficient accuracy for the purposes of prototype development.

In order to interface this analogue signal with the digital microcontroller, it is necessary to convert the signal to a digital equivalent. This is achieved by converting the unmodulated sections to a logical high level, and by converting the modulated sections to a logical low. The code outlined in the previous chapter requires that the signal at the digital input of the microcontroller is low for between one and three samples in order to be correctly decoded as a low signal. Because the sample period is 16 clock periods, in order to guarantee accurate interpretation of incoming signals, pauses in the signals must last at least 17 clock cycles, and no longer than 48 cycles. As each clock cycle is $1/f = 73.7\text{ns}$, this requires a pause to last between $1.25\mu\text{s}$ and $3.54\mu\text{s}$ to guarantee correct decoding. Therefore it is important that the rise time and fall time of the signal at the input of the controller is as short as possible, and that the signal is demodulated in a way which causes a minimum of skew to the incoming signal.

To accomplish this, it is first necessary to rectify the incoming signal, so that it is a continuous positive voltage. This is realized by placing a diode in series with the incoming signal. As the diode does not allow current to pass while reverse biased, the signal at the output of the diode is the positive half cycle of the incoming signal. A smoothing capacitor between the cathode of the diode and ground to remove excess rippling on the signal, and the voltage at this point was measured to ensure that it was sufficient to enable operation of the remainder of the circuit. The resultant signal is shown in Figure 21. This signal was measured with the antenna coil about 3cm distant

from the reader antenna, and the slow roll-off caused by the smoothing capacitor is clearly visible.

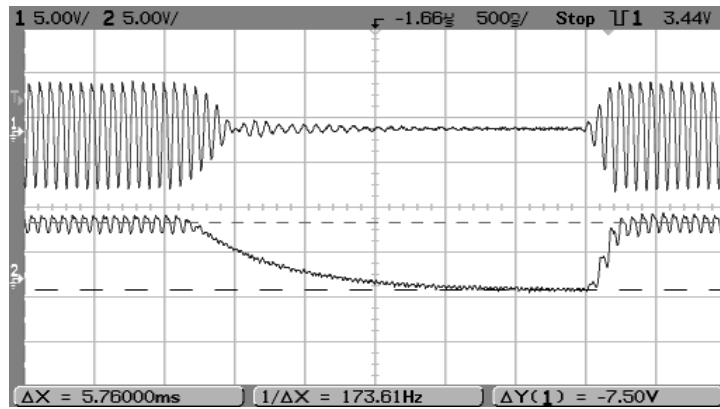


Figure 21: Incoming “pause” compared with signal after capacitor and diode

To achieve the fast switching necessary for consistent operation of the digital inputs of the controller, an operational amplifier is used, configured as a comparator. By varying the fixed voltage level at the inverting pin of the op-amp, the duration of the pause at the output of the comparator may be tuned.

The best balance of voltage smoothness and signal amplitude at the diode cathode is found to result from use of a 10 k Ω resistor in parallel with the capacitor in a low-pass filter configuration. As the voltage is higher than the 5V supply in use, a 1.8k Ω and an 8.2k Ω resistor are used in a voltage divider configuration to achieve the smoothness offered by the 10k Ω load, while reducing the voltage to approximately four fifths of its level at the output of the diode. A second voltage divider is used between the 5V power supply and the inverting input of the comparator to select the appropriate voltage level for switching to occur. Experimentally, it was determined that a switching voltage of 3V offers the required pause duration over an acceptable range of distances between the reader and prototype circuit.

The signal shown in Figure 22 is the output of the comparator, which is connected to the digital input of the ATmega16. The pictured waveforms were captured with the reader at a distance of approximately 3cm from the antenna. As shown, the pause in the signal has duration of 2.8 μ s, which is perfectly within the acceptable range of values of 1.25 μ s to 3.4 μ s. as laid out earlier, so a pause of this duration will consistently be decoded correctly by the software. The complete demodulation circuit is shown in Figure 23.

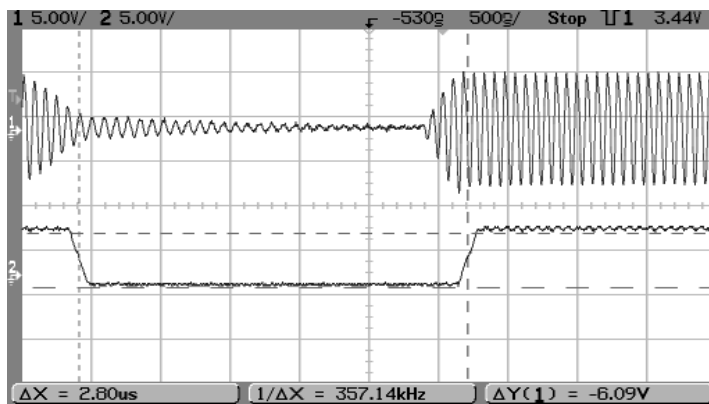


Figure 22: Pause at output of comparator

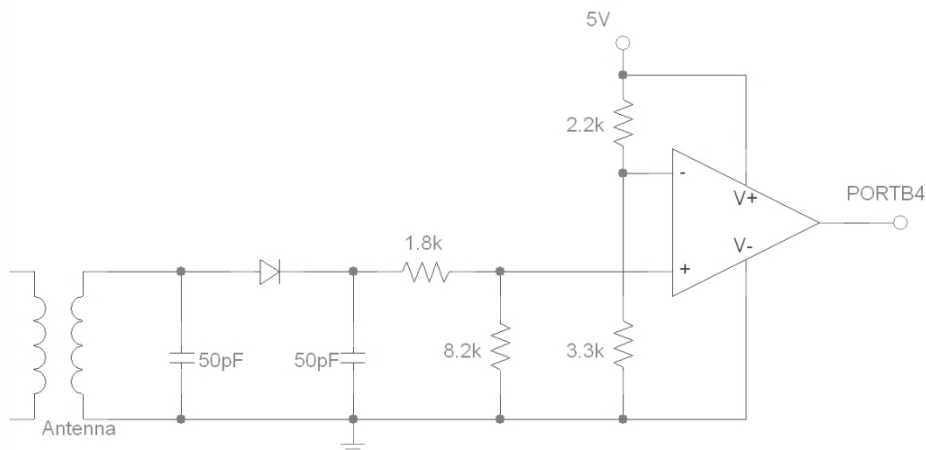


Figure 23 :Receiving circuit including component values

At this point, it was possible to evaluate the overall performance of the hardware in the context of receiving coherent signals from the MIFARE standard. A REQA signal can be transmitted directly from the MIFARE UltraLight UI of the MIFARE magic program. Capturing this signal allowed the comparison of the overall demodulated

signal with the anticipated result. Figure 24(a) shows the REQA signal as described by Elektor Electronics, and Figure 24(b) demonstrates the equivalent signal retrieved by measuring the voltage at the input pin of the ATmega16. The expected signal has strong correlation with the captured one, indicating that the decoding circuit operates as expected.

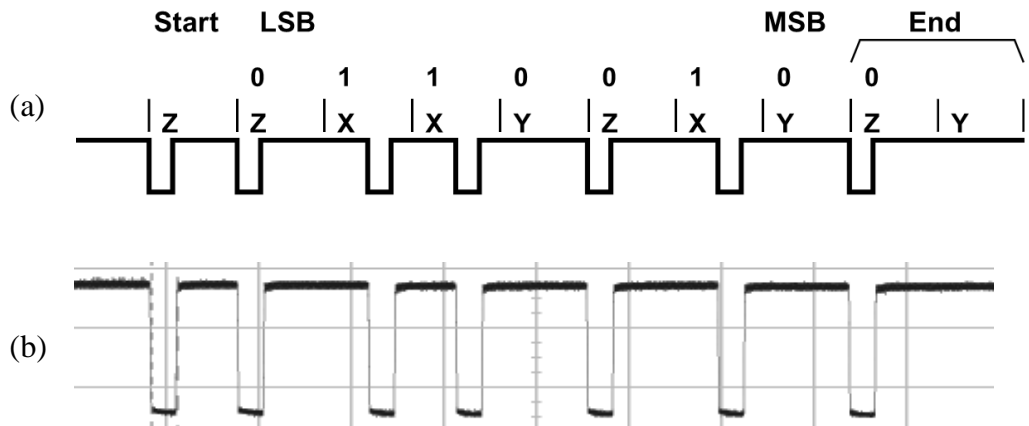


Figure 24(a): Elektor description of REQA signal; (b): Signal measured at microcontroller input

6.3.2 Transmitter

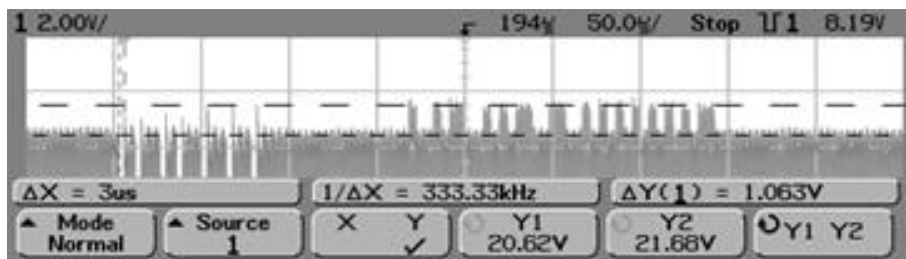
Having designed the receiving features of the RFID circuit, it was necessary to integrate the transmission feature. Again, all of the encoding features are performed by the software embedded on the ATmega16, so the hardware in this section is only required to transmit the signal from the microcontroller via the RFID interface.

PICC responses are encoded directly onto the signal which the reader emits, so it is unnecessary to generate a carrier wave within the prototype circuit. Data is embedded on the signal by modulation of the load applied to the antenna. This is achieved using a transistor - the 2N3820 p-channel general purpose amplifier - in a switch configuration, to short-circuit the pins of the antenna.

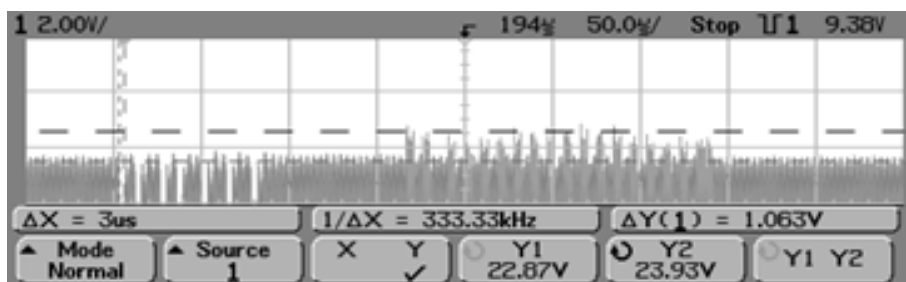
When there is no voltage difference across the gate-source junction of the transistor, the drain-source resistance is negligible. When a positive voltage difference

is created, by applying 5V from the output pin of the ATmega16 to the gate, and 0V to the source, the drain-source junction becomes effectively an open circuit. In this way, by connecting the drain-source junction in parallel across the antenna, the antenna load can be changed by alternating the voltage at the output pin of the ATmega16.

In order to evaluate the effectiveness of the transmission as generated by the prototype circuit, it is necessary to compare the signal which is transmitted by a MIFARE RFID card. This is achieved using a second coil to intercept signals which are transmitted between a reader and a tag. In this way the signal amplitude could be determined and compared with the signal generated by the prototype tag developed for this work to ensure that the signals are of similar amplitude and definition. Figure 25(a) shows the signal measured in this way during communication between the reader and a MIFARE UltraLight ID card, and Figure 25(b) shows the same communication between the reader and the prototype PICC circuit measured at a similar distance. Although the amplitude of the prototype signal is slightly less regular than that from the tag, the loads are in a very similar range.



(a)



(b)

Figure 25(a): Measured induced voltage from MIFARE Card & (b) from prototype

6.3.3 Signalling & reset

In the hardware as described this far, there is no process for examining the operation of the software coding, or of knowing at which point in the software any problems occurred. Also, on occurrence of an error from which the system does not recover through normal operation of the code, there is no process by which to return to normal operation except by removal of the power supply. This process was time consuming in the initial testing processes of the hardware, during which time errors in code execution were relatively common.

As the application software is designed in the form of a state machine, it is possible to signal the approximate position in code by indicating the state code currently being executed by the software. To indicate the current state, three light-emitting diodes (LEDs) were added to the outputs of PORTB, and the program code was modified to light the LEDs in a pattern dependent on the code currently in execution. The signals are given in Table 7 below, with 1 representing an LED which is on, and 0 representing off.

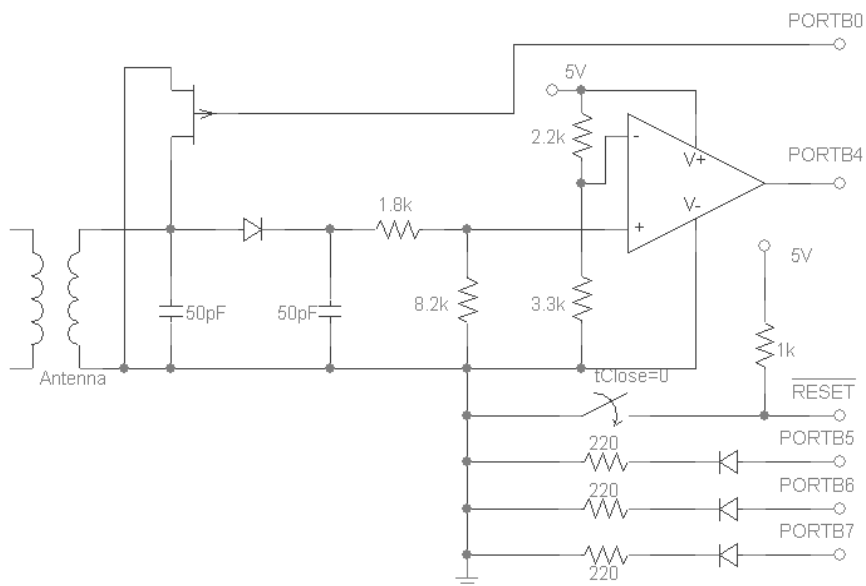
Pattern	State
000	Circuit off/no power applied
001	Programmed, but not yet in idle state (error)
010	In idle state, waiting for REQA command
011	In anti-collision state
100	In ready state, waiting to receive MIFARE commands

Table 7: LED signals for state machine

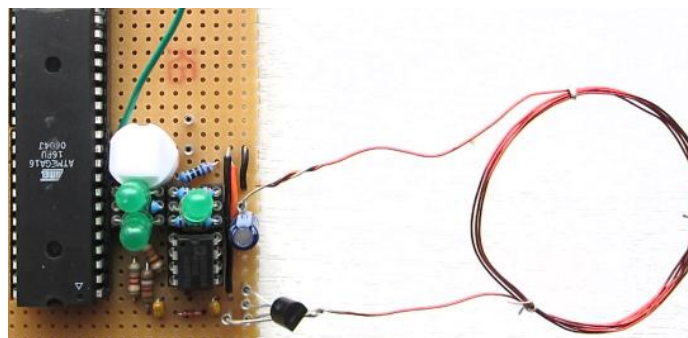
In order to recover from the occurrence of errors from which the software could not return to a valid state, a reset switch was added. This is a push-to-make switch connected between the active low RESET pin and the ground plane, with a 220 Ω pull-up resistor connected between the RESET pin and VCC. When the switch is not pressed, the resistor holds the pin at a high voltage level, and when the switch is activated the pin is pulled low, resetting the circuit.

The combination of the LEDs and the reset switch was useful throughout the debugging process, as the switch allows recovery from error states, while the LEDs aid in the debugging of causes of errors in the code, allowing their easier rectification.

The final combined circuit design, demonstrating the connection of the receiving and transmission sections, is shown in Figure 26(a), and the resultant prototype is in Figure 26(b).



(a)



(b)

Figure 26(a): Complete RFID hardware solution diagram; (b) Prototype circuit

6.4 Software modifications

For the purpose of testing the operation of the PICC circuit, additional features were added to both the MIFARE magic program and to the reader firmware. These

features enabled the reader to correctly identify a tag containing LCM data in the reader field, and allowed it to respond with the appropriate commands to extract the LCM information.

The modifications, which are detailed in appendix B and appendix C, add functionality to the MIFARE Magic UI and allow situation-specific interactions when the presence of a tag with access to LCM data is detected. During the initial handshaking process, the PICC identifies itself using unique SAK and ATQA codes. On using these codes for identification of the presence of a PICC which has access to LCM data, additional features become available in the MIFARE UltraLight interface, allowing activation of the supplemental features in the PICC as shown in Figure 27.

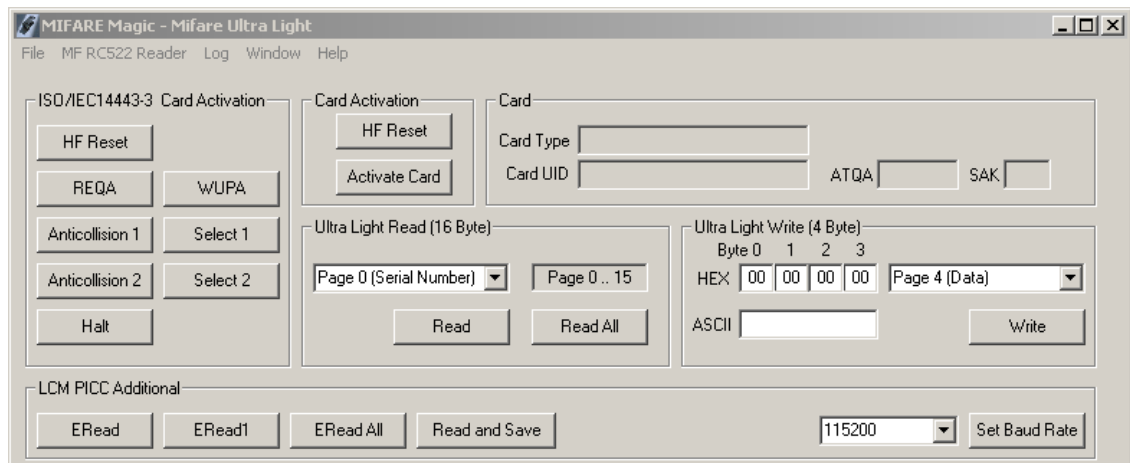


Figure 27: Mifare Magic interface, with additional options for LCM PICCs

The additional options specifically for PICCs with access to LCM data are shown in Table 8 below:

Set Baud Rate	Modifies the communication rate from the reader through the USB interface
ERead	Requests the transmission of the first packet of LCM data from memory by transmitting the code 12 _h
ERead1	Requests subsequent packet of data using the code 13 _h
ERead All	Continuously reads data from the PICC by transmitting 12 _h followed by 13 _h after each received packet, until the termination code, 16 _h is received
Read and Save	Opens interface, allowing user to select save destination and saves all data retrieved from PICC in a file in the selected location

Table 8: Additional features in MIFARE Magic software

The “Read and Save” option is the case which performs the analysis of the information held by the prototype. When a user clicks this, the file save dialog allows creation of a file, and the data retrieved from the PCD is stored sequentially in comma-separated values format in this file. The structure of the file is designed to function intuitively with the structure of rain flow cycle counts as laid out in section 4.6.2, with 170 columns representing all of the cycle duration, and each row representing a different mean value. As the structure of this file would be fixed, analysis of the resultant data would be simple, given information about the relative effects of each type of cycle. Each cell in the analysis could be assigned a weight depending on the impact of its associated cycle, and by multiplying the cells by their respective weights and summing the resultant values, a simple quantification of the devaluation of the PC could be developed.

6.5 Optimal timing

Without taking into account any overheads, the maximum theoretical data rate for communications between the PICC and reader is given by the equation $fc/128 = 105.9\text{kb/s}$. This only takes into account the fact that each bit which is transmitted takes 128 clock periods to transmit. From the section regarding rain flow cycle counting, the

maximum amount of data which would have to be transmitted is 142 kB, which at a data rate of 105.9kb/ s (13.242kB/s) requires 10.723 seconds.

However, there are considerable additional time costs to be considered in a realistic context. Firstly, each byte which is transmitted requires a single parity bit, meaning the data rate for transmission in each direction is only 11.77kB/s. Additionally, for any communications in active mode, there is overhead associated with the CRC bytes which must be appended to the transmitted data, along with the first byte which indicates that there is further data to follow. There are two CRC bytes for every data frame transmitted, which is 16 bytes in length including a single chaining byte. This further reduces the data rate, as only 15 out of every 18 bytes transmitted are actually information. Furthermore, each communication is preceded by a start bit, meaning a total of 28 extra bits are transmitted for each data block, as well as the parity bits. The resultant data rate while the PICC is transmitting data is approximately 9.748 kB/s. Finally, each block from the PICC requires a response from the PCD prior to transmission of the subsequent block. As each block from the PCD is a single command byte followed by two CRC bytes, this requires the duration of three further bytes per block, in addition to start, end and parity bits. Additionally, the minimum time between the end of a block from the PICC and the beginning of a response from the PCD is defined in the ISO 14443 standard as approximately 80.2 μ s, and the equivalent pause for transmissions in the opposite direction is at least 86.4 μ s.

In total, the transmission of 15 bytes of information requires 1.98ms, using the minimum frame delay times permitted by the protocol, with a resulting time per byte of 132 μ s, or an effective data rate of 7.579kB/s, meaning over 42% of the time associated with the transmission of information is devoted to error checking and other overheads.

A graphical representation of the relative time spent transmitting overheads is shown in Figure 28.

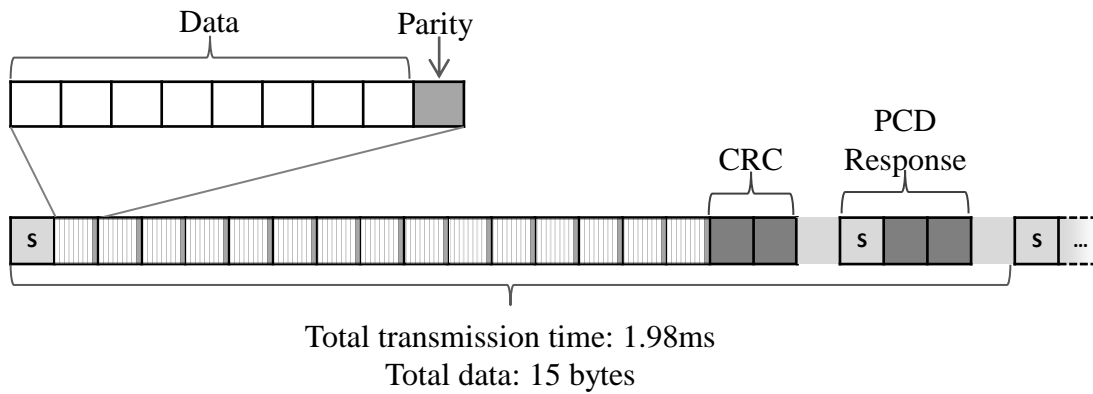


Figure 28: Visualisation of optimal overhead in transmission of data

It is potentially possible to decrease the overall significance of the error checking overheads by increasing the volume of data which is transmitted in each block of information; however the additional processing time required by this approach may increase the time delay between the transmissions of frames, negating any time efficiencies obtained.

Taking the more accurate data rate of 7.579kB/s, the transmission of the total data package of 32kB would require 4.22 seconds. This time is longer than would be ideal for the analysis of used PCs; however it represents a significant improvement when compared with the time investment required to evaluate the viability of PCs at present.

6.6 USB Interface Hardware

The extraction of information through the USB interface was performed primarily through the USB libraries on the PC and the ATMega16. The only requirement from the point of view of hardware development was in order to reduce the 5V data output from

the ATmega16 pins to the 3.3V as required by the USB standard. This was achieved by inserting 3.6V zener diodes between the data pins and ground, as shown in Figure 29.

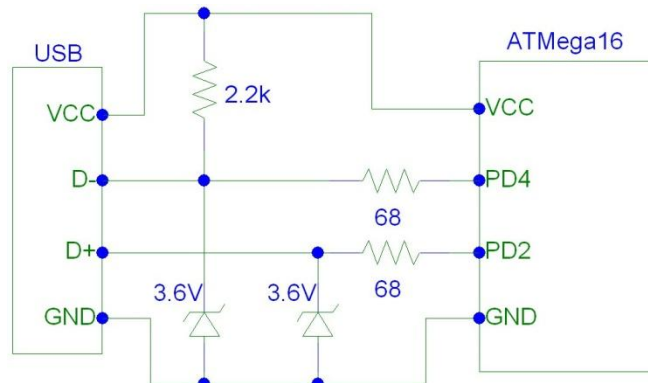


Figure 29: Interface between USB cable and ATmega16

6.7 Conclusion

In this chapter, the hardware components of the solution have been presented. The process of development for the hardware has been laid out for the interfaces required by the system, both between the PC in test and the hardware during the PC life cycle, and between the EoL PC and an RFID reader. In combination with the software as presented in the previous chapter, these circuits allow the monitoring of PC temperatures throughout system lifetime, characterisation of these temperatures by in-situ circuitry in operating PCs, and extraction of this thermal history through the RFID interface at end of life.

Chapter 7

RESULTS

7.1 Introduction

Following the implementation of the project through the prototype as outlined, it is possible to characterise the performance of the different aspects of its operation in order to determine the effectiveness of the solution. The results of these analyses are presented in this chapter.

7.2 Data Reduction

Performing data reduction and analysis on information retrieved from operational computers enabled the examination of the differences between different temperature profiles of PCs. By side-by-side examination of temperature profiles for a desktop PC and a laptop, it is possible to derive some indicative data showing the differences between computers with drastically different thermal profiles.

The data retrieved from ordered overall range reduction and rain flow counting analysis is essentially in a three dimensional matrix of count values, which does not lend itself to being easily graphed, however it is possible to compare pairs of factors on 3D graphs. By looking at the pairs of data it is possible to develop an overview of a system's thermal history.

The desktop PC used to generate these results was operational for 28 days, 24 hours a day. The laptop had operating conditions more characteristic of mobile computing, with regular suspend and shut-down events. The result is a significantly lower total number of cycles, reflecting the shorter operating time. The overall results

may still be used to compare the normal operating temperature trends of the systems, however, as the trends may be expected to scale linearly during continued operation.

When examining the duration of cycles and the amplitude of the cycles from each system, no significant difference exists between the general trends for the desktop and laptop tested beyond the scaling of the cycle numbers, which can be attributed to the overall greater period of time for which the desktop PC was operating. The graphed results for duration and amplitude are shown in Figure 30 and Figure 31.

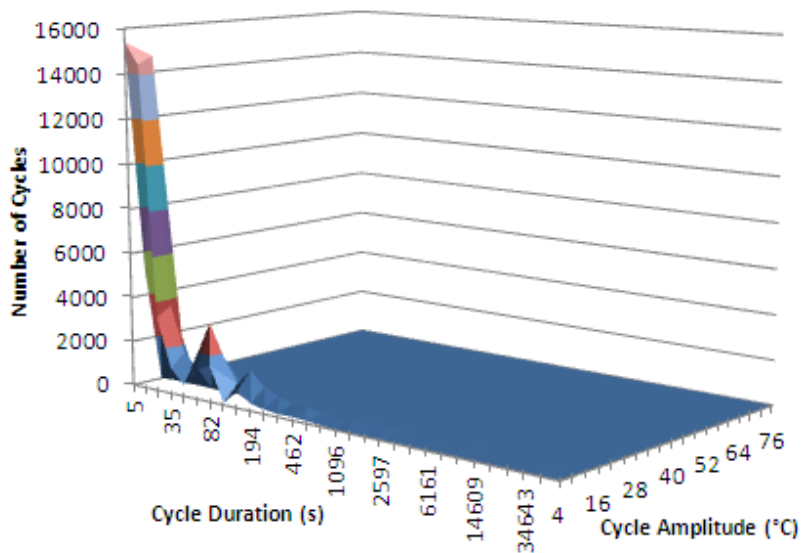


Figure 30: Graph of cycle duration against cycle amplitude counts in desktop computer

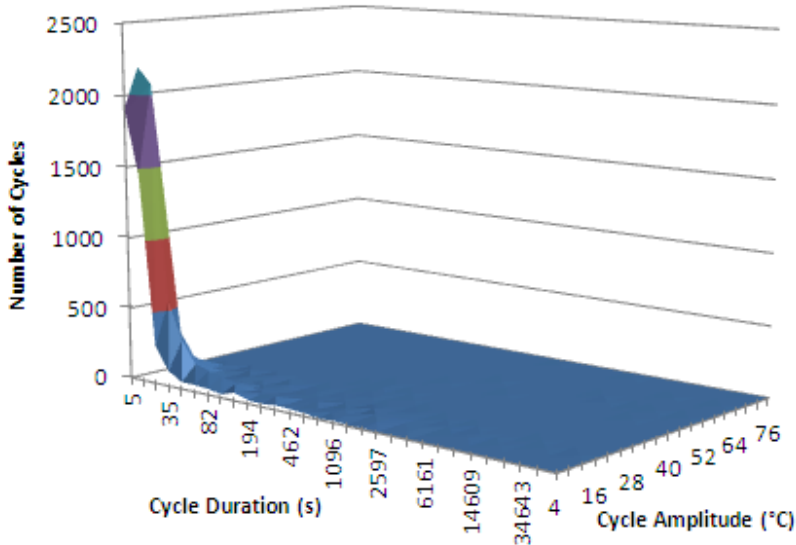


Figure 31: Graph of cycle duration against cycle amplitude counts in laptop computer

In the case of cycle mean against cycle duration, there is a much clearer difference between the profiles of the laptop and the desktop computers. In this analysis, shown in Figure 32 and Figure 33, the difference between the mean temperatures of the different computers is evident with the desktop mean remaining around the minimum sensor reading of 35°C, while the laptop temperature was far higher, around the thermal cut-off point of 100°C.

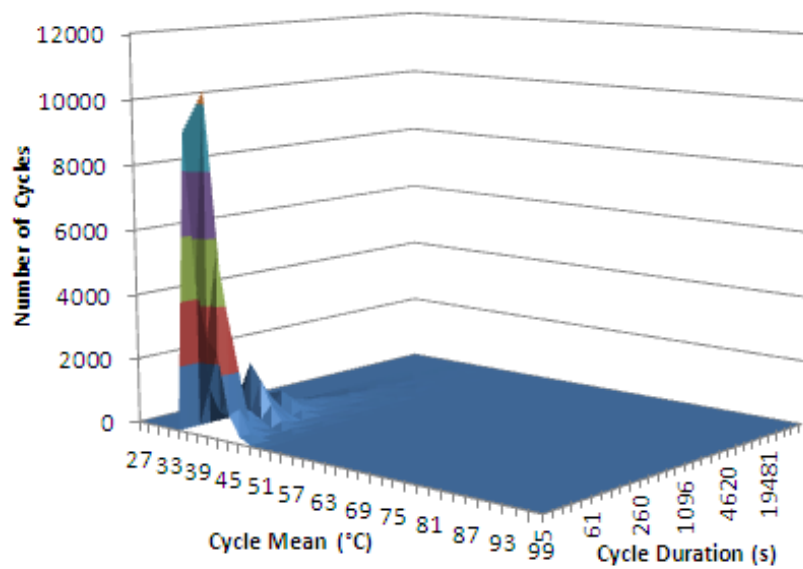


Figure 32: Graph of cycle duration against cycle mean counts in desktop computer

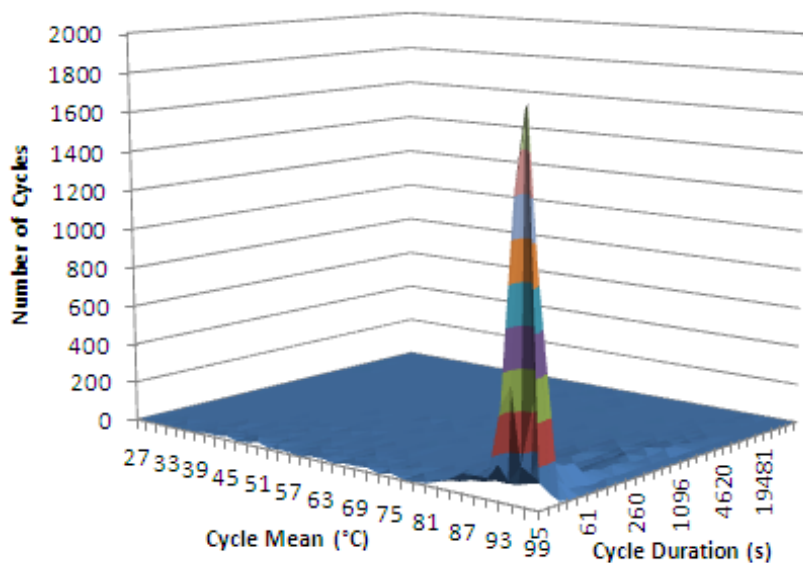


Figure 33: Graph of cycle duration against cycle mean counts in laptop computer

The final pairing of parameters, cycle amplitude against cycle mean, shows the results which could be expected from the previous results; that the amplitude values for the two system configurations are similar, while the mean values are at opposite ends of the temperature range. These results are presented in Figure 34 and Figure 35.

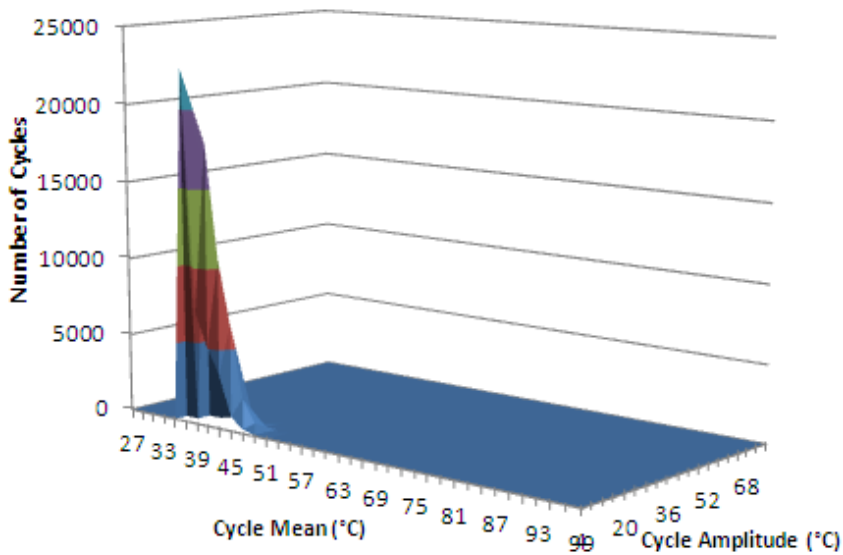


Figure 34: Graph of cycle amplitude against cycle mean counts in desktop computer

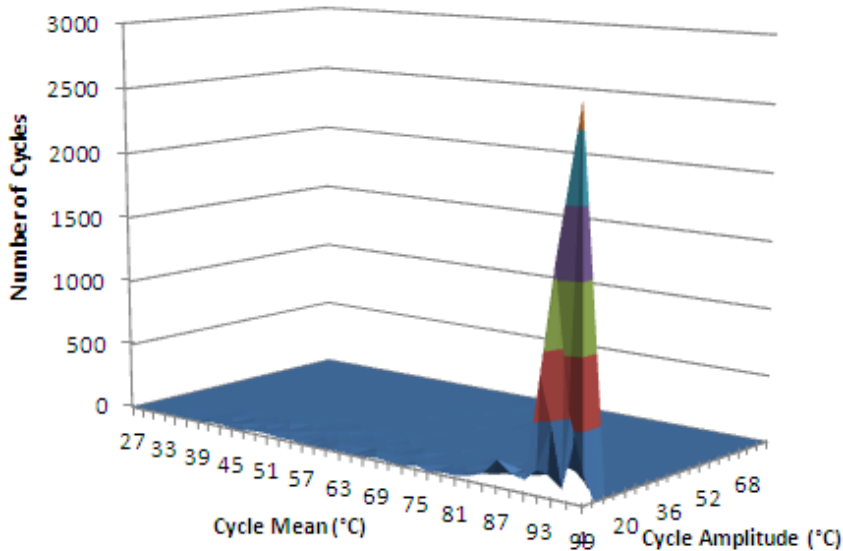


Figure 35: Graph of cycle amplitude against cycle mean counts in laptop computer

These graphs show that the vast majority of temperature cycles in both the laptop and the desktop case were cycles of short duration and of small amplitude.

Examination of the data shows that a huge proportion of the cycle amplitudes are in the lower range, with 99% of desktop cycles in either the 4°C or 8°C blocks. The laptop experienced a wider range of cycle amplitudes, with 78% in the 4°C or 8°C blocks, and 10% in the 12°C block.

76% of the cycles which take place on a desktop are in the first four duration bins, meaning they last less than 35 seconds. The laptop has a general trend of a greater proportion of cycles having shorter duration than the desktop, with 80% of duration cycles in the first three duration blocks, lasting less than 25 seconds.

The mean measurements retrieved are slightly limited by the characteristics of the digital thermal sensors in use. In the case of the desktop, the thermal sensors did not report temperatures lower than 35°C, and owing to its effective cooling solution and generally lower load profile the temperature of the processor normally occupied the lower end of the spectrum. This is reflected in the fact that 94.9% of temperature cycles have mean values between 36°C and 44°C.

The laptop features processor choking to limit catastrophic failure, which cuts the processor operation if the temperature approached 100°C, meaning that the temperature will tend towards 100°C when the computer is in operation. This is shown in the cycle mean temperature distribution, with 84% of cycles occurring between 90°C and 100°C. The laptop demonstrated a wider overall distribution of cycle values than the desktop.

7.2.1 Analysis

These graphs and statistics can be useful in providing an overall indication of the thermal characteristics of a system; however a valuable feature of three parameter cycle counting is the capability of characterising the system's cycle history using all

parameters simultaneously. In-depth study is required into the exact effects of different thermal cycle load profiles on the integrity of electronics and their resultant reliability; however the ability to identify the key combinations of cycle duration, amplitude and mean which can lead to the greatest impact on system functionality is the primary advantage offered by this method of data characterisation.

By weighting different cycle patterns depending on their effect on reliability, it should be possible to use the information retrieved in this way to more meaningfully assess the projected reliability of an individual system. While the process for derivation of the data is relatively complex, the idea of identifying particular patterns of cycles which have the greatest impact on reliability is simple, and the resultant metric should be easily understandable to people of limited technical knowledge.

7.3 Data Extraction

From analysis of signals retrieved from PICC-PCD transmissions, the prototype circuit was found to perform as anticipated, reliably transmitting data within a range of approximately 5cm of the reader. Analysis of the signals at the antenna, the input of the comparator and the input of the ATmega16 shows that the breakdown of the interface owing to range occurs when the high signal at the input of the comparator approaches the switching voltage causing bouncing in voltage levels at the microcontroller input. While lowering the switching voltage by altering the ratio of the resistors in the voltage divider network potentially increases the range of the receiver circuit, the trade-off is that the duration of a pause decreases owing to the slow roll-off of the voltage at the non-inverting input of the comparator. The range offered by the design was considered to be appropriate for a broad range of applications, and a relatively short range may

offer benefits in situations with tags in close proximity, as longer range devices could introduce uncertainty as to the tag being currently targeted by a reader.

A fast data transfer rate is an important feature of this type of system, as it would offer significant benefits in environments where a large quantity of used computers are being analysed. In that regard, the system was found to be disappointing. Initial data transfer time was measured at over 45 seconds, although significant improvements were realised through optimisation of the MIFARE Magic software code and the hardware code on the reader processor.

Further optimisation may be possible, as the preconfigured code is not intended for the transfer of significant quantities of data. At present, the transmission of a sample of 15 kB of data requires roughly 8 seconds; a data rate of approximately only 1.8kB/s, less than a quarter of the optimal data transfer rate of 7.579 kB/s. To transfer the 142kB specified in section 4.6.2 would require 78.9 seconds. The inefficiencies observed are attributed to the fact that the Elektor reader and software in use does not necessarily conform to the absolute minimum frame delays between communications of frames in the ISO 14443 specifications. The reader relays information via the USB interface between receiving packets and replying, and it is likely that the additional interaction between the reader and the computer is significantly increasing the extent of the time taken to transmit and receive data.

In order to identify the exact inefficiencies in the transmission and receipt of data which led to the deviation from the ideal data rate, the communications in each direction were examined in greater detail. The PCD to PICC communication block is one command byte followed by two CRC bytes, with one parity bit added to each byte, and a start bit and end bit for the block. As such, the total data to be transmitted is 29

bits, with each bit requiring $9.49\mu\text{s}$ for transmission. This results in a total expected transmission time of $273\mu\text{s}$. The actual signal, when measured, had duration of $271\mu\text{s}$. The variation from the calculated value can be attributed to the cursors of the oscilloscope not being precisely located at the end of the end bit.

Communications in the opposite direction also require $9.49\mu\text{s}$ per bit to be transferred, however the volume of data is significantly greater in this direction, with 16 data bytes and two CRC bytes being transmitted in this case. Including parity and a start bit, this comes to a total of 1.539ms . On measuring the resultant signal with an oscilloscope, the measured duration conforms to the expectations from calculations, at 1.54ms .

As both of these communications were within the expected range of acceptable values, it followed that the reduction in data rate could be attributed to the delays between the communications, so these delays were considered next. As the delay between PCD communications and PICC response is determined by the program code developed in this work, this was considered first. The minimum value permissible for the delay between the last modulation in PCD-to-PICC communications and the first modulation in PICC-to-PCD communications is $86.4\mu\text{s}$. On measuring the practical time of this delay, it was found to be approximately $220\mu\text{s}$, over double the minimum permissible value.

While this delay would increase the time required for transmission of data, the impact of the additional $134\mu\text{s}$ would be less than the transmission of an additional two bytes per data frame, or less than 7% of the total time which the data required to transmit. This does not account for the magnitude of the difference in communication time between the expected and observed results.

The only remaining time difference to be explored was the length of time for the PCD to respond to each data packet from the PICC. As this delay is not dependent on the PICC prototyping code, was to some extent neglected. During the pause between PICC communications and PCD response, the reader transmits the information received through the USB port to the middleware, in this case a PC, and the middleware performs error checking and stores it prior to transmitting a response. The communication rate between PCD and middleware may be modified by altering the baud rate of the virtual serial port in use, however this was not found to have a significant effect on the overall data transfer rate, leading to the conclusion that the analysis and storage processes required the bulk of the interval. Whereas the ISO standard defines the minimum permissible duration between the transmissions of bytes to be $80.2\mu\text{s}$, the measured interval was in fact 5.78ms , which was 72 times the minimum permitted duration, and almost three times the overall minimum duration for information exchange.

Taking into consideration the measured intervals, the total duration required for the transmission of a single packet from the PICC and a response from the PCD is 7.813ms . This accounts for the duration required for the transfer of 1000 packets of data, or 15kB , at 7.813 seconds. To demonstrate the proportion of the communication time occupied by frame delays the proportions are outlined in Figure 36.

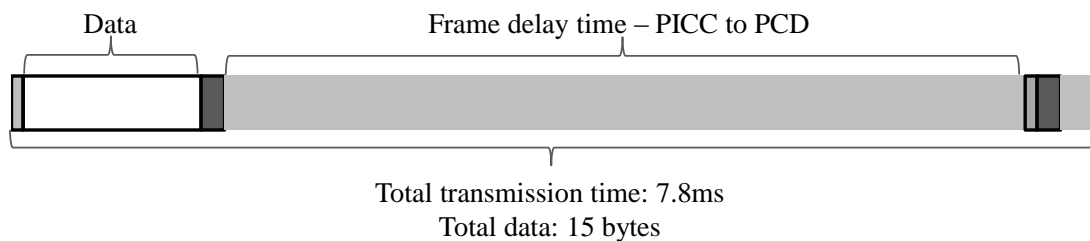


Figure 36: Visualisation of practical data transmission timing

7.4

7.4.1 Analysis

The design of a RFID tag with the capability of transmission of large volumes of data has been shown to be successful. The potential of extending the technology for use with significantly greater volumes of data than those traditionally considered has been demonstrated to be practical by relatively low cost implementation of a prototype circuit.

Although the time required for the transmission of large amounts of data is greater than was anticipated when initially determining the potential of the RFID interface, the solution described here nonetheless represents a significant economy over current processes for analysis of used PCs. Further scope for improvement in data rates exists through optimisation of software and hardware processes. Through examination of the interface between PCD and middleware further efficiencies in timing parameters may be realised.

Finally, it should be considered that the rain flow cycle counting data volume was chosen from a convenience perspective and adapted based on available data, and does not necessarily reflect the optimal arrangement of data bins. Further optimisation of this data reduction methodology could result in significant reductions in the volume of data which it would be necessary to transmit.

Chapter 8

DISTRIBUTION AND IMPLEMENTATION

8.1 Introduction

In previous sections the fundamental principles of a solution for the extraction of life cycle data are laid out. On the basis of the proof-of-concept as designed, it is evident that a simple system of this type can be implemented at low cost, using modified protocols operating on hardware which is in widespread use. The barriers to the acceptance of this type of technology are low, and all stakeholders in the manufacturing, use and refurbishing processes could benefit from its employment.

The system developed in this project is easily modifiable to be used in a large variety of PCs, across different brands and technological configurations. As such it could be modified in such a way as to be a stand-alone product which could be included as required in a wide range of PCs. However, the benefits of its utilisation can be increased significantly if hardware manufacturers become actively involved in the integration of the system by implementing it at the design stage of the development process.

The benefits and outcomes of adoption of the system vary depending on the specific method of its implementation. In this chapter the various ways in which a system for LCM data extraction could be applied at different stages of a computer's life cycle and the advantages of each type of structure to the actors in the market are discussed. Also examined are several platforms whereby the technology could be used by various stakeholders to extend the use phase of the PC, and the value added to the

cycle by each distinct type of implementation is compared. This is by no means an exhaustive examination of the possible applications of this system; however the more obvious means of its implementation in the context of current markets are presented.

8.2 Manufacturers

Although PC manufacturers may be expected to offer resistance to any initiative to increase the rate of reuse of their products, they are in fact one of the actors in the life cycle who are best situated to benefit from a formal secondary PC market. Second hand products may be regarded as unwanted competition for manufacturers' newer products, however benefits of reuse may be realised by manufacturers establishing control over the flow of used products, both through reselling their own products as refurbished, remanufactured or second hand goods and through presiding over the flow of waste products at end of life. As the acceptance of manufacturers is a cornerstone of the implementation of the entire system, it is important that they recognise the prospective benefits.

A prerequisite to the proper implementation of a system of this type - if manufacturers are to benefit - is a closed-loop supply chain. In order for manufacturers to properly implement a system for the reuse of PCs, take-back of used products should be implemented and it should be ensured that users of PCs take advantage of the take-back schemes offered. By ensuring that their products are returned to them after the use phase, manufacturers can govern the reverse logistics process.

To a certain extent, the WEEE directive may be an asset to manufacturers in ensuring that they recover a proportion of their products at end of life. Because the directive includes the requirement that suppliers of products provide a like-for-like take-back service, manufacturers are required to take back a used computer when they

provide a new one. As such they can recover used computers as they provide new ones to customers, although this assumes brand loyalty from customers.

Another method of closing the loop of the supply chain is through the provision of technology as a service (TAAS) or lease-based sales. In the TAAS model - which is recently gaining popularity - rather than providing the traditional once-off product of the PC and peripherals, suppliers provide the service of computing on an on-going basis. This may be inclusive of support, and sometimes of continuous upgrading of the product in line with technological advancement. As such, used PCs would automatically return to manufacturers for scheduled maintenance and refurbishment, and at this time can be analysed to evaluate the usage profile and the expected life remaining in the product. Manufacturers benefit from the process as their products are kept out of the waste stream, reducing their overall WEEE responsibility, and they can also potentially use parts from recovered systems to build refurbished systems for resale.

8.2.1 Advantages

Compared to other actors in the EoL logistics process, manufacturers may have several advantages if they decide to implement resale of used, refurbished or remanufactured products. As the original providers of the product, manufacturers have access to suppliers of parts and components which can be used for refurbishing and remanufacturing of used products. They can utilise economies of scale where new components are required for the refurbishing process, and return used systems to a working state at a lower cost than other actors in the reverse logistics industry. Manufacturers also have the technical knowledge about the compatibility of different components and the technological capability to design working systems from a wide variety of parts. This allows them to replace defective components with new or used ones, maintaining compatibility between parts, and returning used PCs to a close to new

state. Additionally, they are the actors most familiar with their own hardware configurations, and may be those most capable of diagnosing and repairing faults in returned systems.

PC manufacturers have recognised brand names, which consumers seek out first when purchasing new PCs. Although it may seem counter-intuitive, by providing consumers with the option to purchase a used system rather than a new one, along with all the applicable guarantees and warranties, manufacturers would have access to a low-cost source of viable computers and could potentially tap into a customer base of users who require only basic PC functionality and are not willing to pay for features present in newer computers, which they may not use.

The primary advantage of manufacturers' participation early in the adoption of a reuse system is the availability of capital, allowing the initial sale of reused computers to be operated at low profit. As it is unlikely that consumer attitudes would change overnight, there would be some time during which the existence of the option of reuse becomes embedded in the consciousness of consumers. During this time, it is likely that there would be little profit from the systems, or that it would operate at a loss. As such, it is essential that there is sufficient investment in the system to allow it to outlast this initial period and become more economically sustainable.

An additional benefit of the involvement of manufacturers in the secondary sale of their systems is that this provides the manufacturer with a direct economic incentive to excel in creating systems which last beyond the duration expected for a first use phase. By ensuring that the amount of systems with a viable second use phase is maximised, manufacturers in turn ensure a revenue stream for themselves through resale of these systems at end of life.

8.2.2 Benefits to Manufacturers

Beyond the obvious economic and environmental incentives, further benefits of adding condition monitoring technology to PCs sold under a lease-based sales model may be realised in the area of prognostic monitoring. As the system primarily provides information relating to the use of computers, and to the processes which throughout their lifetime cause their physical obsolescence, manufacturers who implement a system of this type are effectively performing a large-scale LCM experiment, gaining access to a vast amount of information related to the way in which customers use their products. Through a clearer understanding of the real environments in which PCs are operating, manufacturers may choose to adjust their marketing and their manufacturing processes to more accurately reflect the real world use profiles. Through the use of the LCM data in this way, manufacturers would have access to a source of almost real-time data which would, at very low cost, allow them to continuously adapt to the changing ways in which their customers use their computers. This could give early adopters a unique competitive advantage in developing more robust and demographic-targeted products.

In a lease-based sales context, if manufacturers are capable of analysing their products during a regular process of replacement, they are also well placed to perform any preventative maintenance deemed necessary at that juncture, rather than waiting for the occurrence of a critical failure. Additionally, in cases when a computer is being replaced, individual components which are viable for reuse can be identified and redirected for use in refurbished systems.

Under WEEE, manufacturers are required to assume responsibility for the disposal of their products at end of life. In general at the moment this is achieved through contribution to group schemes. As a result, computer manufacturers do not necessarily know exactly what proportion of the waste stream consists of their

individual systems. RFID tags have in the past been proposed for the purpose of identifying the proportion of goods which can be attributed to individual manufacturers. Introduction of this system, for identification and separation of viable computers, could have the incidental effect of identifying the brands which are being processed at end of life. By utilising this at the beginning of the waste stream, manufacturers could ascertain exactly what proportion of the waste stream consists of their products, and in this way ensure that they are paying only that proportion of the cost which they owe. In particular this benefits those manufacturers who minimise their products impact at EoU. Simply identifying their products could reduce the overall amount which they are required to pay towards end of life management compared to the current pay by weight system.

Because the waste handling process is generally destructive to the product, as it is aimed at raw material extraction, manufacturers may instead choose to recover their own waste and govern disposal themselves. By identifying their own products before they enter the waste stream, and removing them for handling, manufacturers have the opportunity to take those systems which are not viable for refurbishing and to ensure that they are handled in a way which recovers a maximum of potential value from the components. Owing to their familiarity with the components, manufacturers would be positioned well to identify the most profitable handling process, as well as that which is most environmentally benign.

8.2.3 Drawbacks

While the processes outlined above offer the advantages associated with reuse to the manufacturer, they do not inherently incentivise the first owner of a PC to utilise the computer in a way which would make it viable for reuse. Manufacturers performing scheduled maintenance may be able to anticipate failures, and even to prevent them

through provision of information to consumers about the causes of failure and their avoidance, but there is no economic incentive for consumers to behave in a manner which extends the life of the computer beyond their own needs, as any computers which fail under a lease-based sales model will be replaced by the manufacturers. One way in which this could be changed is if manufacturers were to implement a reward system, whereby discounts are offered depending on the replacement and maintenance schedule required.

8.2.4 Configuration

In a TAAS model, depending on the profile of customers, the configuration of the technology for performing the analysis of used computers can vary. For large companies for whom regularly scheduled maintenance is performed, it may be possible for technical personnel to perform analysis and maintenance in-situ, without the need to remove the computers under examination. This could be achieved using handheld readers which relay information to a central database. For predictable failures precipitated by the measured environmental factors, simply scanning a unit should be enough to ascertain the likelihood of failure, thereby highlighting systems where preventative maintenance is necessary.

For individual consumers the process for retrieving computers would be more complex, and it is probable that the computer may need to be returned to the manufacturer for servicing or replacement. In this situation, where a large volume of computers are analysed, the evaluation of PCs could be performed using a gate-type RFID reader around a conveyor belt, to perform triage and flag those systems which have exceeded a predetermined threshold of lifetime consumption. In this configuration, the RFID tag within the computer could serve a dual purpose, as it could also act as a

unique identifier for the computer, and link the unit with the customer, ensuring that systems are traceable throughout their life cycle.

8.3 WEEE Centres

While WEEE centres are at present used as the first stage in the disposal process for electronics, they exhibit some potential to be partially repurposed for the analysis of EoU computers. This could require significant renovation of existing WEEE handling infrastructure, however it would also render advantages which may not necessarily be realised through manufacturer control of EoU processing.

8.3.1 Benefits

Basing the take-back infrastructure in WEEE centres offers the advantage of utilising current legislative and infrastructural frameworks to incentivise the adoption of the technological reuse solution. By framing this solution as a part of a structure with which the public are already familiar, minimal behavioural change is necessary in order to recover a large amount of computers. Since WEEE has been implemented, the association between WEEE take-back centres and used electronic and electrical goods has become ingrained in popular consciousness, and this provides a structure on which the take-back of used PCs could be based.

One advantage of this application of the system is that it has the potential to recover the computers whose owners have not considered reuse as an option at end of life. By interposing the option for reuse at the stage in the life cycle where people would otherwise dispose of their systems in any case, it is possible to prevent the unnecessary recycling of viable systems by rerouting them to a more environmentally benign destination. Ideally, this would have the result of removing some EoL computers before they have a chance to enter the waste stream.

8.3.2 Drawbacks

Although the goal of analysing each and every computer at end of life - and recovering a theoretical maximum of these which can be reused - is desirable, consideration must be given to the fact that people may have reasonable concerns regarding what happens to their computer when it leaves their possession. As cyber-crime becomes more common, public awareness of the risks associated with their digital data increases, and computer users are becoming more cautious about any private information which could be retrieved from a used computer. As such people may be disinclined to submit their computers for reuse owing to privacy concerns. If all computers in WEEE centres are analysed for reuse potential, this has the potential to dissuade people from the use of WEEE centres for their PCs entirely, and may as such result in an increase in the amount of computers entering landfill as an inadvertent consequence.

A possible solution to this problem is to offer an economic incentive to return a computer for reuse. By offering the public the choice of reuse or recycling at the WEEE centre, and incentivising the option of reuse, while providing information on the methods used for data destruction, it may be possible to ensure that a majority of people elect to allow their systems to be reused. This would help to reduce the data protection concerns of the public, while at the same time rewarding customers for making environmentally preferable decisions.

8.3.3 Configuration

Complete implementation of this type of system at individual WEEE centres has the potential to be a costly solution to the reuse problem. At present, a majority of municipalities have at least one WEEE centre, making recycling of electrical goods accessible to everyone. Additionally, stores which sell electronic goods are required to

provide a take-back service, at present on a like-for-like basis, but this is likely to be expanded to more goods in the future. WEEE centres and take-backs in more rural areas are likely to have a lower throughput of PCs than those located in suburban or urban areas. As such, the cost of implementing the system in every WEEE centre may be unbalanced, with some centres capable of operating profitably, while others lose money. In particular, if reinstallation of operating systems and possible replacement or reinstallation of parts was to be performed at WEEE centres, the cost of labour and equipment would quickly become unreasonable for the smaller centres, and the logistics of implementing this at electronics stores as a part of existing take-back structures seem prohibitively complex.

One alternative to a distributed system of this type is a centralised one, whereby all computers collected at WEEE centres throughout a country or region are handled at a single facility. This allows economies of scale to come into effect, and higher economic efficiency can be achieved. However, it also requires the transport of a large volume of used computers between centres and the central facility, which has the potential to damage some viable systems in the process. Furthermore, the cost associated with the transport of systems which are not viable is wasted money.

A feasible middle ground is performing the analysis of computers at WEEE centres, so only those PCs and components identified as viable need to be transported to a facility for further analysis, refurbishment, and redistribution. This offers significant efficiency advantages compared to a completely distributed system, while removing the requirement for the transport of non-viable computers. The cost of implementing a national network of RFID readers at large take-back centres would be reasonable, and regional or national facilities could be established to allow quality-controlled deletion of personal information and refurbishing of used computers.

Given that this configuration would be likely to be governmentally funded, and the fact that the original owners of the PCs would receive only nominal remuneration, it is reasonable that any viable computers resulting from a system implemented in this way would become the property of the government, and would be distributed to schools, prisons and other government institutions where the information technology infrastructure does not need to be cutting-edge.

8.4 Large-scale Recycling and Refurbishing Centres

The centres currently engaged in the processes of refurbishing and recycling of computers are potentially the actors who are best-placed to perform end-of-life analysis and separation of used computers. As they have existing infrastructure for the more complex aspects of the refurbishing process, such as the destruction of private data and the installation of new licensed operating systems, a system for the early separation of viable systems would benefit them by reducing the time and effort which is spent determining whether a used computer is suitable for reuse.

8.4.1 Benefits

One advantage offered to the refurbishing centre by the adoption of this technology is that by using additional life-cycle information refurbishers would be able to select a larger range of components from used computers which could be reused. Because the information could be used to identify the exact failure mechanism of a system, unaffected components could be retrieved from non-functional assemblies. In this way, a larger proportion of their incoming stock could be profitable, leaving the refurbisher better placed to provide guarantees of reliability.

By performing a triage operation on incoming used computers, centres can focus their attention on the systems which have the most potential for profitable reuse.

Increasing the degree of certainty of a profitable return on computers handled in turn allows a greater quantity of computers to be reused, as the cost associated with the detailed analysis of ultimately unviable computers can be eliminated.

If information relating to the unique hardware configuration of the PC is included in the RFID communications, then additional benefits to the refurbishing centres can be realised. Components which are viable for reuse can be salvaged from systems which are not viable, and used in the refurbishing of other systems. Furthermore, given information on the components and materials in a system, recycling can be performed in the way which would yield the most value from recovered metals.

8.4.2 Drawbacks

Given the potential for refurbishing centres to use this system to identify the profit from reuse and directly compare this with the profit which can be realised from the retrieval of the materials involved, it is likely that centres will make an economic decision rather than an environmental one. Given the consistently increasing value of precious and rare-earth metals, some computers which could potentially be reused may have greater monetary value in recovered metals, resulting in a conflict of interests for the refurbisher. In fact, as the proposed system could be used to allow the refurbishers to quickly and accurately identify the components installed in any given system, and then directly compare the value of retrieving metals with the value of reselling the system in the second hand market, its use could potentially increase the number of PCs which are recycled at end of life, instead of reducing it.

As with the manufacturer and WEEE case, it is difficult to imagine how the refurbishing centre could incentivise the environmentally efficient use of PCs during the first life of the systems, as there is no direct link rewarding PC users for preserving the

quality of their systems. It is not impossible for a solution to be constructed which does this; however such a solution would not be an organic extension of the system, and would require some process of linking PCs to their original owners through the recycling process.

8.4.3 Configuration

At present, the primary destination for PCs emerging from refurbishing centres is government institutions and charitable causes, however if the consumer market can viably be added, the potential for profit from higher value systems is significant. This would require acquiring recognition in the consumer market as a provider of quality, low cost systems. Compared to manufacturers, refurbishing centres would have to invest significantly in the areas of advertising in order to become recognised for sale of PCs.

Logistically, the process for retrieval of used computers by refurbishers is less obvious than that for WEEE centres or for manufacturers. While companies currently provide used computers to refurbishers as part of a planned replacement system, the process for acquiring the systems from the residential market would be more complicated. This could be achieved with a system similar to the existing WEEE collection system, where PC owners register for the collection of their system, and a courier transports it to the centre. The practicality of this system increases when a refurbisher can ensure a significant volume of customers and operate their own collection service on a regular basis.

In refurbishing centres with a large volume of used computers, a potential technological configuration for RFID readers to quickly separate viable systems would be a gate type of reader placed around a conveyor belt. Using this layout, systems upon

the conveyor could be automatically scanned, and the analysed data could determine whether they are sent for reuse, refurbishment or for recycling and disposal. The conveyor could then automatically separate the systems or, alternatively, systems could be labelled with uniquely identifying bar codes, which could be cross referenced with a database storing their life cycle information to provide the information as one input into a decision-making process at a later stage in the refurbishing process.

8.5 Distributed independent take-back

Where value is introduced to the reverse logistics stream, the possibility of small enterprises taking advantage of this value must be considered. In the mobile telephone market, an industry has developed around the sale of used phones at EoU. Many companies have emerged which offer the option of returning a used phone in exchange for a fee dependent on the model and sometimes the condition of the phone in question. These enterprises have enjoyed some success, as most people do not dispose of used phones, and the possibility of selling them offers an attractive alternative.

Given that a system such as the one proposed would increase the value to be gained from EoU computers, it is probable that an equivalent industry would develop among facilities willing to analyse used computers, and offer a fee based on the value of the system in the secondary market. This would most likely develop only in the absence of a service such as one of those outlined previously which caters to the consumer market.

It is likely that this type of service would primarily target the consumer market for computers, based on the fact that a majority of larger businesses would be more likely to deal directly with manufacturers or refurbishing centres. Owing to the volume of systems which would come from large companies, it is unlikely that smaller take-

back enterprises would be equipped to handle the corporate used computer market. The lower initial capital required to become established in the consumer market would also appeal more to a smaller enterprise.

8.5.1 Advantages

The obvious advantage of a system like this is that it would offer the clearest method for rewarding customers directly for the efficient operation of their systems. By evaluating the system on the spot, the companies could pay an amount to customers based on the result of this analysis. If advertising of the service was provided in a way which highlighted the benefits of operating PCs in a way which reduces degradation, this type of service could serve as motivation for potential customers to operate their PCs in ways which extend their operational lifetimes.

8.5.2 Drawbacks

A widespread industry of this type would be difficult to regulate, and similar to the refurbisher case, smaller private enterprises are unlikely to make decisions based on environmental incentives when there is conflict with the economically superior choice. However it is also less likely that a small company would be capable of recovering enough systems to make recovery of metals a viable source of income. This may encourage such enterprises to focus on reuse rather than recycling.

In order for a PC to be securely cleared of data, it is necessary to format the hard drives, thereby clearing the operating system. As such, it would be necessary for the reinstallation of an operating system on a computer prior to resale. The cost of a new operating system would significantly increase the cost associated with the refurbishing process. While free operating systems such as those based on Linux are available, the learning curve of a new OS would most likely dissuade customers from the purchase of

a used PC. The Microsoft Refurbisher programs provide lower cost Windows licenses; however becoming a part of these programs still represents a barrier to entry into the used PC market.

Given the smaller range of computers likely to result from the market which this type of enterprise would target, it is difficult to imagine that the economies of scale which are possible given the volume of interchangeable components which refurbishers would retrieve. It's therefore likely that some of PCs which could potentially be refurbished, given the appropriate components, would instead be recycled because the correct components are not available.

8.5.3 Configuration

The logistics of a system of this kind are more difficult to predict, as the cost and difficulty of transporting desktop PCs would make the postage of used PCs - in the way that phones are currently transported for comparable services - prohibitive. Potentially, a service could be provided through local retail centres offering a cash reward based on scanning the RFID tags in computers. After reclaiming the systems, analysis and rebuilding of the systems could be performed on-site and used systems could be re-sold.

Technologically, the implementation of this type of service would be relatively simple. A handheld RFID reader could be used to perform the diagnosis of the used computer. This type of service could easily be offered over the counter assuming a reasonably small amount of customers. Based on the information from the RFID reader, in combination with the information regarding metal prices and market value of new and used computers, it is possible to gauge a fair price for the system.

The target market for computers sold by this type of enterprise would probably be the consumer demographic. As such, it would occupy the consumer-to-consumer

segment of the used computer market which is most likely to be neglected by the larger service providers. In this niche, the smaller enterprises could provide a valuable service; however it would realistically be preferable to engage manufacturers or else large refurbishers in the consumer market, as the benefits of refurbishing PCs increase proportionately to the volume of PCs being handled.

8.6 Conclusion

While it is impossible to predict every potential application of a system of this type, in this chapter the most likely manifestations of the solution are described, given the current reuse infrastructure. It should be noted that the scenarios suggested are based on the current infrastructure of Ireland, and take into account such factors as the current WEEE regulations, which are currently under revision. These scenarios are likely to be applicable to many developed nations, however. How a system of this type could be integrated in to the reuse networks of developing nations is less obvious, however given the thriving market in reused PCs in developing nations, there could certainly be potential for the implementation of some system which evaluates the functionality and reliability of used PCs.

Based on the analysis in this chapter, the case which would appear most beneficial is that of the manufacturer performing the refurbishment and resale of EoL PCs. The benefit both to customers and to the manufacturers of this structure makes a strong case for the desirability of the implementation of a system of this type at the manufacturing phase. By integrating the evaluation of used PCs into the WEEE process, reusable systems could be recovered before disposal. An additional benefit of the WEEE case is that through labelling non-viable systems as electronic waste, it can be

ensured that their export into developing nations under the auspices of reuse is prevented.

Chapter 9

CONCLUSIONS AND FUTURE WORK

9.1 Contribution

The work presented in this thesis proposes a novel application of existing technologies to enable the improvement of End of Life logistics for PCs with the goal of facilitating the development of a functional secondary market for PCs. By applying the in-situ sensors of current-generation processors and motherboards, it suggests that a prognostic health monitoring system could be applied allowing the estimation of remaining life in PCs in any of several typical end of life scenarios. This addition to PC architecture would cater for a signalling system in used PCs which is intuitive, while simultaneously accounting for the factors which most impact reliability.

This is the first application of prognostic health monitoring life consumption methodologies for a consumer market, and the low cost associated with the use of existing processor temperature sensors means that its implementation at design-phase of PCs would be a relatively inexpensive method for manufacturers to ensure that their products are reusable at end of life. This provides a quantifiable metric of design for reuse, allowing legislators and eco-labels to reward manufacturers who utilise it, which is currently lacking in a reuse context.

It is proposed that smart tags, using RFID technology, may be used to access thermal LCM data, allowing the wireless extraction of information for the resultant determination of value of a used system. RFID has the potential to enable access to information while the PC on which the information is stored is powered off, and to be

independent of operating system or system hardware. At present, however, RFID technology is not used for large volumes of data such as would be required for this type of application, instead being primarily for asset or personal identification purposes.

In order to limit the volume of information which it would be necessary to transmit in the course of LCM data extraction, data reduction techniques which are utilised in fatigue analysis are considered. The technique of ordered overall range, which reduces a time history of environmental loads to only significant reversals in direction of load variation, is examined. In combination with this, the technique of rain flow cycle counting can be used to categorise environmental load cycles based on their characteristics of amplitude, duration and mean. To store these data, a fixed volume of 142 kB was specified, which would be significantly less than would be required by continuous storage of data for a PC operating for any significant period of time. This was subsequently reduced, through consideration of real-world thermal cycles, to 32 kB of data. A simple prototype was developed which allows transmission of processor temperature through a USB interface. The prototype performs ordered overall range analysis and rain flow cycle counting on the incoming temperature data.

From an electronic hardware perspective, the most novel aspect of this application is the RFID module, which would necessarily transmit far greater volumes of data than would be possible using current RFID protocols or technologies. Central to the concept of the work is the requirement that the implementation of the resultant technology would require no major modifications to existing RFID analysis technologies, which are already common in waste-collection applications. Consequently, the process used for the communication of information was based on existing popular RFID technology, the ISO 14443 standard, in the 13.56 MHz band.

This work examines the requirements laid out in that standard from the perspective of prototype development. The design process for the prototype is explained, with particular regard to ensuring the software operates to exactly the timing specifications as laid out in the specifications of the ISO standard, and that the stipulations regarding protocol commands are followed by the prototype outside of transmission of LCM data. As such, the prototype interacts with a reader in exactly the manner dictated in the specification throughout the initial identification and anti-collision processes, and responds to all protocol commands in the same way as a RFID tag.

The design of the prototype circuit showed that the system is functional in extracting and displaying LCM data, although the PC based middleware would require some further development in order to allow the transmission time for 32kB of data to approach the ideal time of about 4 seconds.

Several ways in which a system of this type could be leveraged by the various stakeholders in the computer forward and reverse logistics industries are considered, based on existing technologies, processes and structures. There is significant scope for increased profit for either manufacturers or refurbishers, depending on their uptake of the system. There is also scope for the introduction of new stakeholders in the used PC market, and the provision of increased consumer to consumer transactions services.

9.2 Future Work

Although the solution developed in the course of this work successfully allowed the transmission of significantly greater volumes of information than would be possible using existing RFID technology, the transmission of this information required a significant period of time, and does not represent an optimal solution for the extraction

of the volume of data which was used for the development of data reduction techniques, at over a minute of analysis required per computer. However there are areas in which this performance could be improved.

The data reduction algorithm allows a generous amount of information storage, which is far more than required for the lifetime of a PC. While cycles with short periods could potentially require close to the total amount of data storage provided, longer cycles could not, so the amount of data storage allocated to those cycles could be reduced considerably. Optimisation of data storage algorithms could result in significant reductions in the volume of information which it is necessary to transmit, resulting in improvements to the time required to transmit the time-history data.

While HF RFID was selected as an appropriate technology for the purpose of the work due to its relative ubiquity, there is potential for alternative RFID technologies to allow transmission of the required amount of information in a shorter time. UHF and microwave technologies are used in an increasing number of applications, and there is a correlating reduction in the costs of the technologies, so there is potential in the near future for the application of these devices to the function of LCM data extraction for signalling purposes.

There are slight security concerns to be considered in utilisation of a solution of this type. As the application of this system could determine the outcome of financial transactions, there is incentive for individuals to modify the stored information in the memory device. If implemented, anti-tampering features would be a necessity of this device attaining public understanding and trust.

Privacy represents a further area for concern with any system whereby people's private information may be retrieved. Although access to the private data regarding

thermal profiles of a person's PC does not naturally lend itself to concern, RFID has developed a reputation for lack of security, and it would be beneficial to ensure that information could not be accessed by unsuitable persons.

While reliability is commonly considered to be improving, and research shows that the use phases of computers are decreasing, there exists a void in the literature relating to the actual impacts of environmental loads on the reliability of electronics. While the utilisation of a system such as the one outlined in this work would increase the understanding of the causes of failure, it will not gain traction in a reuse market unless it is accurate in identifying reliable PCs, which requires first a greater understanding of the effects of individual environmental loads and their interactions. By understanding this topic, improvements can be made both to the identification of reliable PCs and to manufacturing and assembly processes, further facilitating the extension of PC life. While temperature seems to be the characteristic of a PC's environment with the greatest impact, further studies are required to determine the actual effects of other factors, such as humidity, vibration and voltage levels.

Finally, as the developed world's addiction to technology is gradually shifting from the desktop PC to the notebook, the tablet and the smart phone, electronic waste will see a similar shift in the next number of years. Although these are regarded almost as "throw-away" products, and a reuse market would not necessarily emerge in the developed world, a system for evaluation of their viability could potentially be used to prevent the export of non-viable units, and also to evaluate value in the developing world, where used mobile electronics enjoy a significant market.

APPENDICES

Appendix A

RFID CODE SOLUTION

1. Final application code for ATmega16

```

/*
 * RFIDProgram1.c
 *
 * Created: 20/07/2011 13:19:35
 * Author: Eanna Cronin
 */

#define ATQA_H 0
#define ATQA_L 1
#define WUPA 0b1010010
#define RATS 0b11100000
#define REQA 0b0100110
#define SENDinf 0b11001010
#define UID_0 0b00000111
#define UID_1 0b01010101
#define UID_2 0b01010101
#define UID_3 0b01010101
#define F_CPU 1356000UL
#include <avr/io.h>
#include "CODEhder.h"

static inline void asm_RX () {
/*This section of code performs the receiving of data from an RFID reader, and is coded in assembly language*/
asm volatile (" .set R0,0 ;get register name ");
asm volatile (" .set temp,16 ;get register name ");
asm volatile (" .set mC1,17 ;get register name ");
asm volatile (" .set mC0,18 ;get register name ");
asm volatile (" .set mC2,19 ;get register name ");
asm volatile (" .set mC3,20 ;get register name ");
asm volatile (" .set mCS,21 ;get register name ");
asm volatile (" .set par,23 ;get register name ");
asm volatile (" .set par1,25 ;get register name ");
asm volatile (" .set YL,28 ;get register name ");
asm volatile (" .set YH,29 ;get register name ");
asm volatile (" .set ZL,30 ;get register name ");
asm volatile (" .set ZH,31 ;get register name ");
asm volatile (" .set PINB,22 ;get register name ");

asm volatile (" push ZH "); //preserve data stored in registers
asm volatile (" push ZL ");
asm volatile (" push YH ");

```

```

asm volatile ("      push YL          ");
asm volatile ("      push temp        ");
asm volatile ("      push mC1         ");
asm volatile ("      push mC0         ");
asm volatile ("      push mC2         ");
asm volatile ("      push mC3         ");
asm volatile ("      push mCS         ");
asm volatile ("      push par         ");
asm volatile ("      push par1        ");

asm volatile ("BEGIN: ldi    YL,lo8(IObuffer); Y points to next byte");
asm volatile ("      ldi    YH,hi8(IObuffer)          ");
asm volatile ("      st     Y+, 0      "); //Clear data in current address
asm volatile ("      clr    mC0         ");
asm volatile ("      ldi    par1 ,1    "); //preset parity check bit
asm volatile ("      clr    mC1         ");
asm volatile ("      ldi    mC2, 10    "); //present bit count
asm volatile ("      clr    mC3         ");
asm volatile ("      ldi    mCS, 8     "); //preset sample count
asm volatile ("      ldi    par, 1     ");

asm volatile ("disc:  sbis    PINB, 4   ");
asm volatile ("      rjmp   disc        "); //discard leading 0s and 1s
asm volatile ("disc2: sbic    PINB, 4   "); //maximum length of time between
                                           pin going low and first sample: 5
                                           clocks

asm volatile ("      rjmp   disc2       ");

asm volatile ("SAMPLE: sbis    PINB, 4   "); //1/2: if the bit is set(1), don't
                                           clear mC1
asm volatile ("      clr    mC1         "); //2: Otherwise
asm volatile ("      dec    mCS         "); //3 decrement the sample count
asm volatile ("      breq   NEWS        "); //4/5 If zero
asm volatile ("      nop                    "); //5
asm volatile ("      tst    mC1         "); //6: If mC1 was cleared(sample was
                                           0) previously then:
asm volatile ("      breq   ZEROS       "); //7/8 Go to the zeros section
asm volatile ("      inc    mC1         "); //8 Otherwise increment mC1(if
                                           the last sample was 1)

asm volatile ("      clr    mC0         "); //9 Clear count of 0s
asm volatile ("      nop                    "); //10 Do nothing
asm volatile ("      nop                    "); //11
asm volatile ("      nop                    "); //12
asm volatile ("      nop                    "); //13
asm volatile ("      nop                    "); //14
asm volatile ("      rjmp   SAMPLE      "); //16

asm volatile ("ZEROS:  nop                    "); //9
asm volatile ("      inc    mC0         "); //10 Increment the number of 0
                                           samples count
asm volatile ("      cpi    mC0, 4     "); //11 If it's higher than 4
                                           samples(at least 49 clocks, 3.6us)

```

```

asm volatile("          brsh   ENDO      "); //12/13 Go to the end, as signal has
                                been lost
asm volatile("          inc     mC1      "); //13 Otherwise increment mC1, so
                                if next sample is 1 there will
                                already be a value there)
asm volatile("          nop                    "); //14
asm volatile("          rjmp   SAMPLE    "); //16 Go back for next sample

//STORE code decodes last received bit and also adds the entire last byte to memory
asm volatile("STORE:    ldi     mC2, 9   "); //10//reset bit count
asm volatile("          cpi     mC1, 12  "); //11
asm volatile("          brsh   EQU      "); //12/13: Tests to see if the bit
                                received last was a parity bit or
                                end bit; greater than 11 successive
                                high samples indicates end bit
asm volatile("          cpi     mC1, 4   "); //13: Greater than 4 samples
                                indicates last bit received was 0
asm volatile("          brsh   SKIP     "); //14/15
asm volatile("          eor     par, par1 "); //15 if last bit was 1, add to parity
asm volatile("SKIP:    ldi     mCS, 7   "); //16 Reset sample count
asm volatile("          sbis   PINB, 4   "); //1/2 Ran over time, so perform
                                next sample
asm volatile("          clr     mC1      "); //2
asm volatile("          eor     par, par1 "); //3 Test parity
asm volatile("          breq   FIN      "); //4/5: Parity bit should not be set.
                                Either it's encountered 0 or an
                                even number of ones, in which case
                                the parity was 0, either way, error!
asm volatile("          st      Y+, temp "); //6 If no error, store last byte in
                                memory and increment address
asm volatile("          clr     temp     "); //7
asm volatile("          inc     mC3      "); //8 Increment the amount of data
                                stored count
asm volatile("          tst     mC1      "); //9 Same as the start here, check
                                if signal last sampled was 1 or 0
asm volatile("          brne   ONES     "); //10/11
asm volatile("          inc     mC1      "); //11
asm volatile("          inc     mC0      "); //12
asm volatile("          cpi     mC0, 4   "); //13
asm volatile("          brsh   BEGIN    "); //14/15
asm volatile("          rjmp   SAMPLE    "); //16 and back to the start for the
                                next sample

//The NEWS section decodes the last received bit in the temp memory register
asm volatile("NEWS:    ldi     mCS, 8   "); //6
asm volatile("          dec     mC2      "); //7 Decrement the bit count
asm volatile("          breq   STORE    "); //8/9 If it has reached 0, store
                                byte in buffer
asm volatile("          cpi     mC1, 4   "); //9: If less than 4 successive 1's
asm volatile("          brlo   ADD1     "); //10/11 Bit was a 1
asm volatile("          ror     temp     "); //11: Otherwise, 0. Carry already 0,
                                so rotate that in

```

```

asm volatile("          cpi      mC1, 12  "); //12 if we had more than 12 1s
asm volatile("          brsh    EQU      "); //13/14 End bits, so end function
asm volatile("          inc     mC1      "); //14 Increment mC1 as before
asm volatile("          rjmp    SAMPLE   "); //16 and sample again

asm volatile("ADD1:     ror     temp     "); //12 Carry is a 1, so rotate it into
                                     next position
asm volatile("          inc     mC1      "); //13 Increment 1's count
asm volatile("          eor     par, par1"); //14 Update parity
asm volatile("          rjmp    SAMPLE   "); //16 And sample again

asm volatile("ONES:    inc     mC1      "); //12 Increment the 1's count,
asm volatile("          nop     "); //13
asm volatile("          nop     "); //14
asm volatile("          rjmp    SAMPLE   "); //16 And sample again

asm volatile("END0:    clr     mC0      ");
asm volatile("          sbr     mC0, 1   ");
asm volatile("          rjmp    OUT      ");

asm volatile("EQU:     lsl     mC3      "); //16: In the normal sequence, this
                                     would be the end of the last sample
asm volatile("          lsl     mC3      "); //1: Multiply mC3 by 8 to change
                                     bytes received to bits
asm volatile("          lsl     mC3      "); //2
asm volatile("          ori     mC3, 7   "); //3: and add 7
asm volatile("          cpi     mC2, 9   "); //4:if mC2 is 9 exactly, then the
                                     data has not been stored yet
asm volatile("          breq    FIN1     "); //5/6 So store it
asm volatile("          cpi     mC2, 8   "); //17
asm volatile("          breq    FIN      "); //18/19: can only be 8 in error, as
                                     this would mean that one of the end
                                     bits as already been stored as a
                                     parity bit
asm volatile("          sub     mC3, mC2"); //for values of mC2 between 1 and
                                     7, number of valid bits is 7-mC2
asm volatile("FIN1:    st      Y, temp  "); //store outstanding data in
                                     IObuffer

asm volatile("FIN:     ldi     YL,lo8(IObuffer) ");
asm volatile("          ldi     YH,hi8(IObuffer) ");
asm volatile("          st      Y,mC3    "); // indicate the total bits received
                                     in IObuffer[0]

asm volatile("OUT:    pop     par1     ");
asm volatile("          pop     par      ");
asm volatile("          pop     mCS     ");
asm volatile("          pop     mC3     ");
asm volatile("          pop     mC2     ");
asm volatile("          pop     mC0     ");
asm volatile("          pop     mC1     ");
asm volatile("          pop     temp    ");

```

```

asm volatile ("      pop YL          ");
asm volatile ("      pop YH          ");
asm volatile ("      pop ZL          ");
asm volatile ("      pop ZH          ");
}

static inline void asm_TX() {
asm volatile (" .set    R0, 0      ; get register name ");
asm volatile (" .set    temp, 16 ; get register name ");
asm volatile (" .set    mC1, 17 ; get register name ");
asm volatile (" .set    mC0, 18 ; get register name ");
asm volatile (" .set    mC2, 19 ; get register name ");
asm volatile (" .set    mC3, 20 ; get register name ");
asm volatile (" .set    mCS, 21 ; get register name ");
asm volatile (" .set    par, 23 ; get register name ");
asm volatile (" .set    par1, 25 ; get register name ");
asm volatile (" .set    YL, 28 ; get register name ");
asm volatile (" .set    YH, 29 ; get register name ");
asm volatile (" .set    PINB, 22 ; get register name ");
asm volatile (" .set    PORTB, 24; get register name ");

asm volatile ("      push YH          ");
asm volatile ("      push YL          ");
asm volatile ("      push temp        ");
asm volatile ("      push mC1        ");
asm volatile ("      push mC0        ");
asm volatile ("      push mC2        ");
asm volatile ("      push mC3        ");
asm volatile ("      push mCS        ");
asm volatile ("      push par         ");
asm volatile ("      push par1        ");

asm volatile ("      ldi    YL, lo8 (IObuffer) ");
asm volatile ("      ldi    YH, hi8 (IObuffer) ");

asm volatile ("      ld      mC0, Y+  ");
asm volatile ("      inc    mC0       ");
asm volatile ("      ldi    mC1, 0    ");
asm volatile ("      ldi    mC2, 1    ");
asm volatile ("      ldi    mC3, 4    ");
asm volatile ("      ldi    par, 1    ");
asm volatile ("      ldi    par1, 1   ");

asm volatile ("      cbi    PORTB, 0  "); //2
asm volatile ("      rjmp   DY1       "); //4

asm volatile ("SBIT:  dec    mC1      "); //1: decrement the bits in byte
                                counter
asm volatile ("      breq   EIGHTH    "); //2/3: if zero, last bit in byte, add
                                parity
asm volatile ("      brmi  NEXT       "); //3/4: if negative, parity has
                                already been added, get next byte

```

```

asm volatile("          ldi      mCS, 14  ") ; //4: Otherwise, wait out rest of
                                unmodulated half-bit
asm volatile("          rcall    DELAY   ") ; //52: Delay = 48 clocks
asm volatile("          nop      ") ; //53
asm volatile("          dec      mC0     ") ; //54: decrement total bit counter
asm volatile("          breq     LASTBIT ") ; //55/56: if neg, add end bit
asm volatile("          ror     temp    ") ; //56
asm volatile("          brcc    AZERO   ") ; //57/58: if carry is set, next bit is
                                one
asm volatile("AONE:     eor      par, par1") ; //58: Next bit is a one, update
                                parity
asm volatile("          tst      mC2     ") ; //59: check current bit
asm volatile("          brne    SAME    ") ; //60/61 if one, next bit is the same
asm volatile("          ldi     mC2, 1   ") ; //61: Otherwise, 0,1
asm volatile("          ldi     mC3, 8   ") ; //62
asm volatile("          rjmp    MOD1    ") ; //64

asm volatile("SAME:    ldi      mC3, 4   ") ; //62: last two bits were the same ,
                                four modulation cycles to next un-
                                modulated
asm volatile("          nop      ") ; //63
asm volatile("MODU:    nop      ") ; //modulates number of times
                                specified by mC3
asm volatile("MOD1:    cbi      PORTB, 0 ") ;
asm volatile("          rjmp    DY1     ") ;
asm volatile("MOD2:    sbi      PORTB, 0 ") ; //2
asm volatile("          nop      ") ; //3
asm volatile("          nop      ") ; //4
asm volatile("          dec     mC3     ") ; //5: Required number of
                                modulations completed?
asm volatile("          brne    MODU    ") ; //6/7: If not, modulate again
asm volatile("          rjmp    SBIT    ") ; //8: If finished, back to start.

asm volatile("EIGHTH:  ldi      mCS, 14  ") ; //4: load delay timer
asm volatile("          rcall    DELAY   ") ; //52: Delay = 48 clocks
asm volatile("          nop      ") ; //53
asm volatile("          nop      ") ; //54
asm volatile("          tst     par     ") ; //55: If par is 1
asm volatile("          brne    AONE    ") ; //56/57: Jump to AONE, adds a 1
                                as parity bit
asm volatile("          ldi     par, 1   ") ; //57: Otherwise make par 1
asm volatile("          nop      ") ; //58: and add a zero below

asm volatile("AZERO:   tst      mC2     ") ; //59: AZERO function: adds a zero
                                as the next bit to transmit. First,
                                check current bit
asm volatile("          breq     SAME    ") ; //60/61:if 0, next bit is the same
asm volatile("          ldi     mC2, 0   ") ; //61: Otherwise 1,0
asm volatile("          nop      ") ; //62
asm volatile("          rjmp    SBIT    ") ; //64

asm volatile("NEXT:    dec      mC0     ") ; //5

```

```

asm volatile("          breq  LBDY      "); //6/7: if mC0 is 0, last bit
asm volatile("          ld      temp, Y+ "); //8: load the next byte to be
                                     transferred from memory
asm volatile("          cpi     mC0, 7   "); //9: see how many bits remain
asm volatile("          brlo   LAST8    "); //10/11: if less than 8,
asm volatile("          ldi    mCS, 11  "); //11
asm volatile("          rcall  DELAY    "); //50: Delay = 39 clocks
asm volatile("          nop      "); //51
asm volatile("          nop      "); //52
asm volatile("          ldi    par, 1   "); //53: Reset parity bit
asm volatile("          ldi    mC1, 8   "); //54: set the bits in byte counter
asm volatile("          ror     temp    "); //55
asm volatile("          brcs   AONE     "); //56/57
asm volatile("          rjmp   AZERO    "); //58

asm volatile("LBDY:     ldi     mCS, 13  "); //8: Last bit delay
asm volatile("          rcall  DELAY    "); //53: Delay = 45 clocks
asm volatile("          nop      "); //54
asm volatile("          nop      "); //55
asm volatile("LASTBIT0: nop      "); //56
asm volatile("LASTBIT: nop      "); //57
asm volatile("          nop      "); //58
asm volatile("          nop      "); //59
asm volatile("          ldi    mC3, 4   "); //60
asm volatile("          tst    mC2     "); //61
asm volatile("          breq   LMODU   "); //62/63
asm volatile("          rjmp   END     "); //64

asm volatile("LAST8:   ldi     mC1, 7   "); //12
asm volatile("          sub    mC1, mC0 "); //13
asm volatile("          mov    mC3, mC0 "); //14
asm volatile("          inc    mC3     "); //15
asm volatile("EQU1:    rol     temp    ");
asm volatile("          dec    mC1     ");
asm volatile("          brne   EQU1    ");
asm volatile("EQU2:    nop      ");
asm volatile("          dec    mC3     ");
asm volatile("          brne   EQU2    "); //45
asm volatile("          rcall  DY3     "); //48
asm volatile("          ldi    mC1, 8   "); //53: set the bits in byte counter
asm volatile("          rol     temp    "); //54
asm volatile("          brcc   THIS    "); //55/56
asm volatile("          rjmp   AONE    "); //57

asm volatile("THIS:    rjmp   AZERO    "); //58

asm volatile("DY1:     nop      ");
asm volatile("          nop      ");
asm volatile("          rjmp   MOD2    "); //delay equal to six clock cycles

asm volatile("DELAY:   dec     mCS     "); //delay equates to 3*(mCS+1)
                                     clocks

```

```

asm volatile("      brne   DELAY   ");
asm volatile("      ret     ");

asm volatile("DY3:   ret     "); //52

asm volatile("DY2:   nop     "); //5
asm volatile("      nop     "); //6
asm volatile("      rjmp   LMOD2  "); //8

asm volatile("LMODU:  nop     "); //modulates number of times
                          specified by mC3
asm volatile("LMOD1:  cbi     PORTB, 0 "); //2
asm volatile("      rjmp   DY2    "); //4
asm volatile("LMOD2:  sbi     PORTB, 0 "); //2
asm volatile("      nop     "); //3
asm volatile("      nop     "); //4
asm volatile("      dec    mC3    "); //5
asm volatile("      brne   LMODU  "); //6/7
asm volatile("      nop     "); //7
asm volatile("      nop     "); //8

asm volatile("END:   ldi     mCS, 40 "); //delay = 126 clocks to ensure end
                          bit is sent.
asm volatile("      nop     ");
asm volatile("      nop     ");
asm volatile("      rcall  DELAY  ");
asm volatile("      sbi   PORTB, 0 "); //resets PINBO to 1, end of
                          transmission

asm volatile ("      pop par1   ");
asm volatile ("      pop par    ");
asm volatile ("      pop mCS    ");
asm volatile ("      pop mC3    ");
asm volatile ("      pop mC2    ");
asm volatile ("      pop mC0    ");
asm volatile ("      pop mC1    ");
asm volatile ("      pop temp   ");
asm volatile ("      pop YL     ");
asm volatile ("      pop YH     ");
}

long int CRCreg;
uint8_t state;
uint8_t SAMPLEbitMASK, TXptr, RXerror, IObuffer[20];
int numbits;

void CRCinit() { CRCreg=0x6363 ;}

void CRCchk() {
    /*At the end of a data block, this checks the CRC to ensure no bits were received in error*/
    if ( CRCreg !=0 ) { RXerror |=0x20 ; } ;
}
/*-----
Function: updateCRC

```

Inputs: uint16_t ch

Purpose: Updates both the receiving and the transmitting CRC bytes. Can therefore be used both to calculate the correct CRC bytes to append onto the outgoing signals, and to check incoming data packages for errors

```
-----*/
void updateCRC(uint16_t ch) {
    ch=ch ^ (CRCreg) ;
    ch = ch ^ ( ch<<4 ) ;
    ch = ch & 0x00ff ;
    CRCreg=(CRCreg>>8) ^ (ch<<8) ^ (ch<<3) ^ (ch>>4) ;
}
/*-----
```

Function: appendCRC

Inputs: none

Purpose: Calculates cyclic redundancy check bytes for amount of data specified in IObuffer[0] and appends to IObuffer

```
-----*/
void appendCRC() {
    int k ;
    long int ch;
    k=IObuffer[0] ;
    CRCinit() ;
    TXptr=1 ;
    while(k){
        ch= IObuffer[TXptr++] ^ (CRCreg) ;
        ch = ch ^ ( ch<<4 ) ;
        ch = ch & 0x00ff ;
        CRCreg=(CRCreg>>8) ^ (ch<<8) ^ (ch<<3) ^ (ch>>4) ;
        k -= 8 ;
    } ;
    IObuffer[TXptr++]= CRCreg & 0x00ff ;
    IObuffer[TXptr++]=(CRCreg>>8) & 0x00ff ;
    IObuffer[0] += 16 ;
}
/*-----
```

Function: EEPROM_read

Inputs: int uiAddress

Purpose: Accesses the EEPROM at the specified address and returns data

```
-----*/
int EEPROM_read(int uiAddress)
{
    while(EECR & (1<<EEWE)); /* Wait for completion of previous write */
    EEAR = uiAddress; /* Set up address register */
    EECR |= (1<<EERE); /* Start eeprom read by writing EERE */
    return EEDR; /* Return data from data register */
}
/*-----
```

Function: eqdelay

Inputs: int l

Purpose: Iteratively performs the number of loops of the integer passed. For delay between packages transmission

```
-----*/
void eqdelay(int l){
    while(--l){
        asm volatile("nop");
        asm volatile("nop");
    }
}
/*-----
```

```

        asm volatile("nop");
        asm volatile("nop");
        asm volatile("nop");
    }
    return;
}

/*-----
Function: transData
Inputs: none
Purpose: This section is responsible for the physical transmission of the information to the PCD. It must
compute and add the epilogue field, containing the CRC bytes,
-----*/

void transDATA() {
    int i = 0, j;
    int MEMAddress=0;
    long int ch;
    CRCreg = 0x6363;
    ch = 0;
    IObuffer[1] = 0x12;
    IObuffer[0] = 8;
    ch= IObuffer[1] ^ (CRCreg ) ;
    ch = ch ^ ( ch<<4 ) ;
    ch = ch & 0x00ff ;
    CRCreg=(CRCreg>>8) ^ (ch<<8) ^ (ch<<3) ^ (ch>>4) ;
    for (j = 2; j<=16; j++){ //transmit 15 bytes from memory per frame
        while(EECR & (1<<EWE)); // Set up address register
        EEAR = MEMAddress++; // Start eeprom read by writing EERE
        EECR |= (1<<EERE);
        IObuffer[j] = EEDR;
        IObuffer[0] += 8;
        ch= IObuffer[j] ^ (CRCreg ) ;
        ch = ch ^ ( ch<<4 ) ;
        ch = ch & 0x00ff ;
        CRCreg=(CRCreg>>8) ^ (ch<<8) ^ (ch<<3) ^ (ch>>4) ;
    }
    IObuffer[j++]= CRCreg & 0x00ff ;
    IObuffer[j]=(CRCreg>>8) & 0x00ff ;
    IObuffer[0] += 16 ;
    asm_TX();
    asm_RX();
    while(i <= 1000 && IObuffer[1]==0x13) {
        CRCreg = 0x6363;
        ch = 0;
        if(i<1000)
            IObuffer[1] = 0x12;
        else
            IObuffer[1]=0x16;
        IObuffer[0] = 8;
        ch= IObuffer[1] ^ (CRCreg ) ;
        ch = ch ^ ( ch<<4 ) ;
        ch = ch & 0x00ff ;
        CRCreg=(CRCreg>>8) ^ (ch<<8) ^ (ch<<3) ^ (ch>>4) ;
    }
}

```

```

        for (j = 2; j<=16; j++){           //transmit 15 bytes from memory per frame
            while(EECR & (1<<EWE));       // Set up address register
            EEAR = MEMAddress++;           // Start eeprom read by writing EERE
            EECR |= (1<<EERE);
            IObuffer[j] = EEDR;
            IObuffer[0] += 8;
            ch= IObuffer[j] ^ (CRCreg ) ;
            ch = ch ^ ( ch<<4 ) ;
            ch = ch & 0x00ff ;
            CRCreg=(CRCreg>>8) ^ (ch<<8) ^ (ch<<3) ^ (ch>>4) ;
        }
        IObuffer[j++]= CRCreg & 0x00ff ;
        IObuffer[j]=(CRCreg>>8) & 0x00ff ;
        IObuffer[0] += 16 ;
        asm_TX();
        asm_RX();
        i++;
    }
    return;
}
/*-----
Function: fullTXRX
Inputs: none
Purpose: Performs the complete transmission and receiving process, so that we don't have to call all of the
individual functions each time we want to transmit or receive
Changes: IObuffer should contain bytes to be transmitted for TXconvert to work
-----*/

void fullTXRX() { //
    appendCRC(); //takes whatever is in IObuffer, generates a CRC, and appends it to the data
    asm_TX(); //Change what's in TXbyteBUFFER into the correct form for transmission and
store in IObuffer
    asm_RX(); //Send the contents of IObuffer
    return;
}

int main(void)
{
    DDRB = 0b11100001;
    PORTB = 0b00100001;

    int CID, FSDI, PPSS;
    state = 0;
    while(1)
    {
        while(state == 0) { //In IDLE or Power OFF state
            PORTB = 0b01000001;
            asm_RX(); //should be the receipt of Request Acknowledge (REQA) or Wake
Up (WUP) command, both only seven bits plus the start and end bits
            if ((IObuffer[1] == REQA) || (IObuffer[1] == WUPA))
                state = 1;
            else state=0;
        }
    }
}

```

```

        while (state == 1) { //In READY1 state - received REQA, have to
send ATQA, then expecting ACK
            PORTB = 0b01100001; //light the LEDs
            /*ISO 14443-3 6.2.3 READY state
This state is entered as soon as a valid REQA or WAKE-UP message has been received and exited when the
PICC is
selected with its UID. In this state either the bit frame anticollision or other optional anticollision method
can be applied.
Cascade levels are handled inside this state to get all UID CLn.*/
            IObuffer[1] = ATQA_L; //ATQA needs to be defined in the header file as
0000000000000001
            IObuffer[2] = ATQA_H; //needs to arrive 1236/13560000(91.15us) after the
last modulation on the other side
            IObuffer[0] = 16; //to this point, from a successful REQA receipt, 76 clocks will
have passed. We want 960
            eqdelay(91); // takes (43*19) + 30 = 847 clocks
            asm_TX(); //until the first modulation is applied, takes 37 clocks
            asm_RX();
            eqdelay(91);
            if((state == 1) && (IObuffer[1] == 0x93) && (IObuffer[2]
== 0x20)) /*not supporting bit frame anticollision at this point, so only accepting 9320, meaning transmit
the entire UID, and 9370, meaning received entire UID ok*/
            {
                IObuffer[1] = UID_0;
                IObuffer[2] = UID_1;
                IObuffer[3] = UID_2;
                IObuffer[4] = UID_3;
                IObuffer[5] = ((UID_0 ^ (UID_1 ^ (UID_2 ^ (UID_3))));
                IObuffer[0] = 40;
                asm_TX();
                asm_RX();
            }
            else {state = 0;} //If bits received are not
            if((state == 1) && (IObuffer[1] == 0x93) && (IObuffer[2]
== 0x70)) { //this will actually have transmitted the entire UID and BCC and CRC bits
                if((IObuffer[3] == UID_0) && (IObuffer[4] == UID_1) &&
(IObuffer[5] == UID_2) && (IObuffer[6] == UID_3)) {
                    /*Next, Select Acknowledge must be transmitted by the PICC. This is defined as xx1xx0xx*/
                    IObuffer[1] = 0b00110000;
                    IObuffer[0] = 8;
                    eqdelay(91);
                    appendCRC(); //takes whatever is in IObuffer, generates a CRC, and
appends it to the data
                    asm_TX(); //Change what's in TXbyteBUFFER into the correct
form for transmission and store in IObuffer
                    state = 2;
                }
            }
            else {state = 0;}
        }
    }
    while (state == 2) { //in ACTIVE state
        PORTB = 0b10000001; //light the LEDs
        /*ISO 14443-3 6.2.4 ACTIVE state

```

```

This state is entered by selecting the PICC with its complete UID.*/
asm_RX ();
eqdelay (91);
//now transparent data may be communicated.
switch (IObuffer[1]){
    case SENDinf:{
        transDATA ();
        break;}
    case 0x30:{
        IObuffer[1]=0x49;
        IObuffer[2]=0x68;
        IObuffer[3]=0x65;
        IObuffer[4]=0x61;
        IObuffer[5]=0x72;
        IObuffer[6]=0x64;
        IObuffer[7]=0x00;
        IObuffer[8]=0x74;
        IObuffer[9]=0x68;
        IObuffer[10]=0x65;
        IObuffer[11]=0x72;
        IObuffer[12]=0x65;
        IObuffer[13]=0x00;
        IObuffer[14]=0x77;
        IObuffer[15]=0x61;
        IObuffer[16]=0x73;
        IObuffer[0]=128;
        appendCRC ();
        asm_TX ();
        break;}
    case REQA:
    case WUPA:
        state=0;
        break;
    default:
        {break;}
}
}
while(state == 3){ //in HALT state
    PORTB = 0b10100001;
    if (IObuffer[1] == WUPA) {
        IObuffer[1] = ATQA_L; //ATQA is defined in the header file as
0000000000000001
        IObuffer[2] = ATQA_H;
        IObuffer[0] = 16
        eqdelay (45) ;// takes (45*18) + 30 = 840 clocks
        asm_TX (); //until the first modulation is applied, takes 37 clocks
        asm_RX ();
        eqdelay (33);
        if((state == 3) && (IObuffer[1] == 0x93) &&
(IObuffer[2] == 0x20))
        {
            IObuffer[1] = UID_0;
            IObuffer[2] = UID_1;

```

```

        IObuffer[3] = UID_2;
        IObuffer[4] = UID_3;
        IObuffer[5] = ((UID_0)^(UID_1)^(UID_2)^(UID_3));
        IObuffer[0] = 40;
        asm_TX();
        asm_RX();
    }
    else {state = 0;} //If bits received are not
    if((state == 1) && (IObuffer[1] == 0x93) &&
(IObuffer[2] == 0x70)){ //this will actually have transmitted the entire UID and BCC and CRC bits
        if((IObuffer[3] == UID_0) && (IObuffer[4] ==
UID_1) && (IObuffer[5] == UID_2) && (IObuffer[6] == UID_3)){
            /*Next, Select Acknowledge must be transmitted by the PICC. This is defined as xx1xx0xx*/
            IObuffer[1] = 0b00100000;
            IObuffer[0] = 8;
            eqdelay(32);
            fullTXRX();
            if ((state != 0) && (IObuffer[1] = 0xE0))
//NB... The response to RFsend in this case should be the RATS, and is handled in state 2
                {state = 2;}
            else {state = 0;}
        }
        else {state = 0;}
    }
}
}
return 0;
}

```

2. Previous code implementation

In the original iteration of the code for the PICC, it was envisioned that the only function performed by the assembly section of the code would be the physical transmission and receipt of data. This was because using assembly code for more complex functions requires an exponential amount of coding, as each command performs only one function, and while assembly language theoretically provides all of the same functionality of higher languages, the optimisation performed during code compilation can normally be expected to result in more efficient assembly code than that which is directly written by a programmer. By performing only the absolutely necessary commands using assembly, the amount of coding required was greatly reduced.

Under the MIFARE protocol, it was necessary to code the transmission of a '1' or a '0' differently. From the standard, a '1' would be signalled by modulating the first half of the bit period with a subcarrier frequency of one sixteenth of the carrier frequency, 847.5kHz, and a '0' would have the second half of its bit interval modulated with the same subcarrier. In order to remove the necessity of encoding the data in the assembly code, this was performed in C by iterating through the data to be transmitted, examining each bit and encoding it depending on the amount of unmodulated time to the next bit. In the case where two similar bits occur, the unmodulated time is a half bit, so a one is stored in the relevant position; when a '1' is followed directly by a '0', a full bit length is unmodulated, denoted by the number zero; and when a '0' is followed by a '1' no modulation occurs between the bits, so the number two is stored in the byte.

The function `TXconvert` was written in C which, depending on the sequence of bits occurring, stored a code in a buffer. This was done by iteratively examining each bit in the output queue, and passing it to another function called `TXemit`. `TXemit` would then, depending on the current bit and the previous bit, store the value one, two or zero in the output buffer. This represented the number of modulated half bits the PICC was to transmit.

```
void TXconvert (uint8_t RXbytes) {
  /*This should convert the number into the modulation pattern, and also append the odd parity bits*/

  int TXbytePTR=0 ;
  int parity = 1;
  uint8_t mask = 1;
  MODtime = 0;
  TXptr = 0;

  IObuffer[TXptr++] = numbits;//sets the first item in IObuffer to the total number of bits
  which are to be transmitted
  while (numbits>0) {
    numbits-- ;
    if ( TXbyteBUFFER[TXbytePTR]&mask ) {
      TXemit(1) ;
      parity = parity ^ 1;}
    else {
      TXemit(0) ; } ;
  }
```

```

    mask = mask << 1;
    if (mask==0 ) {
        mask=1 ;
        TXemit (parity);
        TXbytePTR++;
        parity = 1;
    } ;
    TXemit(0) ; // 0 is the last bit to be transmitted
    } ;
IObuffer[TXptr++]=0 ; // end of list
IObuffer[TXptr++]=RXbytes ;
return;
}

```

```

void TXemit(uint8_t b){
    if (b==0) {
        IObuffer[TXptr++] = MODtime;
        MODtime = 1;}
    else {
        MODtime += 1;
        IObuffer[TXptr++] = MODtime;
        MODtime = 0;}
    return;
}

```

It was then possible, using the assembly code `asm_TX` function, to encode this information into the magnetic field from the PCD. While the PICC outputs were unchanging during the unmodulated sections of the transmission, the next address in the buffer was accessed, so that the PICC was prepared to transmit the next section of modulated bits. After the specified number of modulated half-bits had been transmitted, the PICC transmitted one unmodulated half-bit before determining the next action. If the current buffer contained zero, it immediately transmitted another half bit of unmodulated signal, while if it contained a one or a two it transmitted one or two half bits, respectively, of modulated signal, then accessed the next position in the buffer.

As well as encoding the data to be sent, the C code in the `TXconvert` function also added parity bits to the end of each byte, and kept track of the total number of bits which were being transmitted. This value was accessed by the assembly code along with the output data buffer, so that the program could stop transmitting the contents of the data buffer at the correct time, and to start receiving data from the PCD.

Similar to transmission, in the receiving section it was decided to simply sample the incoming data signal using the assembly section of code, and to perform all subsequent analyses of the received signals in the C code. By sampling the incoming signal at a rate of eight samples per bit, the signal could be stored in the incoming data buffer at one byte of sample data per bit of information. After the receipt of all of the information, it would then be possible to decode the data using C.

Subsequent to the receipt of a complete packet of data, it was necessary to decode the information. The C functions `RXsamplesTObits`, `GETsample`, and `quart_bit` performed this function by considering the number of sequential '1's received and the last bit which was received. On detecting a pause of one to three sequential low readings in the data, it counted the number of high samples which occurred before the next pause. From the RFID standard, it was known that a logical '1' was represented by a pause occurring after half of a bit interval, and a zero is either represented by a bit interval without a pause (in the case of the first zero after a one) or a pause at the beginning of the bit interval (in the case of the second and subsequent zeros). Therefore, assuming a pause of one, two or three samples, if there are seven, six or five high samples, respectively, before the next pause, then it is known that the bit received was the same as the previous one. Each possible sequence of bits was considered, and in this way the reconstruction of the original data was possible, based on the knowledge that the first bit in any sequence transmitted is always '0'. The possible bit sequences and their corresponding sample counts are shown in **Error! eference source not found.** The `RXsamplesTObits` function also performed some error checking, and if it received too long a string of '0's or '1's it would exit with the relevant error code.

```

void RXsamplesTObits(int BYTESEXpected) {
    int kk ;
    sampleINIT();
    //for ( kk=0 ; kk<100 ; kk++){ RXbitBUFFER[kk]=0 ; } // erase old contents (Takes 220us+)

    BITSinBUFFER=0 ;
    prev_BIT = 0 ;
    kk=0 ; //set
    to 0 as the first bit transferred in any signal is the "Z" signal, so while this will be removed, it
    determines what the next bits are by the time gap between them
    Ibytes=BYTESEXpected;

    while ( GETsample()==0 ) { //discard the initial low caused by the Z signal that
    precedes communications from the reader
        kk++ ; if (kk>6 ) { RXerror |= 2 ; return ; } ;
    } ;

    int BITSEXpected=9*BYTESEXpected+1 ;
    // first 1 found , this is the remainder of the Z signal from before;
    while (BITSinBUFFER<BITSEXpected) {
        kk=1;
        while ( GETsample()==1 ) {
            //for all sequential ones, just counts them. Should be no more than 7 quarter bits (28 samples)
            kk++;
            if (kk>30) { //addresses too long a string of ones
                if (BITSinBUFFER==BITSEXpected) {return;}
            }
            //if we have all the data expected, fine
            else {RXerror |=2 ; return ; } ;
            //otherwise report error and return
        } ;
    }
    while(kk>2) { //now, if we have more than 2 sequential ones,
        quart_bit(1) ; //we'll count that as a positive quarter bit
        kk -= 4 ;
    } //that number's handled, next
    kk=1 ;
    //and reset counter
    while ( GETsample()==0 ) {
        //For sequential zeroes
        kk++ ;
        if (kk>8) {
            //too many zeroes
            if (BITSinBUFFER==BITSEXpected) { return ; } ;
            //means an error
        }
    } ;
    while(kk>2) {
        //more than two zeros
        quart_bit(0) ;
        //will be counted as a negative quarter bit
        kk -= 4 ;
    }
} ;

```

```

    return;
}

```

```

int GETsample() {
    int bit ;
    if (SAMPLEbytePTR>=IObytes) { return (1) ; } // append 1 at end of
buffer
    bit=0 ;
    if ( IObuffer[SAMPLEbytePTR] &SAMPLEbitMASK ) { bit=1 ;
} ; // get bit from buffer
    SAMPLEbitMASK=SAMPLEbitMASK>>1 ;
// and update BitMASK
    if (SAMPLEbitMASK==0) { SAMPLEbitMASK=0b10000000 ;
SAMPLEbytePTR++ ; } ;
    return(bit) ;
}

```

```

void quart_bit(int b){
    if (b) {
        numbits+=1;
    }
    else {
        switch (numbits) {
            case 3 : {
                RXbitBUFFER[BITSinBUFFER++] = prev_BIT;
                break;
            }
            case 5 : {
                if (prev_BIT){
                    RXbitBUFFER[BITSinBUFFER++] = 0;
                    RXbitBUFFER[BITSinBUFFER++] = 0;
                    prev_BIT = 0;
                }
                else {
                    RXbitBUFFER[BITSinBUFFER++] = 1;
                    prev_BIT = 1;
                }
                break;
            }
            case 7 : {
                if (prev_BIT){
                    RXbitBUFFER[BITSinBUFFER++] = 0;
                    RXbitBUFFER[BITSinBUFFER++] = 1;
                }
                else RXerror |= 1;
                break;
            }
            default : {
                RXerror |= 1;
                break;
            }
        }
    }
}

```

```

    return;
}

```

As the original bits were reconstructed, each was stored in an individual address in the array `RXbitBUFFER`. The function `GETbytes` was then called to reconstruct this data into bytes, and to remove error checking information. The result was stored in `RXbyteBUFFER`, and was then accessible to the higher level code.

```

void GETbytes (int n ) {           //takes what is in the RXbitBUFFER at the end of a
receive cycle and converts it into bytes... Needs to also check parity bits for >1 bytes

    int theBYTE ;
    int BITptr ;
    int k,j ;
    uint8_t b, parity;

    CRCinit ();

    BITptr=1 ;
    parity=1;
    for (j=0 ; j<n ; j++){
        theBYTE=0 ;
        for ( k=0 ; k<8 ; k++) { //reads through an individual byte of the data
stored in the rxbitbuffer
            b=RXbitBUFFER[BITptr++] ;
            if (b) parity=1-parity ;
            theBYTE+= b<<k ;
        } ;
        b=RXbitBUFFER[BITptr++] ;
        if (b) parity=1-parity ;
        RXbyteBUFFER[j]=theBYTE ;
        updateCRC (theBYTE) ;
    } ;
    return;
}

```

Much of this code was based on similar publically available code for an RFID reader which could function using the MIFARE standard. As most of the basic functionality of a reader and tag overlap, the code was adapted to perform the new

function of the tag. Unfortunately, this meant that there were several areas in which its performance was not ideal.

Data volume

In comparing amounts of data, it is necessary to differentiate between data being stored, and useable information. In order to encode and decode information using the above method, it was necessary to store it in several different buffers during the intervening processes. This data was not being stored in its most efficient manner, so the amount of data being stored for one byte of information might amount to several bytes of data. The goal of the coding was to try to reduce the amount of data storage required to be close to the amount of information being transmitted or received.

Transmitting data, it was necessary to store each transmitted bit as a byte of data in `IObuffer`. To transmit large amounts of information, it quickly became necessary to use a much larger amount of the PICC memory. Additionally, parity bits were added, so each byte of useable information transmitted required nine bytes of storage in `IObuffer`.

Receiving data, the problem was compounded. Information was initially stored by assembly code in `IObuffer` at a rate of one byte (eight samples) per bit of physical data received. The data was stored sequentially, and after the data which was previously transmitted. This also included some data overhead; parity bits making up one out of every nine bits received, and CRC check bits representing two bytes on the end of most transmissions. Once this data is converted into individual bits, it is still stored with one bit per byte in the `RXbitBUFFER`, to facilitate removal of the error checking overhead bytes. Only after it is finally processed is it stored in `RXbyteBUFFER` as bytes of information. This represented quite inefficient storage of information. For each received

transmission, discounting the overhead of the CRC bytes, the following equation represents the amount of memory used on the chip per byte of data received:

$$M = 29 \times B_{\text{information}} + B_{\text{information}} = 19(B_{\text{information}})$$

In this equation, M represents the memory required, while $B_{\text{information}}$ is the number of information bytes received by the PICC. Both are in units of bytes. The inefficiency here is evident, as 19 times the required storage is used in the process of receiving data.

Although the ATMega16 was capable of handling this inefficiency, it is hoped that the application will be applied to a smaller device without the need for excessive modification. As such, during the design process, every attempt was made to minimise the use of resources by the application, and to reduce the inefficiency of the transmission and receiving process.

Error Handling

One of the requirements of the assembly language code for receiving information was that the first address after the transmitted data in `IObuffer` should be the number of bytes the PICC is expecting in response. The device then sampled for the amount of time which would correspond to this amount of data, and the data was analysed. This made it difficult to deal with responses shorter than the expected one, and impossible to deal with longer ones, as all bits beyond the expected last bit were ignored. In correct operation of the device, the number of response bits for each transmission is known, so this solution could theoretically function. In any situation where the device does not receive exactly the response the programmer was expecting, however, there is a chance that parts of the response from the PCD could be disregarded and lost. This considerably decreases the robustness and adaptability of the design.

Although the use of C to decode the incoming data made it possible to detect errors in the received information, it was only after a complete block of information had been received that it was possible to detect these errors. By implementing error detection protocols in the assembly code, it would be possible to detect errors as they occurred,

Timing

The use of assembly solved the problems of timing during the physical processes of transmitting and receiving data, but it resulted in unexpected timing problems elsewhere, which only emerged as the circuit functionality was being simulated for the first time. Prior to simulation, it was difficult to predict what optimisation the compiling program would perform with the C code, making calculating the timing parameters impossible. As mentioned earlier, the time allotted for responses from the PICC were considered to be generous, however, the initial response of the PICC to a Request Acknowledge (REQA) command is required to occur exactly 1236 clock periods after the receipt of the last pause from the PCD. This is to ensure that if there are multiple PICCs in the field, they respond simultaneously and allow anticollision to take place.

Encoding information for transmission is an iterative process. For each bit that is to be transmitted, `TXconvert` calls `TXemit` once, and one additional time per byte to store the parity bit. The `TXemit` function requires 58 clock periods to process a '0', or 65 to process a '1'. When the overhead from the surrounding programming loop is included, this increases to a '0' requiring 160 clock periods to process, while a '1' requires 180 clock periods. The response to a REQA, is 16 bits in length, plus two parity bits, giving a time to process this command of between 2880 and 3240 clock

periods, discounting all other overheads. In either case, this is far above the permitted 1236 clock periods.

After completing reading data from the PCD, the process of converting the resultant samples back to information also took some time owing to the number of iterations required. Each sample required 54 clock periods to process, without considering conversion back into bits or bytes, meaning that for the decoding of a REQA command, which is seven bits long, the initial processing of samples would take 3456 clock periods. At this point in the development it was decided that these timing inefficiencies in the code could not be worked around, and the program must be substantially modified in order to function in the way laid out in the MIFARE standard.

Appendix B

DATA REDUCTION CODE

1. PC-based code

The following code performed the reduction of data in C, and was used to generate graphs from the temperature logs recorded by Speedfan software. The input of the software is a CSV file, and the program will create an output file if it is the first time the program is executed in a folder, or add to an existing output file if it finds one.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#define AMPBINS 19
#define MEANBINS 37
#define DURBINS 30

double therMarg;
int bins[AMPBINS][MEANBINS][DURBINS]={0};

typedef struct revNode
{
    int troughVal;
    int troughTime;
    int peakVal;
    int peakTime;
} revNode;
revNode reversals[100] = {0};

int storeOrNot(int timeIndex, int val, int *preIndex, int *preVal,
int *dir)
{
    if(*dir==1){ //if we were going upwards
        if(val>=*preVal){//then we're still going upwards (preval is
greatest measurement achieved)
            *preVal=val;
            *preIndex=timeIndex;
            return 0;
        }
        else{ //but if the new value is less than the previous
one
            if((*preVal-therMarg)>=(val+therMarg)){ //line forced to
change direction
                *dir=0;
                return 1;
            }
        }
    }
}
```

```

    }
    else //going downwards, but not enough to reverse
direction of line. preVal doesn't change
        return 0;
    }
}
else{
    if(val<=*preVal){//then we're still going downwards (preval is
smallest measurement achieved)
        *preVal=val;
        *preIndex=timeIndex;
        return 0;
    }
    else{ //but if the new value is less than the previous
one
        if((*preVal+therMarg)<=(val-therMarg)){ //line forced to
change direction
            *dir=1;
            return 1;
        }
        else //heading in the opposite direction, but not enough
to reverse direction of line. preVal doesn't change
            return 0;
        }
    }
}
}
/*
function:logCycles
Rounds values to the closest appropriate bin and increments the
appropriate bin value
Only works for mean values above 25 C and for amplitude differences of
above 4 C(this should be ensured by OOR design)
Amp Bins: 4,8,12,...76;(19)
Mean bins: 27,29,31,...99;(38)
Duration bins: 30 bins about the value 3, 10, 20, then
logarithmically with 8 values per decade from 30 to 53348(27 more
values)
*/
void logCycles(struct revNode slope, int n)
{
    double mean = ((slope.peakVal)+(slope.troughVal))/2;
    int ampl = (slope.peakVal)-(slope.troughVal);
    long int dur = (slope.peakTime)-(slope.troughTime);
    int amplBin, meanBin, durBin;

    if (dur < 0) dur = (0-dur);

    mean = (mean-26)/2;
    if (mean<0) mean = 0; //if the average falls less than the
minimum average of 27, round up to the minimum
    if (mean>36) mean = 36; //if it falls above the maximum of 101 C,
round down
    meanBin = (int)mean;

    ampl = (ampl - 2)/4;
    if (ampl<0) ampl = 0;
    if (ampl>18) ampl = 18;
    amplBin = (int)ampl;

    if (amplBin<=18)
        if (amplBin>meanBin) amplBin = meanBin;
}

```

```

else if (amplBin>(36-meanBin)) amplBin = (36-meanBin);

if (dur<5)
    durBin=0;
else if (dur<15)
    durBin=1;
else if (dur<25)
    durBin=2;
else if (dur<35)
    durBin=3;
else if (dur<46)
    durBin=4;
else if (dur<61)
    durBin=5;
else if (dur<82)
    durBin=6;
else if (dur<110)
    durBin=7;
else if (dur<146)
    durBin=8;
else if (dur<194)
    durBin=9;
else if (dur<260)
    durBin=10;
else if (dur<346)
    durBin=11;
else if (dur<462)
    durBin=12;
else if (dur<616)
    durBin=13;
else if (dur<821)
    durBin=14;
else if (dur<1096)
    durBin=15;
else if (dur<1461)
    durBin=16;
else if (dur<1948)
    durBin=17;
else if (dur<2597)
    durBin=18;
else if (dur<3464)
    durBin=19;
else if (dur<4620)
    durBin=20;
else if (dur<6161)
    durBin=21;
else if (dur<8215)
    durBin=22;
else if (dur<10955)
    durBin=23;
else if (dur<14609)
    durBin=24;
else if (dur<19481)
    durBin=25;
else if (dur<25978)
    durBin=26;
else if (dur<34643)
    durBin=27;
else if (dur<46199)
    durBin=28;
else

```

```

        durBin=29;

        bins[amplBin][meanBin][durBin] += n;

        return;
    }

/*
function:peakFlow
operation: Given a newly acquired PEAK(dir=1) or TROUGH(dir=0), works
backward through time history applying
rainflow counting rules, saving cycles and half cycles back to either
(a) start of time history or
(b) it encounters a peak of greater size
Assumptions: *oppSide points to the first slope in the linked list,
which is the most recent complete half-cycle
*sameSide points to the second last slope, which is also head->next
*/
void peakFlow(int troughVal, int troughTime, int peakVal, int
peakTime, int *maxInd)
{
    if (*maxInd<0)           //this is the first value received
    {
        reversals[0].troughVal = troughVal;
        reversals[0].peakVal= peakVal;
        reversals[0].troughTime = troughTime;
        reversals[0].peakTime = peakTime;
        *maxInd = 0;
        return;
    }

    if (peakTime>troughTime)
    {
        dir = 1;
    }
    else
    {
        dir = 0;
    }

    while((((dir==1)&&(reversals[*maxInd].peakVal)<=peakVal)||((dir==0)&&(
(reversals[*maxInd].troughVal)>=troughVal)))&&(*maxInd>1))
        {
            //as long as we haven't encountered a greater peak or slope or
            reached the start of the time history

            logCycles(reversals[*maxInd],2); //record the smaller cycle
            if(dir== 1)
            {
                troughVal = reversals[*maxInd-1].troughVal;//update the
size of current slope
                troughTime = reversals[*maxInd-1].troughTime;
            }
            else
            {
                peakVal = reversals[*maxInd-1].peakVal; //update the size
of current slope
                peakTime = reversals[*maxInd-1].peakTime;
            }
            reversals[*maxInd].peakTime = 0;

```

```

        reversals[*maxInd].peakVal = 0;
        reversals[*maxInd].troughTime = 0;
        reversals[*maxInd].troughVal = 0;
        reversals[*maxInd-1].peakTime = 0;
        reversals[*maxInd-1].peakVal = 0;
        reversals[*maxInd-1].troughTime = 0;
        reversals[*maxInd-1].troughVal = 0;
        *maxInd-=2;
    }
    /*
    Three possible reasons for terminating
    1: Encountered a greater peak/deeper trough than the one in
    consideration
    2: oppSide->next is NULL, so reached the start of the time history
    3: sameSide->next is NULL, so approaching the start of time
    history
    */

    if(*maxInd==0) //reached start of time history at
    the oppSide, meaning rule 4 (>/<)
    { //applies: if current
    range > oppSide record oppSide as half cycle

    if(((dir==1)&&(peakVal>=reversals[0].peakVal))||((dir==0)&&(troughVal<
    =reversals[0].troughVal)))
        {
            logCycles(reversals[0], 1); //log half Cycle
            *maxInd-=1;
        }
    }

    *maxInd+=1;
    reversals[*maxInd].peakTime = peakTime;
    reversals[*maxInd].peakVal = peakVal;
    reversals[*maxInd].troughTime = troughTime;
    reversals[*maxInd].troughVal = troughVal;

    return;
}

void cleanup(int *maxInd){

    while (*maxInd>=0)
    {
        logCycles(reversals[*maxInd--], 1);
    }
    return;
}

int main(int nNumberOfArgs, char* pszArgs[])
{
    int preTime=0, lastTime, timeVal, therVal0, therVal1, preTher0=0,
    preTher1=0, lastTher0, lastTher1, i, j, k, maxInd=-1;
    long int interim;
    char str[80] = {0}, buf[20] = {0};
    char *pch;
    char *ampvals[] =
    {"4", "8", "12", "16", "20", "24", "28", "32", "36", "40", "44", "48", "52", "56", "
    60", "64", "68", "72", "76"};
    char *durvals[] =
    {"5", "15", "25", "35", "46", "61", "82", "110", "146", "194", "260", "346", "462"

```

```

,"616","821","1096","1461","1948","2597","3464","4620","6161","8215","
10955","14609","19481","25978","34643","46199","61605"};
    char *meanvals[] =
{"27","29","31","33","35","37","39","41","43","45","47","49","51","53"
,"55","57","59","61","63","65","67","69","71","73","75","77","79","81"
,"83","85","87","89","91","93","95","97","99"};
    int dir = -1; //1:upwards, 0:downwards, -1 first case;
    int first = 1, slope = 0;
    FILE *inputFile = fopen(pszArgs[1],"r");
    FILE *output1 =fopen("C:\\Rainflow\\OORout1.csv","w"), *output2,
*output3, *output4;
    char buffer[256]={0x0};

    therMarg = -1;

    therMarg=2;

    output3 = fopen("C:\\Rainflow\\rawdata.csv", "r");

    if (output3 != NULL) //if we already have data to read from, load
it into the memory
    {
        for(i=0;i<AMPBINS;i++)
        {
            for(j=0;j<MEANBINS;j++)
            {
                k=0;
                fgets(buffer,sizeof(buffer),output3);
                pch = strtok(buffer,"");
                while ((pch != NULL) && (k<DURBINS))
                {
                    bins[i][j][k++] = atoi(pch);
                    pch = strtok (NULL, "");
                }
            }
        }
        fclose(output3);
    }

    if ( inputFile != NULL )
    {
        while(fgets(buffer,sizeof(buffer),inputFile)!=0) /* read a
record */
        {
            pch = strtok(buffer, ","); //Parse string to tokens
            timeVal = atoi(pch); //Convert string to int
            pch = strtok(NULL, ",");
            therVal0 = atoi(pch);
            pch = strtok(NULL, ",");
            therVal1 = atoi(pch);
            if(first!=0) //if this is the first
value read
            {
                preTher0 = therVal0;
                preTher1 = therVal1;
                preTime = timeVal;
                lastTher0 = therVal0;
                lastTher1 = therVal1;
                lastTime = timeVal;
                first = 0;
            }
        }
    }

```

```

    }
    else if (dir == -1)
    {
        if (preTher0 < therVal0)
        {
            dir = 1;
            lastTher0 = preTher0;
            preTher0 = therVal0;
        }
        else if (preTher0 > therVal0)
        {
            dir = 0;
            lastTher0 = preTher0;
            preTher0 = therVal0;
        }
    }
    else
    {
        if(storeOrNot(timeVal, therVal0, &preTime, &preTher0,
&dir))
        {
            itoa(preTime, buf, 10);
            fputs(buf, output1);
            fputs(",", output1);
            itoa(preTher0, buf, 10);
            fputs(buf, output1);
            fputs("\n", output1);
            if (preTher0 > lastTher0) //an
upward slope
            {
                peakFlow(lastTher0, lastTime, preTher0,
preTime, &maxInd, 1);
            }
            else //a downward slope
            {
                peakFlow(preTher0, preTime, lastTher0,
lastTime, &maxInd, 0);
            }
            lastTher0 = preTher0;
            lastTher1 = preTher1;
            lastTime = preTime;
            preTher0 = therVal0;
            preTher1 = therVal1;
            preTime = timeVal;
        }
    }
}
fclose(inputFile);
cleanup(&maxInd);
fclose(output1);
output2 = fopen("C:\\Rainflow\\Graphs.csv", "w");
fputs("Amplitude vs Mean\n", output2);
fputs(" ,", output2);
for(i=0;i<MEANBINS;i++)
{
    fputs(meanvals[i], output2);
    fputs(",", output2);
}
fputs("\n", output2);
for(i=0;i<AMPBINS;i++)
{
    fputs(ampvals[i], output2);

```

```

fputs(", ", output2);
for(j=0; j<MEANBINS; j++)
{
    interim=0;
    for(k=0; k<DURBINS; k++)
    {
        interim+=bins[i][j][k];
    }
    itoa(interim, buf, 10);
    fputs(buf, output2);
    fputs(", ", output2);
}
fputs("\n", output2);
fputs("\n", output2);

fputs("Duration vs Mean\n", output2);
fputs(" ", output2);
for(i=0; i<MEANBINS; i++)
{
    fputs(meanvals[i], output2);
    fputs(", ", output2);
}
fputs("\n", output2);
for(i=0; i<DURBINS; i++)
{
    fputs(durvals[i], output2);
    fputs(" ", output2);
    for(j=0; j<MEANBINS; j++)
    {
        interim=0;
        for(k=0; k<AMPBINS; k++)
        {
            interim+=bins[k][j][i];
        }
        itoa(interim, buf, 10);
        fputs(buf, output2);
        fputs(", ", output2);
    }
    fputs("\n", output2);
}

fputs("\n", output2);
fputs("Amplitude vs Duration\n", output2);
fputs(" ", output2);
for(i=0; i<AMPBINS; i++)
{
    fputs(ampvals[i], output2);
    fputs(", ", output2);
}
fputs("\n", output2);
for(i=0; i<DURBINS; i++)
{
    fputs(durvals[i], output2);
    fputs(" ", output2);
    for(j=0; j<AMPBINS; j++)
    {
        interim=0;
        for(k=0; k<MEANBINS; k++)
        {
            interim+=bins[j][k][i];

```

```

    }
    itoa(interim, buf, 10);
    fputs(buf, output2);
    fputs(",", output2);
    }
    fputs("\n", output2);
}
fclose(output2);
output3 = fopen("C:\\Rainflow\\rawdata.csv", "w");
for(i=0;i<AMPBINS;i++)
{
    for(j=0;j<MEANBINS;j++)
    {
        for(k=0;k<DURBINS;k++)
        {
            itoa(bins[i][j][k], buf, 10);
            fputs(buf, output3);
            fputs(",", output3);
        }
        fputs("\n", output3);
    }
}
fclose(output3);
}
else /* fopen() returned NULL */
{
    printf("File open operation failed!!\n");
    return 1;
}
return 0;
}
}

```

2. ATMega16 Code

The following code performs the same function; however it is based on the ATMega16 so the source of the thermal data is the USB connection on the device. The timer is used to calculate duration of the slopes by adding the duration between the receipts of successive temperature readings.

```

/*
 * GccApplication2.c
 *
 * Created: 04/07/2012 13:22:55
 * Author: Eanna
 */

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
#include <util/delay.h>
#include <stdbool.h>
#include <stdlib.h>

```

```

#include "usbdrv.h"

#define F_CPU 12000000L

#define USB_LED_OFF 0
#define USB_LED_ON 1
#define NEW_TEMP 2
#define STANDBY 3
#define THERMARG 2

static uchar incomBuf[3];

typedef struct revNode
{
    uint8_t peakVal;
    uint8_t troughVal;
    uint16_t dur;
}revNode;

uint16_t slopeDur;
uint8_t lastVal=0, preVal=0;
uint8_t peakVal, troughVal;
uint8_t isFirst=1, dir=2, maxInd=101;
revNode reversals[100] = {0};

ISR(TIMER1_COMPA_vect)
{
    slopeDur+=1; // increment the slope duration by 1 second
    PORTC = slopeDur;
}

void logCycles(struct revNode slope, int n)
{
    uint8_t mean = (slope.peakVal)-(slope.troughVal)/2;
    uint8_t ampl = (slope.peakVal)-(slope.troughVal);
    uint8_t currBin, init;
    uint16_t dur = slope.dur;

    int durBin;

    if (mean<26)
        mean = 0;
    else if (mean > 98)
        mean = 36;
    else
        mean = (uint8_t)(mean-26)/2;

    if (ampl < 2)
        ampl = 0;
    else
        ampl = (ampl - 2)/4;

    if (mean <= 18) //for mean bin values less than 18, the
maximum amplitude bin is equal to the mean bin
    {
        if (ampl > mean)
            ampl = mean;
    }
    else //for mean bin values greater than 18, maximum
amplitude bin is 36-(mean bin)

```

```

{
    if (ampl > (36-mean))
        ampl = (36-mean);
}

mean = mean-ampl;

init = (90*((38*ampl)-(ampl*ampl)));
init+= (90*mean);

if (dur<5)
durBin=0;
else if (dur<15)
durBin=3;
else if (dur<25)
durBin=6;
else if (dur<35)
durBin=9;
else if (dur<46)
durBin=12;
else if (dur<62)
durBin=15;
else if (dur<82)
durBin=18;
else if (dur<110)
durBin=21;
else if (dur<146)
durBin=24;
else if (dur<194)
durBin=27;
else if (dur<260)
durBin=30;
else if (dur<346)
durBin=33;
else if (dur<462)
durBin=36;
else if (dur<616)
durBin=39;
else if (dur<821)
durBin=42;
else if (dur<1096)
durBin=45;
else if (dur<1461)
durBin=48;
else if (dur<1948)
durBin=51;
else if (dur<2597)
durBin=54;
else if (dur<3464)
durBin=57;
else if (dur<4620)
durBin=60;
else if (dur<6161)
durBin=63;
else if (dur<8215)
durBin=66;
else if (dur<10955)
durBin=69;
else if (dur<14609)
durBin=72;
else if (dur<19481)

```

```

durBin=75;
else if (dur<25978)
durBin=78;
else if (dur<34643)
durBin=81;
else if (dur<46199)
durBin=84;
else
durBin=87;

init = init+durBin;

// PORTA = LOW8 Address
// PORTB = HI8 Address
// PORTC = DATA I/O
// PORTD PIN0 = /OE
// PORTD PIN1 = /WE
// PORTD PIN3 = /CE

PORTD |= 0x0B;
DDRD |= 0x0B; //set pins 0,1 and 3 of
PORTD as outputs
PORTD &= 0xF7; //enable memory device
using /CE
DDRA = 0xFF; //PORTA output mode
DDRB = 0xFF; //PORTB output mode
PORTA = init && 0x00FF; //set initial address
PORTB = (init >> 8) && 0x00FF;
PORTD &= 0xFE; //set memory as output
currBin = PINC; //read current memory bin
PORTD |= 0x01; //Disable memory output
using /OE
DDRC = 0xFF; //switch PORTC to output
mode
if (currBin == 0xFF); //if the current bin is
full, overflow
{
PORTC = 0x00; //reset the count at that
address
PORTD &= 0xFD; //Enable writing using /WE
PORTD |= 0x02; //disable writing
PORTA = (init+1) && 0x00FF;
PORTB = ((init+1) >> 8) && 0x00FF;
DDRC = 0x00; //set portc as input
PORTD &= 0xFE; //set memory as output
currBin = PINC;
PORTD |= 0x01; //Disable output using /OE
DDRC = 0xFF; //switch PORTC to output
mode
if (currBin == 0xFF);
{
PORTC = 0x00; //reset the count at that
address
PORTD &= 0xFD; //Enable writing using /WE
PORTD |= 0x02; //disable writing
PORTA = (init+2) && 0x00FF;
PORTB = ((init+2) >> 8) && 0x00FF;
DDRC = 0x00; //switch PORTC to input
mode
PORTD &= 0xFE; //set memory as output
currBin = PINC;

```

```

        PORTD |= 0x01;           //Disable memory output
using /OE
        DDRC = 0xFF;           //switch PORTC to output
mode
    }
}
PORTC = currBin + 1;
PORTD &= 0xFD;                 //Enable memory chip writing
using /WE
    _delay_us(1);
PORTD |= 0x0B;                 //Disable memory chip
DDR0 |= 0x0B;                 //turn on pull-ups

    return;
}

void peakFlow()
{
    uint16_t dur = slopeDur;
    uint8_t slopeDir;

    slopeDur = 0;

    if (maxInd>100)             //this is the first value
received
    {
        reversals[0].troughVal = troughVal;
        reversals[0].peakVal= peakVal;
        reversals[0].dur = dur;
        maxInd = 0;
        return;
    }

    if (preVal>lastVal)
        slopeDir=1;
    else
        slopeDir=0;

    while((((slopeDir ==
1)&&(reversals[maxInd].peakVal)<=preVal)||((slopeDir ==
0)&&((reversals[maxInd].troughVal)>=preVal)))&&(maxInd)>1))
    { //as long as we haven't encountered a greater peak or slope or
reached the start of the time history
        logCycles(reversals[maxInd],2);           //record the
smaller cycle
        if(slopeDir== 1)                         //if we on an upward
slope
        {
            troughVal = reversals[maxInd-1].troughVal;//update the
size of current slope
            dur = (reversals[maxInd-1].dur)+(reversals[maxInd].dur) +
dur;
        }
        else                                     //if we are on a downwards
slope
        {
            peakVal = reversals[maxInd-1].peakVal; //update the size
of current slope
            dur = (reversals[maxInd-1].dur)+(reversals[maxInd].dur) +
dur;
        }
    }
}

```

```

    reversals[maxInd].dur = 0;
    reversals[maxInd].peakVal = 0;
    reversals[maxInd].troughVal = 0;
    reversals[maxInd-1].dur = 0;
    reversals[maxInd-1].peakVal = 0;
    reversals[maxInd-1].troughVal = 0;
    maxInd -= 2;
}

// Three possible reasons for terminating
// 1: Encountered a greater peak/deeper trough than the one in
consideration
// 2: oppSide->next is NULL, so reached the start of the time
history
// 3: sameSide->next is NULL, so approaching the start of time
history

    if(maxInd==0) //reached start of time history at
the oppSide, meaning rule 4 (>/<)
    { //applies: if current
range > oppSide record oppSide as half cycle

if(((slopeDir==1)&&(preVal>=reversals[0].peakVal))||((slopeDir==0)&&(preVal<=reversals[0].troughVal)))
    {
        logCycles(reversals[0], 1); //log half Cycle
        reversals[maxInd].dur = dur;
        reversals[maxInd].peakVal = peakVal;
        reversals[maxInd].troughVal = troughVal;
        return;
    }
}

maxInd+=1;
reversals[maxInd].dur = dur;
reversals[maxInd].peakVal = peakVal;
reversals[maxInd].troughVal = troughVal;

return;
}

int storeOrNot(uint8_t val)
{
    if (isFirst == 1)
    {
        isFirst = 0;
        TIMSK |= (1 << OCIE1A); //Enable CTC interrupt
        sei(); //Enable global
interrupts
        OCR1A = 11719; //Set CTC compare
value to ~ 1Hz(+0.002%) at 12MHz AVR clock, with a prescaler of 1024
        TCCR1B |= ((1 << CS10) | (1 << CS12)); //Start timer at
Fcpu/64
        lastVal = val;
        preVal = val;
        return 0;
    }

    else if (dir == 2)
    {
        if (val>preVal)

```

```

        dir = 1;
    else if (val<preVal)
        dir = 0;
    return 0;
}

if(dir == 1)
{ //if we were going upwards
    if(val>=preVal) //then we're still going
upwards (preval is greatest measurement achieved)
    {
        preVal=val;
        return 0;
    }
    else //but if the new value
is less than the previous one
    {
        if((preVal-THERMARG)>=(val+THERMARG))
        { //line forced to change direction
            dir = 0;
            return 1;
        }
        else
            return 0;
    }
}
else
{
    if(val<=preVal)
    { //then we're still going downwards (preVal is least
measurement achieved)
        preVal=val;
        return 0;
    }
    else
    { //but if the new value is less than the previous one
        if((preVal+THERMARG)<=(val-THERMARG))
        { //line forced to change direction
            dir = 1;
            return 1;
        }
        else //heading in the opposite direction, but not enough
to trigger
            return 0;
    }
}
}

void cleanup(){
    while (maxInd>=0)
    {
        logCycles(reversals[maxInd], 1);
        maxInd--;
    }
}

USB_PUBLIC uchar usbFunctionSetup(uchar data[8])
{
    usbRequest_t *rq = (void *)data; // cast data to correct type

    switch(rq->bRequest)

```

```

{ // custom command is in the bRequest field
  case USB_LED_ON:
    PORTB |= 1; // turn LED on
    PORTA |= 1;
    return 0;
  case USB_LED_OFF:
    PORTB &= ~1; // turn LED off
    PORTA &= ~1;
    return 0;
  case NEW_TEMP:
    if(storeOrNot(rq->wValue.bytes[0]))
    {
      peakFlow();
      lastVal = preVal;
      preVal = rq->wValue.bytes[0];
    }
    return 0;
// case STANDBY:
//   peakFlow(lastVal);
//   cleanup();
//   break;
  default:
    return 0;
}
return 0;
}

int main(void)
{
  uchar i;
  wdt_enable(WDTO_1S); // enable 1s watchdog timer
  TCCR1B |= (1 << WGM12); // Configure timer 1 for CTC mode
  PORTC=0xFF;
  DDRB|=1;
  PORTB &= 0xFE;
  DDRA|=1;
  PORTA &= 0xFE;
  isFirst = true;
  dir = true;
  PORTA |= 1;

  usbInit();

  PORTD |= 0x0B;
  DDRD &= 0xF4; //set oe,we and ce pull ups active

  usbDeviceDisconnect(); // enforce re-enumeration
  for(i = 0; i<250; i++)
  { // wait 500 ms
    wdt_reset(); // keep the watchdog happy
    _delay_ms(2);
  }
  usbDeviceConnect();

  sei(); // Enable interrupts after re-enumeration

  while(1)
  {
    wdt_reset(); // keep the watchdog happy
    usbPoll();
  }
}

```

```
return 0;  
}
```


Appendix C

MODIFICATIONS TO READER HARDWARE CODE

The code governing the hardware operation of the RFID reader had to be modified in order to enable identification of the prototype tag, and to allow the relaying of large amounts of data. The program was initially modified to enable the identification of a tag containing LCM information through interpretation of the ATQA and SAK signals transmitted. This was achieved by adding the unique identifying values to a header file.

```
#define ATQA_Eannas_Tag      0x0100
#define SAK_Eannas_Tag      0x30
```

The highlighted line in the following section was then added to the `MIFARE_TypeIdentification` function, to allow proper identification of the PICC when the function is called.

```
enum MifareICType MIFARE_TypeIdentification(unsigned char
    *baATQ, unsigned char bSAK)
{
    unsigned int iATQA;

    iATQA = baATQ[0] *256 + baATQ[1];

    if((iATQA == ATQA_Philips_Mifare_UL) && (bSAK ==
SAK_Philips_Mifare_UL))
        return Mifare_UL;
    else if((iATQA == ATQA_Philips_Mifare_1K) && (bSAK ==
SAK_Philips_Mifare_1K))
        return Mifare_1K;
    else if((iATQA == ATQA_Philips_Mifare_4K) && (bSAK ==
SAK_Philips_Mifare_4K))
        return Mifare_4K;
    else if((iATQA == ATQA_Philips_DesFire) && (bSAK ==
SAK_Philips_DesFire))
        return Mifare_DesFire;
    else if((iATQA == ATQA_Eannas_Tag) && (bSAK ==
SAK_Eannas_Tag))
        return Eannas_Tag;
    else
        return Unknown;
}
```

In terminal mode, on identifying a PICC with LCM data, the reader immediately requests sequential relaying of LCM data and relays it automatically to the middleware, shown below.

```

if (eMifareType == Eannas_Tag)
{
    // Read and send via RS232/USB Eannas Tag Data Block 4..15
    status = ERead(abDataBuffer);
    printf("\r\n");
    for(i=0; i<=15; i++){printf("%b02X ", abDataBuffer[i]);};
    printf("\r\n");
    while((abDataBuffer[0]==0x12) && (status==STATUS_SUCCESS))
    {
        status = ERead1(abDataBuffer);
        for(i=0; i<=15; i++){printf("%b02X ",
            abDataBuffer[i]);};
        printf("\r\n");
    }
}

```

The `ERead` function referenced in the above code requests the first block of data from the PICC, and `ERead1` requests all subsequent blocks, both through transmission of the unique codes defined for these functions. The `printf` function relays the information retrieved via the USB interface to the terminal, where it is printed on the screen. The functions `ERead` and `ERead1` are outlined below.

```

short ERead(unsigned char *_data)
{
    short status = STATUS_SUCCESS;

    char tmp = 0;
    ResetInfo(MInfo);
    SerBuffer[0] = 0xCA;
    MInfo.nBytesToSend = 1;
    SetTimeout(10000);
    status = M522PcdCmd(RC522_CMD_TRANSCEIVE,
        SerBuffer,
        &MInfo);
    if (status != STATUS_SUCCESS)
    {
        if (status != STATUS_IO_TIMEOUT ) // no timeout occurred

```

```

    {
        if (MInfo.nBitsReceived == 4)
        {
            SerBuffer[0] &= 0x0f;
            if ((SerBuffer[0] & 0x0a) == 0)
            {
                status = STATUS_AUTHENT_ERROR;
            }
            else
            {
                status = STATUS_INVALID_FORMAT;
            }
        }
    }
    memset(_data,0,16);
}
else // Response Processing
    memcpy(_data,SerBuffer,16);

return status;
}

```

```

short ERead1(unsigned char *_data)
{
    short status = STATUS_SUCCESS;
    char tmp = 0;

    ResetInfo(MInfo);
    SerBuffer[0]=0x13;
    MInfo.nBytesToSend = 1;
    SetTimeout(10000);
    status = M522PcdCmd(RC522_CMD_TRANSCEIVE,
                      SerBuffer,
                      &MInfo);

    if (status != STATUS_SUCCESS)
    {
        if (status != STATUS_IO_TIMEOUT) // no timeout occurred
        {
            if (MInfo.nBitsReceived == 4)
            {
                SerBuffer[0] &= 0x0f;
                if ((SerBuffer[0] & 0x0a) == 0)
                {
                    status = STATUS_AUTHENT_ERROR;
                }
                else
                {
                    status = STATUS_INVALID_FORMAT;
                }
            }
        }
    }
    memset(_data,0,16);
}

```

```

}
else // Response Processing
    memcpy(_data, SerBuffer, 16);
return status;
}

```

Both of these perform essentially the same purpose, using the `M522PcdCmd` function to transmit the command, `CAh` in the case of `ERead`, and `13h` in the case of `ERead1`.

`EReadAll` is a function which performs the functions of both `ERead` and `ERead1` until the PICC indicates that the end of the data has been reached by changing the command byte transmitted.

```

short EReadAll()
{
    short status = STATUS_SUCCESS;
    unsigned char status1;
    unsigned char i;
    char tmp = 0;
    ResetInfo(MInfo);
    SerBuffer[0] = 0xCA;
    MInfo.nBytesToSend = 1;
    SetTimeout(10000);
    status = M522PcdCmd(RC522_CMD_TRANSCEIVE,
                      SerBuffer,
                      &MInfo);
    if (status == STATUS_SUCCESS)
    {
        status1 = 0;
        putchar(COM_SOH); // Send SOF .. Start of Frame
        putchar(status1); // Send STATUS
        putchar(16); // Send LEN
        for (i=0; i<16; i++)
        {
            putchar(SerBuffer[i]); // Send DATA
        }
    }
    else
    {
        if (status != STATUS_IO_TIMEOUT) // no timeout occurred
        {
            if (MInfo.nBitsReceived == 4)
            {
                SerBuffer[0] &= 0x0f;
                if ((SerBuffer[0] & 0x0a) == 0)

```

```

        {
            status = STATUS_AUTHENT_ERROR;
        }
        else
        {
            status = STATUS_INVALID_FORMAT;
        }
    }
}

while(status==STATUS_SUCCESS&&SerBuffer[0]==0x12){
    ResetInfo(MInfo);
    SerBuffer[0]=0x13;
    MInfo.nBytesToSend = 1;
    status = M522PcdCmd(RC522_CMD_TRANSCEIVE,
                      SerBuffer,
                      &MInfo);

    if (status == STATUS_SUCCESS){
        status1 = 0;
        putchar(COM_SOH);           // Send SOF .. Start of Frame
        putchar(status1);          // Send STATUS
        putchar(16);                //Send LEN
        for(i=0;i<16;i++){
            {
                putchar(SerBuffer[i]); //Send DATA
            }
        }
    }
    else{
        if (status != STATUS_IO_TIMEOUT ){ // no timeout occurred
            if (MInfo.nBitsReceived == 4){
                SerBuffer[0] &= 0x0f;
                if ((SerBuffer[0] & 0x0a) == 0){
                    status = STATUS_AUTHENT_ERROR;
                }
                else{
                    status = STATUS_INVALID_FORMAT;
                }
            }
        }
    }
}
return status;
}

```

This was written separately to the other functions, rather than calling the ERead1 function repeatedly, as it offered a slight improvement in computation time, reducing the delay between transmissions of data.

In UltraLight mode, the reader simply waits for a command through the USB interface, so the addition of the following code to the UltraLight mode enabled the use of the above functions in this mode by identifying the codes for reading of large volumes of data from the PICC.

```

void MIFARE_PC_Reader(void)
{
    short status;
    unsigned char idata abResponseBuffer[32], abSNR[8], bSAK;
    unsigned int addr;
    //printf_lcd(1,1," PC Mode  ");
    //printf_lcd(2,1," RcvCmd:0x%b02X  ",gbaSerialRecBuffer[1]);
    switch(gbaSerialRecBuffer[1])
    {
        //-----
        // ISO14443 Activation & Mifare (Ultra Light) Commands
        //-----
        case 0x20: // Reader Command: RF - Reset
            Rc522RFReset(5); //RF - Reset
            SendCmdResponse(0, abResponseBuffer, 0);
            break;

        case 0xCA: // Reader Command: ERead
            status = EReadAll();
            break;

        case 0x12: // Reader Command: ERead1
            status = ERead1(&abResponseBuffer);

            SendCmdResponse(status, abResponseBuffer,
                16);
            break;

        case 0x26: // Reader Command: REQA
            status = Request(ISO14443_3_REQA,
                &abResponseBuffer);

            SendCmdResponse(status, abResponseBuffer,
                2);
            break;
    }
    ...
}

```

Through inclusion of these commands in the program for MIFARE Magic, the transmission of data to the PC interface was enabled.

Appendix D

MODIFICATION OF MIFARE MAGIC VISUAL BASIC SOFTWARE

Several modifications to the software on the middleware PC were required to enable the operation of the advanced features of the PICC. While the Terminal mode interacts directly with the hardware of the PICC, the UltraLight mode offers more functionality, but requires some modification to allow addition of the features required. An overview of these amendments is presented in this appendix.

As previously, it was necessary for the software to identify the prototype PICC, and this was done by means of the ATQA and SAK codes relayed from the PCD when a REQA command is performed. The relevant codes were added as shown below.

```
Public Const ATQA_Philips_Mifare_UL = &H4400' ATQA Byte 0 | ATQA Byte 1
Public Const ATQA_Philips_Mifare_1K = &H400
Public Const ATQA_Philips_Mifare_4K = &H200
Public Const ATQA_Philips_DesFire = &H4403
Public Const ATQA_Eannas_Tag = &H100

Public Const SAK_Philips_Mifare_UL = &H0
Public Const SAK_Philips_Mifare_1K = &H8
Public Const SAK_Philips_Mifare_4K = &H18
Public Const SAK_Philips_DesFire = &H20
Public Const SAK_Eannas_Tag = &H30
```

The type `Eannas_Tag` was also added to the enumeration of the types of tag, as shown below.

```
Public Enum MifareICType
    Unknown
    Mifare_UL
    Mifare_1K
    Mifare_4K
    Mifare_DesFire
    Eannas_Tag
End Enum
```

Finally, the enumeration was applied in the identification code below, to indicate that the tag identified had access to PC LCM data.

```
Public Function Get_MIFARE_TypeIdentification_String(ATQA() As
Byte, SAK As Byte) As String
    Dim MifareType As MifareICType

    MifareType = MIFARE_TypeIdentification(ATQA(), SAK)

    Select Case MifareType
        Case Unknown:
            Get_MIFARE_TypeIdentification_String = "Unknown IC
Type"
        Case Mifare_UL:
            Get_MIFARE_TypeIdentification_String = "Philips
Mifare Ultra Light"
        Case Mifare_1K:
            Get_MIFARE_TypeIdentification_String = "Philips
Mifare 1K"
        Case Mifare_4K:
            Get_MIFARE_TypeIdentification_String = "Philips
Mifare 4K"
        Case Mifare_DesFire:
            Get_MIFARE_TypeIdentification_String = "Philips
Mifare DesFire"
        Case Eannas_Tag:
            Get_MIFARE_TypeIdentification_String = "PC LCM
Info"
    End Select
End Function
```

The additional commands were also required to be implemented in the interface, allowing the relaying of information to the program, either to be printed into the text box, or to be saved to a file. The first of these commands, ERead, simply sends the initial request for information from the PICC, by sending the CA_h signal. The code applying to the ERead and ERead1 buttons are below.

```
Private Sub Command_ERead_Click()
    MifareERead (MIFARE_EREAD)
End Sub
Private Sub Command_ERead1_Click()
    MifareERead (MIFARE_EREAD1)
End Sub
```

```

Private Sub MifareERead(Page As Byte)
    Dim lngStatus As Long
    Dim bytRecBuffer(30) As Byte
    Dim strOutText As String
    Dim bytLoopCounter

    lngStatus = ISO14443_Mifare_ERead(Page, bytRecBuffer())

    If lngStatus <> 0 Then
        Error2Log (lngStatus)
    Else
        strOutText = ByteArray2HEXString(bytRecBuffer, 16, True)
+ " " + ByteArray2ASCIIString(bytRecBuffer, 16, False)
        Text_Log.SelText = " <- Data: " + strOutText + vbCrLf
    End If
End Sub
    
```

ERead1 performs in exactly the same manner as ERead, simply passing a different command to the MifareERead function. Both of these commands are defined as follows.

```

Public Const MIFARE_EREAD = &HCA
Public Const MIFARE_EREAD1 = &H12
    
```

The above function calls the ISO14443_Mifare_ERead function, which is described below. This function simply uses the inbuilt command functions of the software to send the relevant command to the PCD.

```

Public Function ISO14443_Mifare_ERead(Block As Byte, ByRef
ReadData() As Byte) As Long
    Dim bytRcvBuffer(20) As Byte
    Dim bytSndBuffer(16) As Byte
    Dim bytRecStatus As Byte
    Dim lngStatus As Long
    Dim lngRecLen As Long

    lngRecLen = 0
    lngStatus = PCReaderCmdExchange(Block, bytSndBuffer, 0,
ReadData(), 16, lngRecLen)
    ISO14443_Mifare_ERead = lngStatus 'Return Error Code
End Function
    
```

The third button added to the interface is the ERead All command, which performs the initial request for data, and sends command bytes requesting the next

frame as long as the PICC indicates that further data is available. The data returned in this way is printed to the screen and the ASCII representation of the information is also displayed.

```

Private Sub MifareEReadAll ()
    Dim lngStatus As Long
    Dim bytRecBuffer(30) As Byte
    Dim strOutText As String
    Dim bytLoopCounter

    lngStatus = ISO14443_Mifare_ERead(MIFARE_EREAD,
    bytRecBuffer())
    If lngStatus <> 0 Then
        Error2Log (lngStatus)
    Else
        Text_Log.SelText = "Transmitting..." + vbCrLf
        strOutText = ByteArray2HEXString(bytRecBuffer, 16, True)
+ " " + ByteArray2ASCIIString (bytRecBuffer, 16, False)
        Text_Log.SelText = " <- Data: " + strOutText + vbCrLf
    End If

    Do Until ((lngStatus <> 0) Or (bytRecBuffer(0) <> &H12))
        lngStatus = ISO14443_Mifare_ERead(MIFARE_EREAD1,
        bytRecBuffer())
        If lngStatus = 0 Then
            strOutText = ByteArray2HEXString(bytRecBuffer, 16,
True) + " " + ByteArray2ASCIIString (bytRecBuffer, 16, False)
            Text_Log.SelText = " <- Data: " + strOutText + vbCrLf
        Else
            Error2Log (lngStatus)
        End If
    Loop

    If bytRecBuffer(0) = &H16 Then
        Text_Log.SelText = "End of Data" + vbCrLf
    End If
End Sub

```

Although this functioned as expected, the data was not in any structure, simply returned as bytes. Also, the processes which convert the data to ASCII representation and print the results to the screen required significant processing time. As such, the function Read and Save was added, which takes the returned data in the same way as the above, but saves it to a file specified by the user through a save dialog. The functionality of the button is implemented as follows.

```

Private Sub ReadSave_Click ()
    CommonDialog1.DialogTitle = "Save Output as..."
    CommonDialog1.InitDir = "%USERPROFILE%\My Documents\"

```

```

        CommonDialog1.Filter = "Comma-Separated Values
(*.CSV)|*.CSV"
        CommonDialog1.ShowSave
        If Len(CommonDialog1.FileName) <> 0 Then
            Text_Log.SelText = "Saving to " +
CommonDialog1.FileName + vbCrLf
            MifareEReadAllSave (CommonDialog1.FileName)
        Else
            End If
    End Sub

```

```

Private Sub MifareEReadAllSave(filename As String)
    Dim lngStatus As Long
    Dim bytRecBuffer(30) As Byte
    Dim strOutText As String
    Dim counter1 As Integer
    Dim current As Long
    Dim addr As Integer
    Dim result As String

    counter1 = 1
    current = 0
    addr = 1
    result = ""

    Open filename For Output As #1

    lngStatus = ISO14443_Mifare_ERead(MIFARE_EREAD,
bytRecBuffer())
    If lngStatus <> 0 Then
        Error2Log (lngStatus)
    Else
        Text_Log.SelText = "Transmitting..." + vbCrLf
        Do Until addr >= 16
            If counter1 Mod 3 <> 0 Then
                current = 256 * current + bytRecBuffer(addr)
                counter1 = counter1 + 1
            Else
                current = 256 * current +
bytRecBuffer(addr)
                result = result + str(current) + ","
                current = 0
                If counter1 >= 510 Then
                    Print #1, result
                    result = ""
                    counter1 = 0
                End If
                counter1 = counter1 + 1
            End If
            addr = addr + 1
        Loop
    End If
End Sub

```

```

Do Until ((lngStatus <> 0) Or (bytRecBuffer(0) <> &H12))
    lngStatus = ISO14443_Mifare_ERead(MIFARE_EREAD1,
bytRecBuffer())
    If lngStatus = 0 Then
        Do Until addr >= 16
            If counter1 Mod 3 <> 0 Then
                current = 256 * current + bytRecBuffer(addr)
                counter1 = counter1 + 1
            Else
                current = 256 * current +
bytRecBuffer(addr)
                result = result + str(current) + ","
                current = 0
                If counter1 >= 510 Then
                    Print #1, result
                    result = ""
                    counter1 = 0
                End If
                counter1 = counter1 + 1
            End If
            addr = addr + 1
        Loop
        addr = 1
    Else
        Error2Log (lngStatus)
    End If
Loop

Close #1
If bytRecBuffer(0) = &H16 Then
    Text_Log SelText = "End of Data" + vbCrLf
End If
End Sub

```

This function also formats the data in the file, so that 170 columns are created in the resultant csv file, each representing a specific frequency. Each row of the file represents a specific mean and amplitude.

Appendix E

PUBLICATIONS

1. Mechatronics, University of Limerick, 8th June 2008

Radio Frequency Identification as a Method of Providing Signalling Information in Second Hand Computers

Eanna Cronin¹, Stewart Hickey and Colin Fitzpatrick

Department of Electronic and Computer Engineering
University of Limerick
Limerick, Ireland

ABSTRACT

The work presented in this paper proposes a technological means of extracting life cycle data from used computers. Using this data it is possible to signal the quality of the computer system, and reliably determine the reuse value of the system. The aim is to enhance consumer perception of the value of second hand computers and promote lifetime extension through reuse. This would help to mitigate a large portion of the environmental impact associated with personal computer manufacture, as well as reducing the volume of electronic waste.

1. INTRODUCTION

Failure rates in electronic systems have traditionally been modelled as a bathtub, with a large number of failures in the early stages of the life cycle (infant mortality) followed by a long period with very few failures (useful life). After this there is again an increasing failure rate as the components burn out when they reach the end of their usable life [1]. However, with personal computers, increases in reliability mean that the bathtub model no longer applies [2]. Extended lifetimes in electronics mean that system use tends to be discontinued long before the components become obsolete [3]. The result of falling computer costs is that people are willing to replace their system much earlier than is necessary, and in the absence of a reliable second hand market for personal computers, there a large volume of resultant electronic waste and a corresponding amount of unnecessary manufacturing taking place.

While the computer industry therefore represents a considerable burden on the environment, information technology has also been identified as a key element in the transition to sustainable development. It is predicted that with increased access to computer services will come decreased dependence on transport, as telecommuting and video conferencing become routine[4]. As such, increasing the prevalence of information and communications technologies should be a goal of an environmentally aware society, but it is also necessary to attempt to reduce the environmental impact of the manufacture of these technologies.

Initiatives put in place by governments are making it more desirable for manufacturers to build environmentally friendly systems[5, 6, 7]. These initiatives focus on the environmental effects of the manufacturing, use, and disposal stages of the product life cycle. EPEAT is a certification initiative which awards a gold, silver or bronze standard to products which meet a set of specific specifications. Included

¹ Corresponding author: Tel: (061) 233290; E-mail: eanna.cronin@ul.ie

in the EPEAT criteria are several regarding life cycle extension[5] Coupled with recent moves by various computer manufacturers to appear more environmentally friendly[8], these initiatives will foster demand for any systems which could cost-effectively introduce life extension to the PC market.

One solution to this problem is to reduce the environmental overhead of individual computer systems by fostering a second hand computer market. A major barrier to second hand computers has always been the inability to test the reliability of a system, and the absence of a guarantee to inspire consumer confidence. There are various benchmarking tests that exist to determine the level of functionality of systems[9], but given the decreasing cost of new computers, together with the time and energy it would take to test each individual second hand computer, it is unfeasible to expect a profitable return on tests of this kind. An optimal solution would be an automated health monitoring system with which it could be determined which type of secondary market process is best suited to the system.

2. ADVERSE SELECTION

Adverse selection is a principle from economics which relates to a market in which there is asymmetry of information[10]. The seller may know the value of the system being sold, but lacks a means of verifying this value to any buyers. The resulting lack of confidence drives down the market value of the goods. This causes people with high quality merchandise to be unwilling to sell, flooding the market with inferior quality goods, further decreasing the perceived value of the product. People with valuable goods are unable to get a fair price, so they hoard their goods until they are too old to be sold, and then dispose of them. In electronics this has become known as the closet effect, and results in both the unnecessary manufacturing of new systems, and the generation of electronic waste.

One reason for the difficulty in ascertaining PC value is that the applications of computers vary from user to user. In general, an office computer will mostly be used for word processing, spreadsheets, presentations, internet and email. These applications cause minimal stress to hardware and a computer used for only these purposes is under very little strain throughout its operational lifetime[4]. Routine upgrading or early adoption of new features means that a large number of fully functional systems are being discarded on a regular basis. In general, these systems will simply be thrown away, even when there is nothing wrong with their hardware.

On the other hand, one of the driving forces in the development of new computer hardware is the graphics requirements of modern computer games. People who use their computers primarily for playing games are placing the hardware under considerable stress throughout a systems lifetime. As well as the regular pressure to which a system is exposed when playing games, overclocking is an established feature of gaming culture whereby the voltage to the CPU is increased and the frequency at which it runs is boosted. This practice is not approved by hardware developers and frequently causes component failure owing to overheating. The stress of overclocking the processor decreases the reliability of the processor, and as such, a gamer's second hand computer might be considerably closer to failure at end of life than an office PC, despite the propensity of gamers to purchase new hardware on a much more regular basis, sometimes replacing or upgrading components as often as every six months. For this reason, it is unreliable to judge a PC's value based purely on its age, or the system specifications[12].

It has been shown that sellers can improve their market position by signalling their private information to less well-informed buyers[13]. An example of a market in which signalling is used is the second-hand car market, where the mileage of the car and the vehicle service history serves as a signal of the relative quality of the car being sold. This is a method of signalling which is accessible to people who may not have any great understanding of cars, giving them a simple metric with which to compare products.

By introducing a similar metric into the second hand computer market, it is hoped that adverse selection could be combated and that it would be possible to exploit the valuable resource of second hand computers. This would serve to reduce the environmental burden of both the manufacturing and the disposal stages of the personal computer's life cycle.

In order for a metric of this kind to be functional, it must reliably represent the buyer's perceived value of the computer system. Research has shown that the perception of value in the electronics is continuing system reliability. As such, it is necessary for signals to be indicative of system reliability, which is largely governed by the condition of the hardware. To ascertain the value of a system, it is therefore necessary to quantify the degradation of components that has occurred in the course of the life cycle to date. In order to estimate the extent of this degradation, we need to have access to information on the conditions in which the system is operating.

3. ACQUISITION OF LIFE-CYCLE DATA

Embedded sensors are used in many safety-critical applications as part of early-warning systems and for system performance analysis [14]. In mechanical, structural and biomedical systems the use of sensors to monitor system health is common practice [15-17]. The data retrieved from these sensors is monitored for advance warning of system degradation and imminent failure. Through analysis of retrieved data, systems can avoid reliability problems through preventative action.

In many electronic systems, sensors are used in order to provide warning of environmental dangers which can lead to catastrophic failure. It is difficult, however, to use sensors for health monitoring purposes because of the sub-micron scale of electronic devices. It is proposed that, instead of attempting to monitor the degradation in the device itself, the contributing environmental factors be monitored. These factors have been shown to be critical to the health of electronic systems and, moreover, to be relatively easily monitored through hardware.

4. FACTORS CONTRIBUTING TO COMPONENT FAILURE

By monitoring the principal causes and indicators of impending system failure, it is possible to quantify the reliability of a PC. A reliability metric should be indicative of system stability and likelihood of failure under normal operating conditions. Studies carried out into computer failure have identified the primary factors influencing reliability in electronic components [18], and by recording the influence of these factors on a computer system, we can develop an idea of the degradation which has occurred in the lifetime of the computer.

4.1 .OPERATING TIME

The most obvious characteristic of a computer's usage which would interest buyers is the total operating time. The number of hours for which a computer has been operating serves as a simple indicator of the age of the system, as well as the stresses on the hardware. It is not in any way comprehensive however, and we must look at operating time in combination with the conditions under which the computer is being operated if we want to develop a clear image of the usage pattern.

4.2. TEMPERATURE

Operating temperature has been shown to be one of the most critical factors in electronic component degradation [19]. High temperatures can, in the short term cause the degradation of solder connections and burnout of components, and in the long term cause slow deterioration in the reliability of circuits owing to chemical degradation effects. It has also been noted that time dependant temperature changes and temperature gradients can contribute towards damage to electronic systems, as temperature time gradients can cause mechanical stress to connections, particularly in cases where dissimilar materials are in contact[20, 21]. Using temperature sensors on various areas of the computer hardware allows the estimation of temperature gradients and the calculation of temperature cycles.

4.3. VOLTAGE

Voltage monitoring is another method of determining system health. Any deviation from the nominal voltage can be indicative of damage to the power supply [18]. Furthermore, people who overclock their computers tend to increase the voltage supplied to the processor, which can lead to the temperature effects already mentioned. By recording the voltage supplied to key components, it is possible to detect voltage spikes or deviations which would indicate degradation of components.

4.4. HARD DRIVE SMART ATTRIBUTES

Hard drives installed in PCs incorporate SMART monitoring which allows users to monitor the health of the mechanical and electrical attributes of the drive. Failures of hard drives are often characterised by the degradation of various attributes over time. Certain parameters logged by SMART technology (scan errors, reallocation counts, probational counts and offline reallocation counts) have been shown to be strong indicators of drive reliability [22].

Through the logging of the above parameters, a profile of the use of a computer can be built up over the course of it's lifetime. Using this profile, it is possible to determine a fair resale value for the computer, and this would stimulate a second hand market for computers [11].

5. EXTRACTION OF LIFE-CYCLE DATA

In order for logging of life cycle data to be a financially viable means of stimulating second hand computer markets, the extraction of data must be quick, easy and affordable. This would provide motivation for manufacturers to adopt the standard, as it opens up the possibility of generating revenue by selling refurbished or second hand computers.

5.1. RADIO FREQUENCY IDENTIFICATION

Radio frequency identification (RFID) is a method of transferring data over short distances whereby the power for the transfer is provided by the device receiving information. An electrical current is induced in the device using an electrical field and a resonant induction loop. This electrical current then enables the device to transfer data using the induction loop as an antenna.

RFID as a method of signalling end of life potential has been investigated under the PROMISE initiative, where sensors in various mechanical systems were used for condition monitoring[23]. RFID was used at end of life in order to extract life-cycle data and recommend a process for reuse, rebuilding or recycling. The technology was installed in cars, machine engines, and industrial machinery, and the data extracted was used by manufacturers to extract failure information from systems, as well as to perform predictive maintenance, where faults are avoided by performing maintenance on components based on probable future problems. This approach could also be applied to computer systems without the need for additional hardware over what would be required for end of life analysis.

5.2. ADVANTAGES

This technology is already in use for monitoring of electronics through the Electronic Product Codes (EPC) system. This is a system whereby electronic goods can be tracked from manufacture through to disposal using a globally unique product code [25]. This system has been hailed as the next generation of barcode, allowing global identification of individual products. Given the acceptance of RFID for the purpose of tracking and monitoring of electronics, it is well placed to also play a role in the monitoring of the usage of components. Also, RFID antennas are continuously becoming smaller and less expensive, and as such it would not represent a significant additional cost to include an antenna in a new system.

Using RFID, it is possible to retrieve data almost instantaneously from a computer system without the need for the system to be turned on. A PCI card with an RFID antenna incorporated, or an equivalent located on the motherboard, can allow the lifetime consumption and usage profile data to be read from computers without the need to power up the system. The retrieval of data could be performed on a conveyor belt structure, or using handheld readers, which are typical of recycling depots. The crucial element is the separation of reusable equipment before it is inappropriately handled and possibly irretrievably damaged. The data retrieved from the system in this way can then be subject to analysis, providing information which would be relevant to potential buyers, or to recycling depots.

5.3. DATA REQUIREMENTS

Because of the serial nature of an RFID link and the limitations placed on data volume by transfer speed, it is necessary for the data collected throughout the lifetime of the computer to be compressed so that it can be transferred in the time provided. The analysis collected from years of computer use must be communicable in a matter of seconds. One of the challenges of the project is the reduction of the data size without the introduction of errors or the loss of relevant data. Cycle counting methods have been proposed as a method of data reduction. These are used to capture the vital characteristics of a set length of data and at the same time reduce the amount of data without loss of information.

6. SECONDARY MARKET PROCESSES

Under WEEE, consumers can now have their electronic waste collected and recycled free of charge. With companies obliged to provide a takeback service when replacing electrical and electronic goods, they may be willing to perform life cycle data extraction themselves, and recover systems which are suitable for resale. These could then be sold under a new warranty, which would counteract the lack of confidence in second hand computers.

Alternatively, the data could be recovered at recycling depots, and operational systems appropriately handled there. There are a variety of possible routes for a second hand computer at the end of life. After

extracting the life-cycle data, it can be examined using RFID middleware, and a decision made on the preferred secondary market route for the system.

6.1. REUSE

A computer which has had a relatively short first life could conceivably be resold directly into a second life. The requirements for this would simply be the deletion of data on the hard drives and the reinstallation of operating systems, and as such it is the simplest and most environmentally beneficial method of generating a second-hand life.

Conceivably, even if the value of the computer is such that reselling would not be profitable, it would be possible to donate the working computers to charity. Availability of ICT has been identified as key to economic growth, and computer literacy is a valuable asset in getting better jobs in the labour market. As such, provision of computers to schools or disadvantaged areas could be a valuable reuse goal.

6.2. REBUILD

If reuse of a system is deemed unprofitable, it might still be possible to reuse individual components of the computer, while the remainder is sent to appropriate recycling facilities. Reusing components would serve to reduce the cost and environmental impact of building a new system. Similarly, if only a small proportion of the components of a system are not fit for reuse, these parts could be replaced, and the refurbished system resold.

6.3. RECYCLING

If a computer is beyond reuse or refurbishment, it is a waste product, and once identified as such should be treated appropriately. Under the Basel Convention, it is illegal to export electronic waste, however a large amount of waste is still exported under the guise of export for reuse[25]. If computers can be flagged as waste using RFID, then their export is controlled under the Basel Convention on the Control of Transboundary Movement of Hazardous Wastes and their Disposal [26], and it can be ensured that they are dealt with in the appropriate manner.

In Figure 1, the possible routes for end of life computer systems are shown. By separating waste from non-waste early in the reverse logistics process, it can be ensured that the waste systems are handled in the correct manner. Furthermore, by removing functional parts from non-functional systems, these parts can be utilised in the manufacture of new systems. This would aid in the prevention of the export of hazardous materials to third world countries.

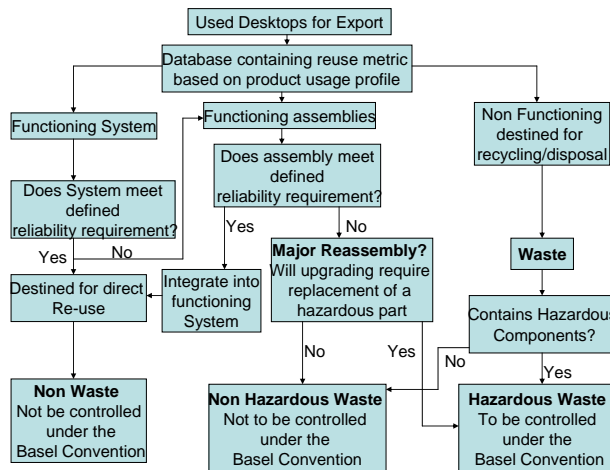


Figure 1: Routes for PCs showing removal of waste products before export

7. RESEARCH METHODS

The optimal RFID antenna and reader for the application must be selected. This will be achieved by assessing the requirements of read range, antenna size, and the properties of the environment in which it will be operating. The various methods of reducing life consumption data must then be analysed in order

to determine the one most suited to minimising data without compromising life cycle information. Once this has been completed, a suitable memory device will be selected for the application.

Once this has been completed, communication between the central processing unit and the memory device must be established. With the memory device located on a PCI card, life consumption data must be transferred through the PCI bus and stored on the memory device for extraction. Once this has been achieved, the RFID antenna can be incorporated into the PCI board, and integrated with the memory device.

The final part of the research will be the communication of life consumption data to the memory device while the computer is turned on, and the retrieval of the same data while the computer is turned off. Assuming correct operation of the system, the design could then be incorporated into a computer motherboard to provide for more economical storage and extraction of data.

It will be necessary to try to quantify reliability in such a way as to be capable of determining a fair resale value for the system. This will require the analysis of many second hand computers in order to ascertain the effects of the parameters being monitored, and a market survey to determine what price ranges would be achievable for second hand systems. On completion of this research, it will hopefully be possible to influence the design of new PCs in a way which would encourage the standardisation of a life cycle data extraction system.

8. CONCLUSION

A technological solution to more readily enable the reuse of PCs has been suggested. The solution utilises life consumption monitoring (LCM) as a means of determining suitability for reuse at end of life. RFID is proposed as a method of extracting critical LCM data from PCs in a cost and time efficient manner. This avoids the requirement of time intensive testing at end of life and also enables electronic waste to be identified and treated appropriately.

9. ACKNOWLEDGEMENTS

This work was supported by the Irish Environmental Protection Agency (EPA) and the Irish Research Council for Science Engineering and Technology (IRCSET) and in part by Microsoft Research through the European PhD Scholarship Programme.

REFERENCES

- [1] W. Q. Meeker and L. Escobar, *Statistical Methods for Reliability Data: John Wiley & Sons*, 1998.
- [2] K. Wong, "What is wrong with the existing reliability prediction methods?" *Quality & Reliability Eng'g Int'l*, Vol. 6, 1990, pp 251-257.
- [3] S. Hickey, C. Fitzpatrick, "An Onboard Diagnostic System for PC Re-Use Analysis", *Care Innovation, Electronics Goes Green*, Vienna, Austria, 12-16th Nov. 2006
- [4] M.W. Toffel, A. Horvath, "Environmental Implications of Wireless Technologies", *Environmental Science & Technology*, 2004
- [5] J. Katz, W. Rifer, A.R. Wilson, "EPEAT: Electronic Product Environmental Analysis Tool – development of an environmental rating system of electronic products for governmental/institutional procurement"
- [6] <http://www.federalelectronicschallenge.net/> (Accessed on 17-04-08)
- [7] <http://www.eco-label.com/default.htm> (Accessed on 17-04-08)
- [8] Dell Aims to go Carbon Neutral, http://www.businessweek.com/technology/content/sep2007/tc20070926_491108.htm (Accessed on 01-05-08)
- [9] PassMark BurninTest Software – PC Reliability and Load Testing, <http://www.passmark.com/products/bit.htm> (Accessed on 01-05-08)
- [10] G.A. Akerlof, "The Market for 'Lemons': Quality Uncertainty and the Market Mechanism", *The Quarterly Journal of Economics*, 1970. 84(3): p. 488-500
- [11] S. Hickey, C. Fitzpatrick, "Signalling information from embedded sensors to promote lifetime extension in electrical & electronic equipment", *Sensors and their Applications XIV*, Liverpool, UK, 11-13th September 2007
- [12] E. D. Williams, "Extending PC lifespan through secondary markets", 2003.

- [13] M. Spence, "Job Market Signaling", *Quarterly Journal of Economics*, 1973. 87(3): p. 355-374
- [14] G.J. Kacprzynski, M.J. Roemer, G. Modgil and A. Palladino, "Enhancement of physics of failure prognostic models with system level features," in *Proc. IEEE Aerospace Conf.*, vol. 6, 2002, pp. 2919–2925.
- [15] A. Lymberis, "Smart wearables for remote health monitoring, from prevention to rehabilitation: current R&D, future challenges". 2003.
- [16] W. Dai-Hua and L. Wei-Hsin, "Wireless transmission for health monitoring of large structures", *Instrumentation and Measurement, IEEE Transactions on*, 2006. 55(3): p. 972-981.
- [17] I. Tume and A. Bajwa, "A survey of aircraft engine health monitoring systems," in *Proc. 35 the Joint Propulsion Conference Proceedings 1999*, pp. 1–6.
- [18] "ELIMA - Environmental Life Cycle Information Management and Acquisition for Consumer Products: Final Deliverable Report D16", 2005
- [19] M. Osterman, "We Still Have a Headache with Arrhenius," *ElectronicsCooling*, February 2001, Vol.7, No.1
- [20] H.V. Nguyen et al, "Effect of Thermal Gradients on the Electromigration Lifetime in Power Electronics", *IEEE Annual International Reliability Physics Symposium*, Phoenix 2004
- [21] Lu, Z, et al. "Analysis of Temporal and Spatial Temperature Gradients for IC Reliability", *TECH. REPORT CS-2004-08 Mar 2004*
- [22] E. Pinheiro, et al, "Failure Trends in a Large Disk Drive Population", *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST'07)*, February 2007
- [23] *Product Lifecycle Management and Information Tracking Using Smart Embedded Systems*, <http://www.promise.no>
- [24] R. Weinstein, "RFID: A Technical Overview and its Application to Enterprise", *IT Professional*, 2005
- [25] The Basel Action Network, "The Digital Dump: Exporting Re-use and Abuse to Africa"
- [26] "Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and Their Disposal"

2. International Symposium on Sustainable Systems and Technology, Arizona,
May 2009

Developing RFID signalling to close the loop on second hand computers

Eanna Cronin, Pat Sweeney, Stewart Hickey, and Colin Fitzpatrick

Abstract—Because of the underdeveloped market for used computers, a large number of functional systems are going to waste. A significant barrier to the development of a market is the inability to communicate the worth of a computer. In order to overcome this barrier, a system for determining the value through monitoring of the conditions under which it is used is suggested. This system uses sensors already present in modern systems to monitor usage conditions, then utilises Radio Frequency Identification to transmit this information for analysis. The proposed system will streamline the reverse logistics process by providing a quick, simple method for the identification of reusable computers, and some of the technical issues are discussed in this paper.

Index Terms— End of Life Logistics, Personal Computers, Radio Frequency Identification, Reuse

I. INTRODUCTION

THE services provided by personal computers (PCs) have become indispensable to modern industrialized societies. This has resulted in an explosion in the number of computers in use throughout the developed

world. PCs have been suggested as a key element in the transition to a sustainable society[1]. They can reduce our reliance on fossil fuels through telecommuting and video conferencing, and also reduce the amount of paper being consumed in offices. On the other hand the power being consumed directly, by PCs, and indirectly, by server farms and other service providers, is a definite disadvantage of our increasing reliance on computers.

A considerable percentage of the energy consumed directly by a PC throughout its life is in the manufacturing phase. The energy required for the manufacture of a single PC and monitor has been estimated to be in the range of 3200MJ[2]. This manufacturing burden represents a significant portion of the energy consumed by the PC throughout its life cycle[3].

As well as the effects of manufacturing, the disposal of used computers must be considered when analysing their impact. PCs are regarded as having a relatively short serviceable life, 3-5 years in most cases, and the materials used in their manufacture are difficult to recover at end of life. This means that instead of being recycled in the appropriate manner, many used computers are simply exported to developing countries, with the stated goal of export for reuse. This is an effort to circumvent the Basel Convention, which prohibits the cross border transport of hazardous materials[4].

In recent years, the increasing reliability of electronics has meant that many PCs are actually being replaced due to obsolescence before they stop functioning. This suggests that there is a potential market for used computers, and reuse has been identified as an optimal method of alleviating the environmental burden of PCs[5]. Unfortunately, such a market has not yet

This work was supported by the Irish Environmental Protection Agency (EPA).

E. Cronin is with the Electronic and Computer Engineering Department, University of Limerick, Ireland (phone: +353-87-7752762; e-mail: eanna.cronin@ul.ie).

S. Hickey was with the Electronic and Computer Engineering Department, University of Limerick, Ireland (e-mail: stewart.hickey@ul.ie).

C. Fitzpatrick is with the Electronic and Computer Engineering Department, University of Limerick, Ireland (e-mail: colin.fitzpatrick@ul.ie).

developed owing to several factors.

A major barrier to the development of such a market is lack of faith in used electronics. In part due to the complexity of computers, consumers are unable to determine the reliability of a used system, and in the absence of any guarantee of quality, will place much lower value on a used computer than they would on a new one with similar specifications.

By using sensors, which are already present in modern PCs, it is possible to continuously monitor the conditions under which a computer is operating throughout its lifetime and hence determine the level of ongoing reliability which can be expected of the system as a whole, or of its individual components[6].

Using this, it is hoped to provide a guarantee of reliability for quality second hand systems which will encourage consumers who require lower-end machines to invest in a used PC, avoiding the manufacturing of a new PC and delaying the disposal of the old one. This will require the cooperation of PC manufacturers who will have to integrate the data, but it is expected that with the recent trends towards “greener” electronics, manufacturers will be endeavouring to make any

This system also has the potential to mark PCs as waste if they are determined to be beyond use. This will mean that these systems must be dealt with in the appropriate manner at end of life under relevant legislation. It is hoped that this could prevent the export of large amounts of electronic waste to developing nations, where if handled incorrectly, it can result in damage to the environment.

II. THE CHANGING PC MARKET: SUPPLY AND DEMAND

Moore’s law describes a trend in the rate of advancement of technology. It states that approximately every two years, a technology doubles its capabilities. This applies across most technologies, from processor speed to hard disk drive (HDD) capacity. This trend has been shown to be accurate in predicting the advancements in technologies over the last four decades[7].

Research carried out at the University of Limerick shows that people who own PCs rarely take advantage of the full processing power of their systems. Although the capabilities of computers have increased, the requirements of

many users have reached a plateau. When asked about the functions for which they use their computers, a majority of respondents said that their PCs were used mainly or exclusively for office or web applications.

TABLE 1:
PRIMARY FUNCTIONS OF PCs

Function\Response	1	2	3	4	5
Office Applications	9%	15%	14%	24%	36%
Email, Browsing, IM	4%	1%	6%	27%	61%
Games	53%	23%	13%	8%	2%
Programming	68%	11%	5%	11%	5%
Photo Editing	29%	14%	29%	23%	5%
Viewing DVDs	60%	21%	10%	8%	1%
Recording DVDs	72%	14%	9%	5%	1%

Consumers were asked to rate the frequency with which their PC is used for each purpose used on a scale of one to five, with one representing “Not at all” and five representing “All the time/Exclusively”. As can be seen from the results in Table 1, a high number of respondents use their computers mainly or exclusively for office and web applications while a large number of respondents rarely use their computers for the more intensive applications of gaming, programming or movie editing.

The requirements of office and web based applications increase over time, but only at an incremental rate, compared to the advancements in requirements for high end applications, such as gaming or video editing. Since high end applications are the ones which drive the development of technology, most average PC users are effectively wasting the potential of their computers.

III. BARRIERS TO A SECOND HAND MARKET

While the evidence of an increase in the environmental stewardship of manufacturers is encouraging, it does little to alleviate the concerns of consumers purchasing PCs. A majority of potential buyers surveyed said that they would not feel comfortable buying a second hand computer. The reasons they gave for their discomfort were primarily lack of knowledge about the previous lifetime of the computer and the absence of any guarantee of quality or warranty accompanying the system.

A. Endowment Effect

Further to the lack of confidence in buyers, sellers have also been shown to gain an attachment to their computers. In the market

research, when sellers were asked to estimate the value of their computers and also to suggest the minimum price they would accept, their estimated values were usually much less than the price they would accept. This is a market characteristic known as the endowment effect, whereby the owner of an object places more value on it simply owing to the ownership. This effect was clearly shown in our research, where, as shown in Table 2, 38% of respondents estimated the market value of their PC as less than €500, while less than 23% would be willing to accept that price.

TABLE 2:
ESTIMATED COMPARED TO ACCEPTABLE VALUE

Value	Estimated value	Acceptable value
<€100	10%	1.1%
€100-€199	5%	4.1%
€200-€299	9%	6%
€300-€399	8%	7.5%
€400-€499	6%	4.1%
€500-€599	7%	5.6%
€600-€699	4%	4.5%
>€700	18%	17.7%
Don't know	33%	49.2%

B. Adverse Selection

Lack of consumer confidence drives down the prices buyers are willing to pay while the endowment effect drives up the minimum price sellers are willing to accept for used computers. These combined effects result in a market where equilibrium cannot be achieved. As a result, in a process known as adverse selection, the only sellers willing to put their computer on the market are those who know that they are worth less than the value the market will support. As a result, the market becomes flooded with inferior quality products, which further damages consumer perception of the second hand market, driving down prices.

C. The Closet Effect

This can lead to the “closet effect”, whereby owners of computers are unwilling to sell them in the knowledge of the fact that they will not be able to get a fair price for their system. This results in a large number of high quality systems being stored until they devalue to the point where they can be sold, or until they are disposed of.

D. Information Asymmetry

The problems outlined above are a result of information asymmetry in a market. The functionality of the second hand computer market can be improved by increasing the

amount of information shared between buyers and sellers. If there was a method of providing verifiable information to buyers about the quality of systems, then a fair price could be decided within the market, accurately reflecting the quality of the goods on offer.

A similar situation to the one seen in the used computer market can be seen in the second hand car market, where in the absence of additional information about a vehicle, it is very difficult to make an informed decision about its value. Information symmetry in the used car market was achieved through the use of signals to communicate information. The signals chosen were the number of miles travelled by the vehicle. Buyers in possession of this sort of information would be in a much better position to make an informed decision about the value of the vehicle than they might be otherwise.

In order for any signalling process of this type to be effective, it must reduce the complex processes of gradual degradation over time or through use to a simple metric which appears transparent to the average buyer. In order to create a signalling metric for the second hand computer market it is necessary to first identify the factors which influence the degradation of PCs[8].

IV. SIGNALLING IN USED COMPUTER MARKETS

Research has been carried out into the major causes of degradation in electronics in order to ascertain which signals should be used. Several environmental factors have been identified as critical to the reliability of components. The most intuitive signal that can be used is simply the length of time for which the system has been powered on. This is a simple to understand metric, but not necessarily the most important one, as environmental factors have been shown to have significantly more influence when combined with the length of time the PC has been powered on.

A. Temperature

Temperature has been shown to have a strong influence on the reliability of a system[9]. High temperatures in electronic components can indicate damage which has already occurred, or can cause damage to components. In fact, an average temperature difference of just 10 degrees centigrade throughout its lifetime can halve the reliability of a system.

Power intensive applications, if run over a

long period of the life of a system, have been shown to have the effect of increasing the temperature of the system. As such, constantly running high demand applications can have the effect of shortening the lifetime of a PC.

In addition to the steady state temperatures of a system, rapid cycles in temperature, even between relatively low temperatures, can cause damage to the mechanical components of a computer as they expand and contract with changing temperature.

B. SMART attributes

Most hard disk drives (HDDs) in the current generation of computers feature Self Monitoring, Analysis and Reporting Technology (SMART), which monitors and records events as they occur in the HDD. Factors which are monitored include power-on time, read error rate and temperature, along with a range of others. Several SMART attributes have been identified as being relevant when determining the reliability of a system[10].

C. Monitoring critical factors

All of the above influences can be monitored through hardware already present in a modern computer. Intel's newest processors feature a digital thermal sensor which can perform temperature monitoring and averaging and output this information. These details, if stored and aggregated, would provide invaluable in estimating the depreciation a processor has undergone throughout its lifetime.

By combining the information from multiple sensors, an accurate estimation of a computer's value can be derived, hopefully allowing market equilibrium to be reached.

V. END OF LIFE

When a system is being replaced at end of life, there are several ways of dealing with the used equipment. Each route which a used PC can take will represent a different burden on the environment, and the goal of a signalling system is to ensure that the route taken is the one with the least negative impact.

In order to overcome the lack of consumer confidence in the second hand market, some sort of warranty must be provided with a second and system. In the case of computers, it can currently be difficult to ascertain the quality of a system without a time-intensive testing process. With an effective signalling process, however, the

systems with potential can be identified early in the process, allowing suppliers and refurbishers to focus on the systems which are most likely to prove viable for resale.

There are several possible methods for dealing with used computers at the end of life. Under existing European legislation, manufacturers are obliged to finance the take back of used electronics at end of life and given the correct market conditions they could well find it commercially beneficial to develop outlets for reused equipment. They would already have the necessary supply chain logistics in place for PCs and would also have the confidence of consumers and as such, should they begin dealing with used systems, they would not necessarily face many of the barriers to entry that would otherwise need to be overcome. This can be seen in the recent interest among certain manufacturers in developing the Individual Producer Responsibility (IPR) model of WEEE compliance in conjunction with the collective approach to Extended Producer Responsibility (EPR) that has emerged since the WEEE directive came into force[11].

Although some manufacturers may initially be resistant to a system which may appear to stifle new computer sales, it is important to note that every reused computer sold does not necessarily represent a lost sale of a new computer, as the price difference would allow consumers to buy second-hand systems who would not necessarily be willing to buy new systems. Also, the investment in a second hand system on the part of the manufacturer can be less than the investment needed in a new system, so the profit margin could be greater than that on a new system.

VI. END OF LIFE DESTINATIONS

Triage is a method of prioritising systems for analysis based on their probable potential for reuse. The process of triage is intended to streamline the reverse logistics process by eliminating many of the non-viable systems early in the analysis process, so that it is possible to focus mostly on the systems with high potential. By reducing the investment in wasted computers in this way, it should be possible for a greater number of systems to be reused, as the marginal profit from each individual reused system can be less without making the process uneconomical.

Reuse is regarded as the optimal end of life solution for computers[12]. This is because it

requires the least amount of energy investment before the system is in a state suitable for reuse.

An alternative to reuse is refurbishment, whereby individual components of a system can be reconstituted into a new PC. Computers are very simple to disassemble and reassemble, and many of the parts which make them up are essentially interchangeable between systems. This means that, even if a system is deemed unsuitable for reuse owing to a critical flaw, with the replacement of some parts it might make a more viable prospect.

After all the usable parts have been removed, the volume of waste should have been significantly reduced. By removing the usable components and labelling the remainder as waste, it will be possible to ensure this reduced waste stream is then handled correctly under appropriate legislation.

While not every PC which is analysed may be deemed profitable for reuse, there is a benefit to be gained from operational computers which may not be sold. From a manufacturer's point of view, since recycling represents a cost to them, there is motivation to keep computers out of the waste stream for as long as possible.

VII RFID DATA EXTRACTION

The technology to be used in order to determine a system's reusability must be cost effective, so that it can economically be included in computers without affecting their price. It must also be capable of quickly and automatically communicating information. In order for the process of triage to have maximum effect, a wireless technology which doesn't require the system to be powered on was optimal.

Radio Frequency Identification (RFID) was chosen for this purpose as it meets the above criteria. The technology is already in use for the purposes of identifying and recovering reusable and valuable parts of systems at end of life[13], which is a similar application to the proposed signalling system. This system would represent advancement over previous ones of this type, however, by offering information about the condition of the system.

There are a selection of different RFID standards, each with different benefits and drawbacks with relation to this application. In any RFID data transfer, the transfer of information is initiated by the presence of a reader in the area of an antenna. The antenna

then transmits data to the reader using the specified frequency. The means of transfer is largely dependent on the amount of data to be transmitted.

In passive RFID, all the energy for the transfer is provided by the reader. The antenna then utilises this energy to transfer a small amount of data back to the reader. In general these are used for simple identification purposes, as the small amount of data to be transferred is ideal as a next generation of bar code, where the antenna can transmit a unique code to the reader, remotely identifying the item to which the antenna is attached.

Active RFID systems feature antennas which are constantly in an operational state, and which await contact from a reader to initiate communication, which is powered by a battery connected to the antenna. These are suitable for applications which require the transfer of larger amounts of data, as the amount of power provided by the reader in passive RFID communications is enough only to transmit small amounts of data.

In semi-passive RFID communications, the data transfer is initiated when an antenna comes into range of a reader. The reader initiates the data transfer process by "waking" the antenna with a specific command. Then, powered by an on-board battery, the antenna transmits the stored data to the reader for analysis. This form of communication was determined to be the most suitable for the application proposed, as the amount of data to be transferred is too great for passive RFID, while the length of time for which the antenna will be employed is at least several years, ruling out the short lifetime of active RFID antennas.

In addition to the various methods of powering RFID transfers, there are also a variety of frequencies outlined in the RFID standard. Each offers a trade-off of benefits and costs to the type of communication required by this system.

The various frequencies of operation of RFID offer different data rates and read range. Low frequency operates between 125 and 135 kHz, offering a low read range combined with a very low data rate. Read range is dependent on antenna size, but for low frequency tags, is generally in the range of about 10cm. The advantages of low frequency are the low cost of tag production combined with its tolerance of obstacles and liquids, making it suitable for asset tracking of livestock .

High frequency (HF) RFID tags, which

operate at 13.56 MHz offer the best balance between data transfer speed and range, with a maximum potential data rate of 848 kilobits per second, and a range, depending on the reader, of up to 1 meter. HF is relatively tolerant of obstacles, although for the application with PCs, the tag would need to be located on the outside of the computer case, as the radio waves cannot propagate through metal.

This means that the means of data extraction would need to be considered at the design stage of the computer, as the layout of the tag on the case will determine the requirements of the end of life analysis. Using this technology, handheld readers or readers on a well designed assembly line could be used to read the lifetime consumption information stored in a PC and relay this information to a central system for processing. The used PC can then be automatically directed to the end of life stream to which it is best suited.

VIII. DATA VOLUME REDUCTION

Owing to the amount of data which would need to be stored throughout the lifetime of a PC and the limitations imposed by the data rates of RFID, it will be necessary to limit the volume of data being transmitted. As such, methods for reducing data volume without affecting integrity are being examined.

Temperature is a constantly changing environmental stress. One method of analysing such a complex load is to divide it into blocks of constant amplitude loads using a process known as 'cycle counting' this method is used to transform a time history consisting of peaks and valleys into a cycle history. Cycle counting methods are usually employed in systems which monitor fatigue and stress in materials.

To analyse and monitor these device stresses, cycles which place less stress on the device and cause less damage can be removed from the load analysis. A cycle occurs when the applied load returns the material to the state it was in before the load occurred. When the load reaches a value at which the stress was previously changing in the reverse direction a full cycle occurs. The stress path beyond this point is the same as if the initial loading had not occurred and it no longer affects the behaviour of subsequent cycles[14]. For temperature analysis one cycle is identified as a temperature increase then a corresponding decrease.

Amongst the most common methods of

reducing data to cycles are:

- Level counting
- Peak counting
- Simple range/mean counting
- Rainflow Counting
- Probability density function (PDF)

Of these the rainflow counting method and its variations are the most widely accepted cycle counting techniques as it can handle non-repeating time histories and results in the least amount of data loss. Using data reduced by cycle counting, damage calculation times can also be greatly reduced.

A. Case Study: Rainflow Temperature Cycle Counting

Research has been carried out in the University of Limerick into the efficacy of rainflow cycle counting for this application. By monitoring the temperature of a PC using software, a plot of the temperature was obtained, and cycle counting was carried out on the sample.

The main purpose of this cycle counting technique is to ensure that the more relevant data is highlighted among the rest of the data. The mean temperature at which cycles occur needs to be recorded, as unlike stress measurements the temperature is not likely to fluctuate about zero and is likely to cycle about a varying mean temperature. This method of counting also allows for the use of a cut-off where only temperature cycles above a certain level are recorded.

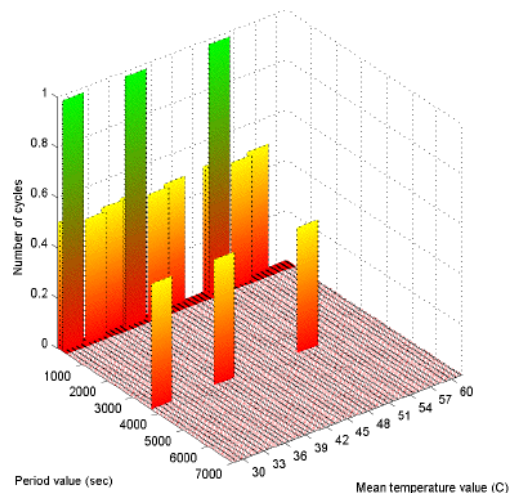


Figure 1: Results of period and mean temperature analysis

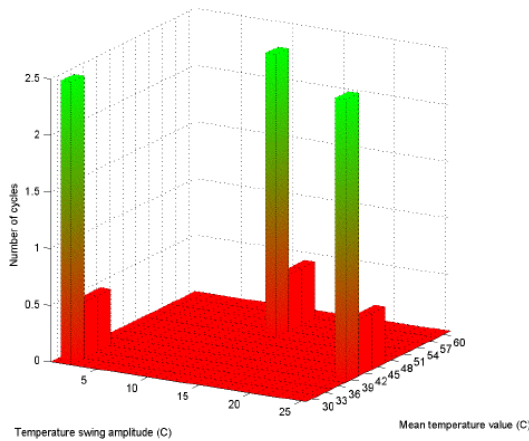


Figure 2: Results of temperature swing vs. mean temperature analysis

Before cycle counting the temperature data must be filtered to turning point data only, this means that all time history would be lost if time indexing of the inputs did not take place. There are several filtering techniques used to reduce data, the method used in this application is known as Ordered Overall Range filtering. This algorithm must be developed to suit the temperature data being analysed.

The graphs in figure 1 and figure 2 show the results of cycle counting in a PC. Figure 1 is comparing the number of cycles at each mean temperature with each period, while figure 2 shows the same results compared under the headings of mean temperature and temperature swing amplitude. In order to capture all of this data, it is necessary to express the results as a four dimensional array.

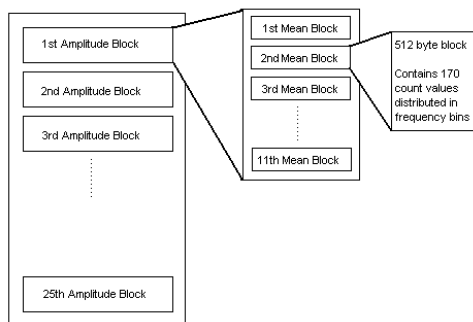


Figure 3: Block diagram of data storage methodology

For the purpose of the research, a limit of 145 kilobytes was put on the data to be stored, owing to the limitations of the SD card being used, and as this would represent a transfer time of approximately one second using high-frequency RFID. This was broken down, as shown in figure 3 above, into 25 blocks, each representing a specific temperature swing, with a resolution of one degree centigrade. Within each of these blocks is 11 further blocks, each representing a

different mean temperature, with a temperature resolution of three degrees centigrade. Finally, within each mean block are 170 frequency bins, of 6 bytes each, each representing a specific period of temperature swing, with a resolution of 30 seconds. The number in a bin is incremented every time a temperature cycle of that frequency, mean and amplitude occurs.

In the worst case scenario, where the same temperature swing about the same mean is exhibited throughout the lifetime of the system, that specific bin would take almost 8 years to fill. This represents a very unlikely confluence of conditions, and would be very improbable in practice. Nonetheless, for comparison purposes, if temperature values were to be constantly stored using the same sample rate over the same period, over 8.3 MB of storage space would be required, at one byte per sample. Therefore, through the implementation of this data reduction method, the volume of data is reduced to less than two percent of its potential maximum size. To transfer this data using RFID at 848 kilobits per second would take in excess of one minute. Swift transfers of data are required, as interference is common in industrial environments, and through the reduction of the length of time taken to transfer data, the probability of an error is reduced.

One restriction of rainflow counting using the ordered overall range method is that the algorithm must be applied to a block of data, and cannot be implemented on real-time, running data. This places some restrictions on the maximum temperature swing that could be recorded, as device memory constraints dictate how large the maximum data blocks can be, and thus how large the maximum possible recording period is.

IX. CONTINUING WORK

While it has been established that there is potential for a used computer market to develop, such a market would require significant support from both manufacturers and customers.

In order to gain confidence in the marketplace, it must be demonstrable that the outputs of a signalling system of this type can accurately determine the reliability of a PC. In order to achieve this, testing must be carried out to determine the effect each of the critical factors will have on the reliability of a PC, and how they will interact to affect reliability.

Further testing is also required to determine the optimum performance criteria for the RFID

data retrieval system. Various data reduction algorithms are under examination in order to identify one which can reduce the volume of data without compromising the integrity of that data. Also, the positioning of the antenna on the PC needs to be examined to ensure that its location allows the maximum data throughput.

REFERENCES

- [1] "Smart 2020: Enabling the Low Carbon Economy in the Information Age," The Climate Group 2008.
- [2] E. Commission, "Preparatory studies for Eco-design Requirements of EuPs: Lot 3, Personal Computers (Desktops and Laptops) and Computer Monitors, Final Report," ed: European Commission, 2007.
- [3] E. D. Williams, R. U. Ayres, and M. Heller, "The 1.7 kilogram microchip: Energy and material use in the production of semiconductor devices," *Environmental Science & Technology*, vol. 36, pp. 5504-5510, Dec 15 2002.
- [4] S. D. Murphy, "The Basel Convention on Hazardous Wastes," *Environment*, vol. 35, pp. 42-44, Mar 1993.
- [5] E. D. Williams, "Extending PC lifespan through secondary markets," *2003 IEEE International Symposium on Electronics & the Environment, Conference Record*, pp. 255-259, 382, 2003.
- [6] A. Ramakrishnan and M. G. Pecht, "A life consumption monitoring methodology for electronic systems," *Ieee Transactions on Components and Packaging Technologies*, vol. 26, pp. 625-634, Sep 2003.
- [7] E. Mollick, "Establishing Moore's law," *Ieee Annals of the History of Computing*, vol. 28, pp. 62-75, Jul-Sep 2006.
- [8] S. Kara, M. Mazhar, H. Kaebernick, and A. Ahmed, "Determining the reuse potential of components based on life cycle data," *Cirp Annals-Manufacturing Technology*, vol. 54, pp. 1-4, 2005.
- [9] N. Vichare, P. Rodgers, V. Evely, and M. G. Pecht, "In situ temperature measurement of a notebook computer - A case study in health and usage monitoring of electronics," *Ieee Transactions on Device and Materials Reliability*, vol. 4, pp. 658-663, Dec 2004.
- [10] S. Hickey and C. Fitzpatrick, "An Onboard Diagnostic System for PC Re-Use Analysis," presented at the Electronics Goes Green, Vienna, Austria, 2006.
- [11] (20/02/2009). *IPR Works*. Available: <http://www.iprworks.org>
- [12] E. D. Williams and Y. Sasaki, "Energy analysis of end-of-life options for personal computers: Resell, upgrade, recycle," *2003 IEEE International Symposium on Electronics & the Environment, Conference Record*, pp. 187-192, 382, 2003.
- [13] A. G. Kulkarni, A. K. N. Parlikad, D. C. McFarlane, and M. Harrison, "Networked RFID systems in product recovery management," *2005 IEEE International Symposium on Electronics & the Environment, Conference Record*, pp. 66-71, 262, 2005.
- [14] S. Mishra, S. Ganesan, M. Pecht, and J. S. Xie, "Life consumption monitoring for electronics prognostics," *2004 IEEE Aerospace Conference Proceedings, Vols 1-6*, pp. 3455-3467, 4192, 2004.

3. 16th CIRP International Conference on Life Cycle Engineering, Cairo, 2009

An RFID solution to promote PC lifetime extension

Eanna Cronin¹, Stewart Hickey¹, Colin Fitzpatrick¹

¹ Department of Electronic and Computer Engineering, University of Limerick, Ireland

Abstract

The increase in the pervasiveness of personal computers throughout the developed world is widely regarded as indicative of improvement in the quality of life. An increasing number of people require computers in their day-to-day lives. One unfortunate corollary of the increase in computer manufacturing is the resultant increase in the number of computers which will have to be dealt with at end of life. This paper suggests a solution which will deal with the increasing waste stream by offering a method of providing a swift first diagnosis of used computers, and determining whether a system is best suited for reuse, refurbishment or recycling

Keywords:

Personal Computers, Lifetime Extension, Radio Frequency Identification, Eco-design, Life-cycle management

1 INTRODUCTION

Although personal computers (PCs) serve many important purposes in society, their increasing prevalence represents a significant burden on the environment [1]. Many PC owners will replace their systems every 4-6 years [2], which requires the disposal of the used system, as well as the manufacture of its replacement. In corporations this replacement regularly takes place regardless of the functionality of the used systems.

Reuse of computers is the optimal means of reducing their environmental burden, as it reduces the impact of the manufacture, use and disposal stages of the product's life cycle [1]. Despite this fact, the second-hand market for PCs has never thrived.

One barrier to reuse of PCs is a perceived lack of reliability of second hand electronics. This is in contrast to recent trends, as electronics have been becoming more reliable as manufacturing methods mature and traditional models for estimating failure rates no longer apply [3].

One means of overcoming the lack of confidence in this regard is through the provision of a system guarantee. Market research shows a significant increase in the number of people who would be willing to purchase second hand PCs given some sort of warranty.

Using sensors embedded in most modern PCs, it is possible to continuously record factors which influence the reliability of a system and hence estimate the continuing reliability at the end of life. By providing a projection of continuing reliability, it is possible to give consumers an assurance of quality, which would restore some confidence in second hand systems.

In order for a system of this type to be viable, the extraction of reliability signals needs to be possible in a

cost- and time- efficient manner. Using embedded radio frequency identification (RFID) technology, it would be possible to extract data from a system while it is powered off, allowing the large scale retrieval of signals from used computers.

If implemented in the system design stage of PCs, this process could revitalise the second hand market for personal computers in the consumer to consumer, business to consumer, and even the business to business markets.

2 MARKET DEMAND FOR PCS

The amount of people who require the latest technology in PCs has consistently represented a small proportion of the overall market. Nonetheless, when it comes to buying a PC, there is still an attitude among buyers that the best technology will allow a longer period before they need to replace the system again. This attitude may be correct if the user's requirements are changing along with the capabilities of computers, as is the case with some gamers and software developers. For a large number of people, however, the requirements they have of their PCs are being met by the current generation of computers.

Market research carried out at the University of Limerick in March 2007 confirms this point. The research was carried out into the functions for which people use their PCs both at home and at work. Respondents were asked to estimate the frequency with which they used their computers for each function. The answer '1' represents 'not at all', while '5' represents 'very often/ all the time'.

Function\Response	1	2	3	4	5
Office Applications	9%	15%	14%	24%	36%
Email, Browsing, IM	4%	1%	6%	27%	61%
Games	53%	23%	13%	8%	2%
Programming	68%	11%	5%	11%	5%
Photo Editing	29%	14%	29%	23%	5%
Viewing DVDs	60%	21%	10%	8%	1%
Recording DVDs	72%	14%	9%	5%	1%

Table 1: Market research results on PC Use

From the results in Table 1, it appears that a large majority of respondents used their PCs mostly or exclusively for office and web based applications. 60% of respondents said that they used their computer for office applications such as word processing, spread sheets or presentations either 'often' or 'very often', and 88% replied that they used their PCs for web browsing, email and instant messaging either 'often' or 'very often'.

The number of people who said they were using their PCs for the more processor intensive applications of gaming and viewing or recording DVDs was much lower, with 53%, 60% and 72% respectively indicating that they never used their PCs for these functions.

While there are some incremental advances in web-based and office applications, their requirements do not place much stress on a processor [4], and as such users require less frequent upgrades than those using their PCs for gaming or video editing.

This plateau in requirements has been noticed by computer manufacturers, as evinced by the recent development of 'netbook' type notebooks which, rather than using technological advances to achieve higher processor power or greater component density, appeal to the average user by reducing the cost and size of the computer [5]. They are advertised as useful mostly for web browsing and office applications. These smaller, cheaper systems are an example of a disruptive technology, as they are changing the definition of quality in the computer market in a way which contrasts with previous trends.

Further evidence of the stabilisation of consumer demand for power can be seen in the Energy Star rating, which now categorises PCs according to their specifications.

Given this change of focus in the PC market, it is logical to expect that a market for second hand computers can develop strongly in the coming years. The low cost market shows that there is demand for reasonably priced PCs, which can be supplied by people replacing their own computers.

2.1 Environmental electronics

Furthermore, manufacturers and suppliers of computers are being encouraged to make their processes and products less damaging to the environment. Initiatives such as EPEAT in the US, the Eco-Design framework and Energy Star award recognition to manufacturers who conform to environmental standards [6-8]. In the US, certain federal bodies will only purchase PCs which have achieved EPEAT certification [9], providing great incentives for manufacturers to continue seeking to improve their environmental performance.

Electronics take back schemes are already in existence, and in particular, several companies are offering money for returned mobile phones or gadgets, basing the

amount offered on the brand and model of the device on offer [10, 11]. These companies make money by switching parts from used devices to make refurbished products. This approach would not be quite so simple with PCs as the modular nature of a computer means that a single model could have any of dozens of potential hardware configurations, making estimating its value difficult.

There is also evidence that various PC manufacturers are trying to become more environmentally aware, with companies such as Dell, Intel and HP producing 'green' products, which feature lower power consumption, longer life and clean recycling of components [12].

3 CONSUMER ATTITUDE

While the evidence of a market for lower powered PCs is encouraging, it does little to improve the attitudes of the public towards second hand computers. Consumers regard second-hand systems as unreliable in the absence of any evidence to the contrary, and as such wouldn't be inclined to buy used PCs, particularly given the ready availability of reasonably priced new alternatives. Research has reflected this fact; with 76% of respondents saying that they wouldn't consider buying a second hand PC.

3.1 Adverse Selection

A major discouraging factor in the second hand computer market is the perceived lack of reliability of the systems on offer. This reduces the value which people are willing to pay for second hand systems.

As a result sellers with good quality computers are less likely to accept the value being offered by the market for their system. This drives these sellers out of the second hand market, resulting in a greater proportion of low quality systems available. This has a sustained adverse effect on consumer confidence in the second hand market, further driving down the market value for used PCs [13].

This trait, known as adverse selection, was first identified in the used automobile market, where the mix of good and bad quality vehicles on offer was confusing to many potential customers. This was resulting in the owners of quality vehicles being unable to acquire a fair price, meaning the market became flooded with 'lemons'.

In the used PC market, this can be seen where the market is flooded with low-quality goods, while the owners of higher value PCs are forced to take them off the market as they are unable to get a fair price. This results in the better PCs normally being stored for an extended period of time before being disposed of, in what is known as the closet effect.

3.2 The endowment effect

As well as this, there is the endowment effect to consider, a trait which means that people place more value on something they already own than they would on an identical product which doesn't belong to them [14]. This effect has also been observed in research, where participants were asked to estimate the current value of their PC, and also asked what price they would accept for the system.

The results of this enquiry are shown in Table 2. The 'Estimated value' column shows the percentage of people who think their current PC is valued in each range, and the 'Minimum accepted' column shows the percentage of people who would accept each price range for their PC.

	Estimated value	Minimum accepted
<€100	10%	1.1%
€100-€199	5%	4.1%
€200-€299	9%	6%
€300-€399	8%	7.5%
€400-€499	6%	4.1%
€500-€599	7%	5.6%
€600-€699	4%	4.5%
>€700	18%	17.7%
Don't know	33%	49.2%

Table 2: Endowment effect on the perceived values of PCs

While 38% estimated the value of their PC at less than €500, less than 30% said they would be willing to part with it for less than €600. This suggests that the value perceived in a PC by its current owner is greater than the value perceived by the market.

This could affect sellers, who would feel an attachment to their own computers, and may therefore be unwilling to part with them for the value being offered by the market. The conditions required in a market in order for trade to occur are that the value placed on a product by the buyer must meet or exceed the value placed on it by the seller.

3.3 Information Asymmetry

The problem identified in the second hand market for automobiles is very similar to the one which exists in the PC market. The seller is the only person with an idea of the genuine value of the product on sale. In the absence of a system for communicating this information, people with lower quality goods will settle for the value being offered by the market. The owners of higher-quality goods will be unable to get a fair price for their products, and withdraw from the market.

4 SIGNALLING

In order to overcome the problem of information asymmetry and hence avoid adverse selection in the market, it is necessary to attempt to communicate the knowledge held by the seller to the buyer. By achieving symmetry of information in the market between the people supplying the product and the people buying them, a price can be agreed upon as both parties would have a better image of the factors influencing the value of the system.

The central consideration in designing a signalling process is in the reduction of the complicated processes of the product into a simple metric which can be understood by an average user. In the automobile market, while the majority of people do not know about the exact operations of the engine of a car, they can easily understand the number of miles it has driven, which is why the odometer reading was adopted in the second hand automobile market as an ideal indicator of the reliability, and hence the remaining value of the car.

Other signals which were adopted by the used car market include the vehicle service history and tags such as 'one lady owner'. All of these factors combine to give a buyer a more comprehensive idea of the value of the vehicle, and its remaining utility.

5 FACTORS INFLUENCING RELIABILITY

In order for signalling to be used in an effective way, it must be accepted by both manufacturers and the public as an accurate means of determining reliability. As such, a survey has been carried out in order to determine what the public consider to denote reliability. The survey respondents were asked what they would consider to be relevant factors when they were buying a second hand computer. The overwhelming response was that people considered the number of crashes a computer has had to be indicative of its continued reliability.

Several sources have shown that the factors which actually influence reliability are more likely to be environmental factors, which can be easily monitored using sensors and recorded within a PC throughout its lifetime.

5.1 Operating time

While this may seem to be an obvious metric for determining the value of a system, research has suggested that while leaving a computer running may not be environmentally ideal, it isn't necessarily a valid indicator of depreciation sustained by a system. In fact, it may be worse for a PC to be switched off and on, owing to stresses placed on components during boot up and shut down, than to be left running for sustained periods of time.

On the other hand, operating time in combination with the conditions under which a computer was being used can have an effect on its reliability, and is a simple to understand signal to use as the basis of a signalling system.

5.2 Operating temperature

Research has shown temperature to be a very clear indicator of reductions in the continued reliability of a PC. High temperatures can both cause and indicate damage to computer components. A steady temperature difference of just 10 degrees centigrade can halve the reliability of a computer [15]. This temperature difference is not atypical of two differently designed cases under normal operating conditions.

A high temperature can indicate a poorly cooled case, or a blocked air intake, or could be indicative of components running at high speeds. Regardless of the cause of the high temperatures, it can be catastrophic to the reliability of the system as a whole. Ensuring that a computer is maintained within a safe temperature range by using adequate cooling solutions and ensuring there is air circulation through the system is vital to extending life.

Research has shown that increasing the demands on the processor of a PC has the direct effect of increasing the temperature at which the system is running [4]. As such, it can be surmised that running power intensive applications and processes have the effect of increasing case temperature and hence reducing system life.

5.3 Temperature cycles

As well as steady state temperatures, changing temperature can impact reliability. Damage to components accumulates every time the temperature goes through a cycle in temperature [16]. Temperature can go through cycles as a result of power management profiles which cause a system to go into sleep mode too quickly, requiring that the system cool down and heat up repeatedly. They can also happen when a computer is shut down and restarted after a short period of time.

The major effects of temperature cycles are to mechanical components, where expanding and contracting materials will put stress on joints and can lead to total mechanical failure. Some computers have variable speed fans which vary their speed depending on the temperature of the core components. These devices help to minimise temperature cycles by maintaining steady temperatures in a system.

5.4 Power use

It is established practice among gamers to overclock the processors of their computers. This is achieved by increasing the frequency at which the processor runs by increasing the voltage supplied to it above the manufacturers' recommendations. While experienced gamers are capable of attaining higher clock speeds and increased performance in this way, it is not uncommon for the process to result in the processor being burnt out. This is because the increased speed results in an increased processor temperature, and may require the disabling of safety features in the processor.

5.5 SMART attributes

Hard disk drives (HDDs) in current generation computers feature Self Monitoring, Analysis and Reporting Technology (SMART), which monitors and records events as they occur in the HDD. Factors which are monitored include power-on time, read error rate and temperature, along with a range of others. Several SMART attributes have been identified as being relevant when determining the reliability of a system [17].

5.6 Monitoring critical factors

Both steady state and variable temperature can be monitored using inexpensive sensors. The latest generation of Intel processors, and most PC motherboards feature temperature sensors which are used to control fan speeds and to reduce processor speed at times of high thermal stress. The outputs of these sensors are visible to both software and hardware, and can easily be monitored and recorded to provide an indicator of the conditions under which the system was used throughout its lifetime.

SMART attributes are monitored by the HDDs. The attributes determined to influence the reliability of the drives can be recorded throughout the lifetime of a computer and recovered for the purpose of determining the value at end of life.

While exact quantification of the amount of damage occurring as a result of any or all of these factors is difficult, it is hoped that an indication of depreciation will improve the information exchange between buyer and seller and improve the chances of market equilibrium being achieved.

6 END OF LIFE LOGISTICS

A central factor which requires consideration when designing a process of this type is the means of extracting information. In order for the process to work and be economical for companies, it must be affordable. While it makes economical sense to devote amount of money to the recovery and reselling of operational systems and their components, investment of money in unprofitable systems should be minimised. Therefore the method for separating viable systems from unusable ones must be quick and low-cost.

6.1 Triage

Triage refers to a method of quickly prioritising systems based on their potential for reuse. By streamlining systems for reuse, recycling or refurbishment, the cost of identifying viable systems is reduced. A more costly, in-

depth analysis can then be performed on systems deemed to be of value, to determine the best means of reuse for each system. Triage is suited to situations in which a large volume of systems are being assessed for viability.

Triage can be performed by manufacturers, who are obliged under WEEE legislation to remove old computers when supplying new ones [18], and would be well placed to determine the value of systems, and to refurbish and redistribute used PCs. Alternatively, the triage could be carried out at recycling depots, where functional components can be recovered from systems before they are sent to the energy-intensive recycling process.

Once viable systems and components have been identified, more work can be carried out to determine their value based on market factors and system history. In this way, a large amount of money is not being invested in thoroughly testing systems which have little or no resale value.

6.2 Radio frequency identification

This separating method requires a simple, fast, technological method for the analysis of the data recorded during the lifetime of the PC. Radio Frequency Identification (RFID) technology has been suggested for this purpose. This is a low-power wireless communications technology, in which the energy for the data transfer is generally provided by the reading device.

Research has been carried out into the optimal type of RFID technology for this particular application. The limitations of the specific RFID standard chosen will determine the capabilities of the process, as different technologies will offer different ranges and data rates. Consideration also needs to be given to the placement of the RFID tags, as data cannot be transferred through some materials, and it is likely that the metal case of a computer would interfere with the data transfer process. As a possible means of overcoming this limitation, it is proposed that the RFID tag be located on the outside of the case.

It is likely that the technology used to transfer data will be a semi-passive RFID. In this standard, the transfer is initiated by the reading device, but the power for the transfer is normally provided by a battery attached to the RFID antenna. The advantage of this is that the reader only has to transmit once in order to 'wake' the tag, and a significant amount of data can then be transferred from the tag to the reader.

High frequency RFID, which operates at a frequency of 13.56 MHz, has been selected as the best mode of data transfer, as it offers a balance between a long range and a high data rate.

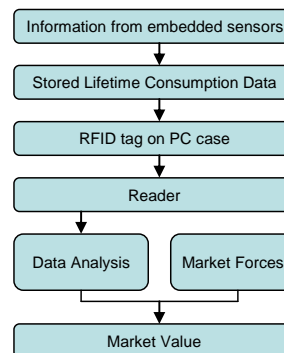


Figure 1: Data flow diagram showing determination of value

The overwhelming advantage of RFID as a data transfer technology for this application is the fact that the technology does not require the PC under test to be powered on or even plugged in for the data to be retrieved. As a result of this fact, the data retrieval can be carried out on a large scale, for example on a conveyor belt structure, allowing the total automation of the initial process of triage.

6.3 In-depth analysis of reuse potential

After triage, systems which are identified as useful would need to be analysed to determine the method of reuse best suited to the system. Owing to the modular nature of computers, it is possible that certain components would have to be replaced in order to obtain maximum profit from a PC. This makes it relatively easy to resell computers, and is another reason for the triage to be carried out at computer manufacturing plants. The manufacturers would be easily able to direct components for the manufacture of refurbished systems.

Another advantage of the low cost of an effective triage system is that, as well as identifying systems which could be profitable, it can also identify systems which are beyond resale, but suitable for reuse for charity purposes. These systems could be used for helping to bridge the 'digital divide' by helping low-income households to obtain an early education in ICT, which has been identified as central to attaining employment in later life [19].

The signals being used may need to be tailored to the application of the PC. In the business to business market, it is likely that companies will have some knowledge of the factors that contribute to degradation, and as such, an in-depth set of signals would allow them to make a decision regarding value. On the other hand, when selling directly to consumers, simpler signalling may be preferable, in order to be understandable to a larger portion of the market.

6.4 Value indicator software

In the smaller scale market of consumer to consumer sales, where a commercial evaluation method isn't being used it will be necessary to evaluate computers' value on a case by case basis. In these situations, if the recorded life consumption information were accessible by the owner, they could use this information to communicate the value of their system to potential buyers. The method for extraction of data for this purpose has not yet been examined, but it is likely that the recorded information would need to be accessible through software.

In order for software of this type to be reliable, it will have to provide accurate and secure information about a system. The information provided must not be reproducible by any other means, or people would be capable of changing the apparent value of their systems, which would damage confidence in the process.

7 CONTINUING WORK

While the proposed process would enable the development of a market for second hand computers, it requires the intensive testing of a number of systems in order to determine the exact effect of each of the influencing factors on the reliability of a system.

An extensive set of experiments must be carried out to determine the capabilities of a system of this type. Communication must be established between the central processing unit (CPU) of the computer and a device on a PCI card which will reduce the data and store it in

memory. Data must then be extracted from the memory using an RFID tag. Tests will determine the range of the system and the optimal configuration of the tag and reader, as well as the data volume which will be possible.

Further work is required to ensure the security of the data recording system. It is important that the information in the computer be tamper-proof so that people can't artificially inflate the value of their PC by modifying the outputs of the sensors or the recorded data, which would have the effect of further damaging consumer confidence in the second hand market, worsening the effects of adverse selection.

Market research will also have to be carried out to determine the level of consumer confidence in a system of this type. If confidence is high, it will be more profitable and a greater proportion of PCs can be recovered for the purpose of reuse. Research will determine the value that consumers consider to be fair, as well as finding out exactly what signals people would have the greatest appreciation of. The level of depth offered by the signals will depend on exactly what factors people consider to be important, whether they want an in-depth reading of all relevant factors or just an overview offering a simple metric of lifetime consumption.

Furthermore, with emerging technologies such as solid state drives, continued research is going to be required in order to determine the factors which influence these technologies. If the general reliability of electronics continues to increase as it has up to this point, the metric will have to be revised in order to accurately represent the current market conditions.

Finally, research needs to be carried out into information security issues raised in PC reuse. Given the number of people who store personal data on their PC, it is likely that the only way to ensure data protection would be to format the memory.

This in turn raises the issue of operating systems. If the memory of a PC is formatted, the operating system is going to have to be reinstalled, which is a costly process. This limitation will have to be overcome in order for the process to reach its full potential.

8 CONCLUSION

A process for encouraging the reuse of personal computers has been proposed. This process seeks to counteract the lack of faith in second hand PCs by offering a guarantee of reliability which is based on knowledge of the conditions under which a computer has been used.

The process also offers a low cost technological method for the separation of viable systems from nonviable ones on a large scale, eliminating the costly analysis of used systems which would be normally required in order to reclaim reusable PCs.

9 REFERENCES

- [1] Williams E. D., 2003, Extending PC lifespan through secondary markets, IEEE International Symposium on Electronics & the Environment, pp. 255-259, 382.
- [2] European Commission, 2007, Preparatory studies for Eco-design Requirements of EuPs: Lot 3, Personal Computers (Desktops and Laptops) and Computer Monitors, Final Report, Accessed from 14/2/2008

- <http://extra.ivf.se/ecocomputer/downloads/Eup%20Lot%203%20Final%20Report%20070913%20publi shed.pdf>.
- [3] Wong K., 1990, "What is wrong with the existing reliability prediction methods?" Quality & Reliability Engineering International, Vol. 6, 251-257.
- [4] Hickey, S., Fitzpatrick, C., 2007, Signalling information from embedded sensors to promote lifetime extension in electrical & electronic equipment, Journal of Physics: Conference Series, Volume 76, Issue 1.
- [5] "Rational Consumer – Small is beautiful", Economist Magazine: Technology Quarterly, December 6th 2008.
- [6] Katz J., Rifer W., Wilson A.R., 2005, EPEAT: Electronic Product Environmental Analysis Tool – development of an environmental rating system of electronic products for governmental/institutional procurement., IEEE International Symposium on Electronics & the Environment, Conference Record, 1-6.
- [7] Eco-Label, Accessed from: <http://www.eco-label.com/default.htm> on 14/12/08.
- [8] United States Environmental Protection Agency, ENERGY STAR and other Climate Protection Partnerships Annual Report 2006.
- [9] Federal Electronics Challenge – Home Page, Accessed: <http://www.federalelectronicschallenge.net/> on 14/12/2008.
- [10] BuyMyTronics, Accessed from: <http://www.buymytronics.com> on 15/12/2008.
- [11] Gazelle, Accessed at: <http://www.gazelle.com> on 15/12/2008.
- [12] "Dell Meets Carbon Neutral Goal Ahead of Schedule", Accessed from http://www.dell.com/content/topics/global.aspx/corp/pressoffice/en/2008/2008_08_06_rr_000?c=us&_en&s=corp on 15/12/2008.
- [13] Akerlof, G.A., 1970, The Market for 'Lemons': Quality Uncertainty and the Market Mechanism, The Quarterly Journal of Economics., 84(3):488-500.
- [14] "The endowment effect: It's mine, I tell you", The Economist, June 21st 2008.
- [15] Lall, P., Pecht, M., and Harkim, E., 1997, Influence of Temperature on Microelectronics and System Reliability, New York: CRC Press.
- [16] Failure Mechanisms and Models for Semiconductor Devices, JEDEC Publication JEP122-A, 2002.
- [17] Pinheiro, E. et al, 2007, Failure Trends in a Large Disk Drive Population, Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST'07).
- [18] Waste Electrical and Electronic Equipment: Accessed from http://ec.europa.eu/environment/waste/weee/index_en.htm on 15/12/2008. on 14/12/2008
- [19] One laptop per child. Accessed from <http://laptop.org>

4. International Symposium on Sustainable Systems and Technology,
Washington, May 2010

A Method for Extracting Historical Thermal Data from Used PCs to Foster Reuse

Eanna Cronin, Stewart Hickey, Colin Fitzpatrick

Abstract—In order to encourage the development of a secondary market for PCs, a system which utilises embedded sensors to perform prognostic monitoring of components has been proposed. This paper examines the technological aspects of the data recording phase of this system. The data extraction process is also discussed, and the technologies used for this are examined in depth.

Index Terms — Lifetime Extension, Radio Frequency Identification, End of Life, Waste Stream Management

I. INTRODUCTION

ENVIRONMENTALLY, it has been shown that the optimal method for reducing the impact of computers is through lifetime extension. At end of use, the method of lifetime extension with least environmental overheads is reuse[1]. But, as examined in previous research, the market conditions which allow the development of a second hand business to consumer (B2C) market

for Personal Computers (PCs) do not exist, owing in large part to a wide scale lack of consumer confidence in the structures which currently support the resale of used PCs, which are in general unregulated[2].

This has stunted the development of a secondary market for PCs, in spite of the fact that a large number of the PCs at end of use are fully functional and potentially have significant remaining value. Owners have been shown to be more likely to store a used PC for an extended period of time before disposing of it than to sell it in a secondary market. Owing to the rapid advances in electronics and computer technology, stored PCs tend to depreciate very quickly, and over the course of a short number of years may have lost any residual value which remained at the end of use.

A system for extending PC life has been proposed which takes advantage of the sensors already in place in PCs to encourage reuse. Information on the conditions under which the computer is operating is stored throughout the lifetime of the PC, and at the end of a system's use, this information may be retrieved, allowing the value remaining in the system to be estimated.

Temperature and operating time are regarded as some of the factors which can have the greatest impact on the failure of PCs. By monitoring the output of thermal sensors located in the components of a PC it is possible to estimate the depreciation which is taking place owing to thermal effects. This data can be stored in the PC and can then be extracted at end of life to allow a decision to be made on the future of the system.

Radio Frequency Identification (RFID) was selected as an ideal method for extracting the lifetime data from used PCs. This offers the possibility of extracting the lifetime data of the

³ This work was supported by the STRIVE program of the Irish Environmental Protection Agency (EPA) project 2009-PhD-ET-13.

Eanna Cronin is with the Department of Electronic and Computer Engineering, University of Limerick, Ireland (phone: 353-61-213102; e-mail: Eanna.Cronin@ul.ie).

Stewart Hickey is with the Department of Electronic and Computer Engineering, University of Limerick, Ireland (e-mail: Stewart.Hickey@ul.ie).

Colin Fitzpatrick is with the Department of Electronic and Computer Engineering, University of Limerick, Ireland (e-mail: Colin.Fitzpatrick@ul.ie).

computer without the need for the system to be powered on. This will drastically reduce the amount of time required to ascertain whether a computer is suitable for resale. Through reduction of this time and cost, the proportion of computers for which it will be viable to offer a guarantee can be increased, allowing the removal of a greater number of used systems from the waste stream.

II. COMMUNICATIONS

The central factors considered in this paper are the technologies which allow the communication of thermal data within a computer system, the storage of this data, and its extraction at the end of a PC's life. This can be broken into three separate communications systems. The first is the internal system which allows environmental data to be periodically recorded and stored in a memory device. The second is a receiver which can detect the presence of a reader circuit in the vicinity of the used PC at end of life and read RFID commands. The third communications component allows transmission of lifetime consumption data at the end of use for the system.

A. Internal Communication

The internal communications take place between temperature sensors contained in the components of a PC and a microcontroller, which stores the data. For reasons outlined below, it was decided that the microcontroller and the data storage would be located on a PCI card on the motherboard of a computer. The choice of a PCI card as the best location for the hardware associated with the project is based on the fact that the PCI bridge represents one of the longest lived components of a modern computer. It was introduced to IBM compatible PCs in 1992, and since then, despite numerous modifications to the standard, the basic compatibility of the PCI bus has remained the same, meaning that the majority of legacy PCI devices would be capable of functioning in a modern PC [3]. In an industry with such a short time before obsolescence occurs in components, it is difficult to find a protocol which remains relevant for an appreciable length of time. In this, the PCI bus is unique, and offers the opportunity to develop a prototype which is compatible with many previous generations of PCs, and which also has a low risk of obsolescence in the near future.

A second advantage of the PCI bus over other potential protocols is the fact that PCI cards declare their requirements to the host machine at start-up. The PC then allocates the resources to that PCI card, allowing efficient division of resources. This also means that a PCI card can be

removed from a PC and placed in another without the need to install support software. This allows the more wide scale testing of prototype devices, as the PCI bus is widely compatible across various system architectures.

PCI devices are located on the PCI bridge of a PC, which is on the southbridge of the motherboard, as shown in **Error! Reference source not found.** below. The southbridge controls the inputs and outputs to the processor, including the keyboard and mouse, while the northbridge controls the higher speed and memory devices on the PC.

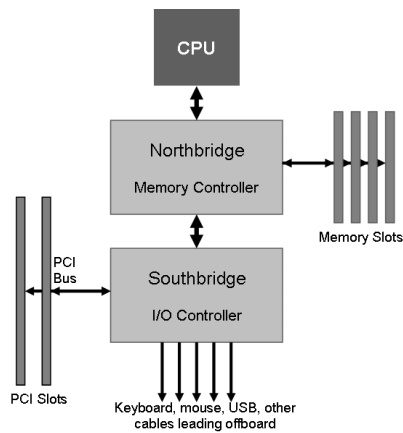


Fig. 1. Architecture of a PC motherboard. The PCI bus is located on the southbridge, or I/O Controller Hub (ICH) of the board.

Communications from the PCI bus take the form of interrupts, which can request data. These requests are carried by an internal bus to the northbridge, which is directly connected to the computer's processor.

Once thermal data has been retrieved from the processor, it is transferred to the PCI card, where any additional processing is carried out on the retrieved data using a microcontroller, as shown in **Error! Reference source not found.** This is also he stage at which rainflow cycle counting is carried out, allowing the reduction in the volume of data being stored in memory, and in the length of time required to extract the data [4].

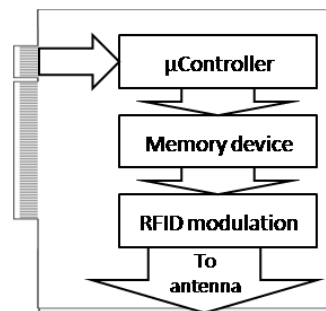


Fig. 2. Circuitry on the PCI card allowing collection, reduction,

storage and transmission of data

B. Reader to Transponder Communications

The MIFARE standard of RFID communications has been selected as the protocol for the extraction of data. The choice of this standard was based on its suitability for testing in a laboratory environment and relative ease of modification. It describes a 13.56MHz carrier, in which the energy for transponder to reader communication is provided by the reader.

This protocol is based on the ISO/IEC 14443 contactless communications standard. The transponder will be located in the PC which is being analysed, and owing to the fact that the volume of data which is to be transferred will be much greater than normally used with this protocol, it is necessary to design a transponder specially for this application, and to modify the data transmission process to handle higher data volumes.

Data transmission in RFID is achieved using inductive coupling. The reader uses a looped antenna to produce a magnetic field, which produces an electric field in a similar antenna in the transponder. The initial transponder circuit circuitry amplifies the incoming signal, removes the carrier frequency from the signal using a low pass filter, and restores it to a digital signal using a comparator. This digital signal is then inputted into a microcontroller which is capable of performing the necessary analysis on the incoming signal and responding.

Communications from the reader to the transponder take the form of simple numeric commands, which the microcontroller in the transponder will be capable of analysing and responding to in the appropriate manner. The rate of data transmission is given by Equation 1.

$$\text{Data Rate} = \frac{f_c}{128} = 105.9kb/s \tag{1}$$

In order to allow the communication of the relatively large volume of data which is stored in the memory device on the PCI card, it is necessary to implement some additional commands beyond those currently defined by the MIFARE standard.

In normal MIFARE communications, the reader transmits commands using amplitude shift keyed (ASK) encoding. This is applied in such a way as to ensure that the information can be transmitted on the carrier wave with the minimum possible amount of time where the carrier is “off”, ensuring that the transponder is kept energised for a maximum amount of time and allowing it sufficient power to respond.

Error! Reference source not found. shows the

it keying used for reader to transponder communications. An X is transmitted for a logical one, a Y is transmitted for the first logic zero in a series, and a Z is transmitted for all subsequent zeros. [5]

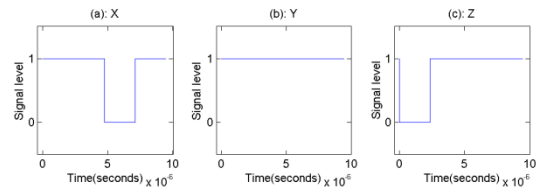


Fig. 3. Reader signal amplitude keying scheme. In (a) the carrier is keyed 'off' briefly half way through the bit period. In (b) the signal remains high throughout. In (c) the carrier is keyed 'off' at the beginning of the bit period.

For this application, the transponder needs to be powered by a battery, so that the memory device and microcontroller will have sufficient power to operate. However, for simplicity the encoding scheme is being preserved in the design of the system.

It is possible to increase the range of conditions being monitored by the system, in which case more commands might be introduced in order to extract specific aspects of a PC's lifetime data, for increasing information on viability of reuse. This would allow an initially small amount of data to be transmitted, allowing the fast division of potentially viable from non-viable systems. Following this, a larger amount of data can be communicated to allow more in-depth analysis of value.

Additional signals which could provide insight into the reuse potential of a system include the humidity of the inside of the system case, the vibration which occurs during system operation, and characteristics such as the number of read and write cycles to solid state hard drives.

C. Transponder to Reader Communications

(1) In transponder to reader communications, data is encoded differently. Load modulation is applied to the carrier wave as shown in Equation 2:

$$f_m = \frac{f_c}{16} = 847.5kHz \tag{2}$$

Data is then transmitted at a rate as given by Equation 3:

$$\text{Data Rate} = \frac{f_m}{8} = 105.9kb/s \tag{3}$$

To transmit a binary 1, the first half of the bit period has load modulation applied, and to transmit a binary 0, the second half is modulated, as shown in **Error! Reference source not found.**[5]

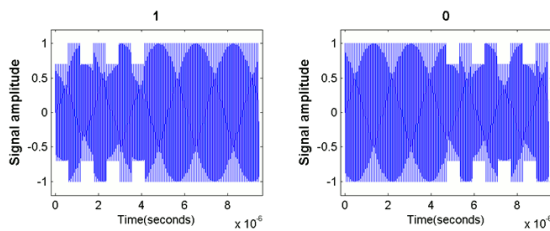


Fig. 4. Transponder signal modulation for binary one and zero. Modulation applied in the first half of the bit period for one, and in the second half for zero

On receipt of the appropriate signal from a reader, the transponder will begin to transmit all of the information which has been stored in the memory device throughout the life of the PC. As RFID is a serial interface, the information from the memory device will be passed through a parallel to serial shift register before having modulation applied. The shift register and modulation is implemented in a microcontroller. The signal is then amplified and transmitted through the antenna.

III. TEMPERATURE MONITORING AND ANALYSIS

At present, there are a variety of methods used in PCs to monitor temperature. One disadvantage of the modular nature of PCs is that regardless of the rate at which technology improves, compatibility must be maintained between components. This means that new technologies can take a long time to be adopted, particularly where the technology is replacing one which already functions. This has resulted in the situation which currently exists in PCs, where current generation systems could be using any of a variety of methods for monitoring the temperature of the CPU.

The most recent method of temperature monitoring is Intel's Platform Environment Control Interface (PECI) which is a single wire interface offering the potential to monitor the output of a variety of sensors. This has the potential to be very useful for the application of condition monitoring, since as well as monitoring temperature it is tuned to the individual processor being monitored, and is capable of monitoring multiple digital thermal sensors (DTS) on one processor. [6, 7]

While this interface has the potential to work well with condition monitoring processes, it has unfortunately not yet been adopted by a wide range of motherboard manufacturers, as the thermal diodes which are used for catastrophic failure prevention in older generation systems are still in use. As a result, motherboard manufacturers tend to use the more widely compatible standard, resulting in newer standards taking longer to be

adopted than might otherwise be the case.

A possible method of improving the rate at which new technologies are adopted is through standardisation of requirements. Schemes such as EPEAT and the EuP directive make it possible to ensure that motherboard manufacturers conform to the standard which allows the most environmentally beneficial use of the technology.[8] By standardising the requirements of motherboards to meet minimum temperature monitoring capabilities, it would greatly simplify the process of extracting thermal data from components.

One of the intended results of this research is to influence the minimum standards for PC manufacture to incorporate reuse. Owing to the rate at which new technologies are adopted in the IT sector, potential environmental benefits are often overlooked, such as with thermal monitoring hardware. However, if the benefits of additional applications of these technologies can be demonstrated, it is possible to use these applications as justification to make the adoption of technologies a minimum requirement for PC manufacturers.

A. Data Reduction: Rainflow Cycle Counting

In order to minimise the amount of data it is necessary to store, and the length of time required for data transfer, rainflow cycle counting is used. Rainflow cycle counting is used in mechanical engineering to examine the cumulative effects of stresses on building elements. It operates on the basis of a binning technique, where, rather than storing each individual temperature reading in a distinct address, the processor will analyse the cycles in temperature for repeated patterns of amplitude, mean and period. Each time a specific pattern is repeated, a memory location referenced by that pattern has its contents incremented.

This makes it possible to directly limit the maximum amount of data which will need to be transmitted, as the total data stored will always be the same as the sum of the memory in the bins.

One benefit of this is that it has the effect of standardising the length of time required to retrieve full lifetime data from a used system. As the stored data will always be the same size, the extraction process can be optimised to the exact length of time required to extract the data.

To compare the value of using rainflow cycle counting over simple linear storage of data, it is possible to examine the relative sizes of the storage space required. **Error! Reference source not found.** shows the comparative stored data over time, where it can be seen that the amount of space required for linear storage of data exceeds the

amount required by rainflow cycle counting with storage space of 142kB after 1200 hours, or about 150 days for a PC operating for 8 hours per day.

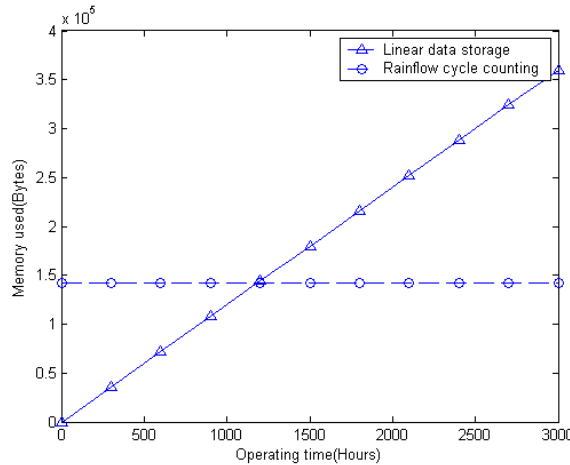


Fig. 5. Comparative data sizes for reduced data compared to linear storage.

IV. SECURITY CONCERNS

As with any system which can be used to monitor the actions of people, the issue of privacy is one which must be examined in depth. As well as concerns over the security of the data on extraction at end of use, it is possible that users will have issues with the possibility of data being extracted from their computers before end of use.

It is difficult to imagine a use for the data which could be extracted in this way, as the information would be useless to most people. The availability of thermal data about a computer is comparable to the visibility of a car's odometer through the window, however it is necessary to protect personal information as much as possible.

The MIFARE RFID standard includes the possibility of simple data encoding to prevent access by unauthorised readers. The native security supported by the MIFARE classic card uses a key which is only 48 bits long, making it relatively easy to crack, but by using the microcontroller, it is possible to implement more secure methods of data transmission if necessary. However, this would significantly increase the length of time which would be required to transfer data.

Standards other than MIFARE offer some additional security benefits over encryption. Some technologies will allow read and destroy, where the data on an RFID card is deleted as it is read, preventing malicious access. Through adoption of this and other security features, it is possible to remove the concerns over data security which are inherent in wireless communications.

V. CONCLUSION

The technical aspects of the proposed system to

promote lifetime extension have been examined in this paper. This system could be used for the purpose of strengthening the market for used PCs. In order to develop the system to the point where it operates satisfactorily, it is necessary to perform additional research. The end of life process must be examined to find the stage in the reverse logistical cycle at which used PCs can be analysed for reusability. Research must also be carried out into the markets for PCs, in order to determine the value which can be assigned to a used PC after a number of years, regardless of time to failure.

Error! Reference source not found. shows some potential end of use destinations for PCs, along with one possible decision making process. This could be further developed to allow faster initial analysis and division of systems, and increased breakdown of the value of used systems. Alternatively, in a free market scenario, the information retrieved in this way could be directly uploaded to an online database or auction site, allowing the direct determination of value by market forces, and removing a portion of the analysis phase.

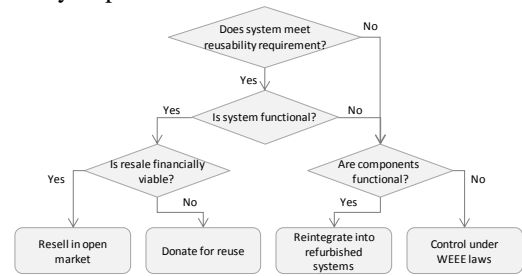


Fig. 6. Potential end of use destinations of PCs and a possible decision tree

REFERENCES

- [1] E. D. Williams and Y. Sasaki, "Energy analysis of end-of-life options for personal computers: Resell, upgrade, recycle," *2003 IEEE International Symposium on Electronics & the Environment, Conference Record*, pp. 187-192, 382, 2003.
- [2] S. W. Hickey and C. Fitzpatrick, "Combating adverse selection in secondary PC markets," *Environmental Science & Technology*, vol. 42, pp. 3047-3052, Apr 15 2008.
- [3] B. K. C. Kluth. (2001, April 2001) PCI Bus Prototyping Card (2). *Elektor Electronics*.
- [4] E. Cronin, P. Sweeney, S. Hickey, and C. Fitzpatrick, "Developing RFID signalling to close the loop on second hand computers," in *Sustainable Systems and Technology, 2009. ISSST '09. IEEE International Symposium on*, 2009, pp. 1-6.
- [5] I. O. f. S. I. E. Commission, "ISO/IEC 14443-2 Identification cards - Contactless integrated circuit(s) cards - Proximity cards," in *Part 2: Radio frequency power and signal interface*, ed. 1999.
- [6] "Dual-Core Intel® Xeon® Processor 5100 Series Datasheet," August 2007.
- [7] M. Berktold, "CPU Monitoring With DTS/PECI," September 2009 2009.
- [8] "The Energy Using Products (EuP) Directive, Version 7," ed.

5. CARE Innovation, Vienna, November 2010

TECHNICAL CONSIDERATIONS IN A SYSTEM FOR LIFETIME DATA EXTRACTION IN PERSONAL COMPUTERS*Éanna Cronin, Colin Fitzpatrick*Department of Electronic and Computer Engineering
University of Limerick

Abstract: This paper considers the technical aspects of a system which allows the swift analysis of reuse potential at end of life for PCs. The system utilises thermal sensors already present in the components of a PC to track the health of the PC throughout its lifetime. At end of life, this data can be extracted using Radio Frequency Identification (RFID) to allow a fast first diagnosis of the reuse potential of the system. The paper examines the choice of RFID as a suitable technology, the requirements of the MIFARE RFID standard, and the modifications made to the standard to allow the transmission of the volumes of data required for lifetime condition monitoring. Also examined are the method of data reduction in use for thermal data, and the location of the system on the PCI bus of the PC. The advantages offered to a system of this type by its inclusion in eco-labels are also considered.

INTRODUCTION

Consumers are becoming more aware of the environmental effects of their purchases, and it has been shown that reducing a product's negative environmental effects has a positive impact on the market for a product [1]. Eco-labels such as EPEAT and Energy Star help to encourage this effect by giving consumers a simple indicator of environmental performance covering all stages in the life cycle[2, 3].

For personal computers, however, the end of life performance of a product can be difficult to ascertain. Manufacturers can reduce the number of hazardous materials used in production, and design products for easy disassembly, but these efforts do not help to encourage reuse of products, which has been identified as the optimal end of life process for some PCs[4, 5].

One reason for the difficulty in monitoring the end of life in PCs is the massive variety in the quality and age of systems at end of life. In order to determine the optimal end of life destination for any individual system, it is necessary to ascertain not only whether or not it is functional, but also the environmental performance of the system, and the length of time for which its life may be expected to be extended through reuse.

While many PCs are routinely replaced owing to functional failure or obsolescence, the occurrence of physical failure is much rarer, meaning that a large number of physically

functional PCs are disposed of. The value of an end of life PC should be determined not only by the age and features of the system, but also by the expected remaining functional life.

Unfortunately, it is difficult to separate functional PCs which are viable for reuse from those which are not, and as a result a large number of functional PCs are handled as waste, and conversely a large number of non-functional PCs are exported from developed countries "for reuse" in contravention of the convention on the export of hazardous waste[6].

1 THERMAL LIFECYCLE MONITORING

It has been shown that the temperature at which a PC operates has a major impact on the reliability of its components at end of life[7]. A PC operating in a well ventilated, cool environment will have a significantly different end of life prognosis to one which has been operating in a hot environment, or one which is not sufficiently cooled, leading to large temperature changes.

By monitoring the temperature of key computer components throughout the lifecycle of a PC, it is possible to more accurately determine the potential of the system for reuse. It has been shown that PCs which support an intensive workload will have higher temperatures, on average, than those used for less processor-intensive applications [7].

All CPUs feature thermal diodes, which are used for prognostic monitoring of the condition of

the die. In situations where the processor reaches a temperature where it is likely to be catastrophically damaged, the processor is capable of preventing the damage by reducing the clock speed of the CPU (choking) or shutting down the computer altogether.

These diodes are also visible through software and hardware. PC users such as gamers who require very fast operation of their PCs will often monitor the temperature of the CPU as an indicator of the health of the system.

By accessing the thermal data from these sensors and recording it over the lifetime of a PC, it is possible to develop an image of the thermal performance which the system has demonstrated throughout its lifetime, and thereby estimate the overall health of the system.

2 RFID COMMUNICATIONS

While the thermal history of a PC may be used to estimate the remaining life of the system, it is first necessary for this information to be accessible at end of life. Ideally, extraction of data should be possible for someone with little or no technical skills. By minimising the cost associated with quantifying the reusability of a PC, it is possible to ensure that a greater number of PCs are reused, as the cost of analysing a computer for reuse must be offset against the profit generated through its reuse, or added to the total end of life cost of handling the systems as waste.

Ideally, a system for determining reusability is capable of extracting the necessary data from a PC while the PC is powered off. This allows triage to take place, where systems can be quickly analysed at end of life and their ultimate destination decided quickly. Radio Frequency Identification (RFID) offers the capability of wireless communications where the power for both sides of the communication is provided by the reader device. This is an ideal method for extracting information from end of life PCs, as the machine under test does not have to be powered on or mechanically coupled to the reader circuit for data transfer to take place, allowing quick and seamless examination of a large volume of PCs in a short time.

RFID allows communications between a “reader”, which provides the energy for communications, and a “tag”, which is powered by an inductively coupled circuit to respond to requests for information. Both parts feature an antenna to transmit and receive data, which is tuned to a specific frequency of operation depending on the type of standard in use.

3.1 Customisation

RFID is normally used for the transmission of relatively small amounts of data. As such, standard RFID equipment is not designed to allow the transmission of the volumes of data required for the operation of a lifetime consumption analysis system. For this reason it is necessary to modify the operation of an RFID reader, and to design a tag for the transmission of significantly larger amounts of data, while continuing to operate within the defined requirements of the standard.

An RFID reader was purchased which operates using the MIFARE RFID standard, primarily used for smart identification cards. The frequency of operation for this standard is 13.56MHz. This particular reader was selected as the most appropriate candidate owing to the fact that it features a processor which is easily modified by the user. Under normal operation, it can transmit instructions to a tag and read a reply of up to 16 bytes, analysing and relaying the reply over a USB link to a PC.

Taking advantage of the demodulation hardware which is already in place on the reader circuit, it becomes a relatively trivial task to modify the function of the device. It is possible to introduce new commands in addition to the ones defined by the MIFARE standard. These could include various levels of information retrieval, to allow a greater or lesser level of triage, depending on the requirements of the user. The exact capabilities of the reader will be determined through collaboration with manufacturers and refurbishers, and will reflect their requirements for the system.

The USB link on the board can be used for reprogramming the device to meet the requirements of the prototype, and also to relay information retrieved from the tag circuit to a PC for processing and analysis.

3.2 The MIFARE standard

The RFID standard used by the reader defines specific requirements for transmitting and receiving data. In order to best utilise the capabilities of the reader, the specifications of the standards are followed as closely as possible in the design of the tag circuit and the modification of the reader.

In communications from the reader to tags, data is modulated in such a way as to maintain a continuous flow of energy as much as possible during communication, as tags rely on energy provided by readers in order to send a response. As such, the modulation of each bit requires that a maximum of one quarter of the bit period has the carrier frequency keyed off.

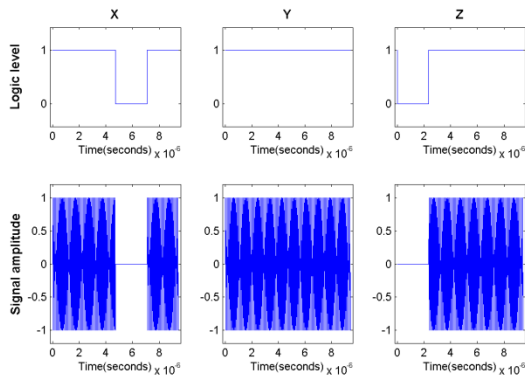


Figure 1: Bit coding in reader-tag communication

In the case of the transmission of a logic ‘1’, the reader will transmit an ‘X’ signal, which keys the carrier off for one quarter bit period starting half way through the bit period. For the first logic ‘0’ in a sequence, a ‘Y’ signal is sent, which does not key the carrier off at all, and for all subsequent ‘0’s a Z is sent, which keys the carrier off for a quarter bit period starting at the beginning of the bit period. The relevant waveforms are shown in Figure 1. The tag circuitry must be capable of receiving these commands, demodulating them into their original ‘0’s and ‘1’s, and responding correctly in the manner described in the MIFARE standard.

As well as this, the tag circuit must be capable of performing its own modulation for the transmission of data. The standard describes a different form of modulation for tag to reader communications, which must be observed when responding to reader commands. For these communications, it is necessary to apply partial amplitude modulation to the carrier. The carrier frequency is modulated with a modulation frequency of:

$$f_m = \frac{f_c}{16} = \frac{13.56\text{MHz}}{16} = 847.5\text{kHz}$$

For the transmission of a logical ‘1’, this modulation is applied to the first half of the bit period and for transmission of a ‘0’ modulation is applied to the second half. The partial amplitude modulation means that the wave which is output is reduced in amplitude to approximately 80% of its maximum value where the modulation is applied. The resultant waveforms for transmission of a 1 and a 0 are shown in Figure 2. In demodulating the wave, the modulation frequency is isolated by the reader using a tuned circuit allowing the original message to be retrieved.

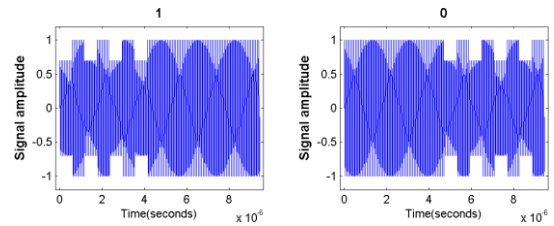


Figure 2: Modulation in tag-reader communication

3.3 Demodulation circuitry

In order to convert the analogue signal received by the antenna on the tag into a digital signal which can be decoded by the microcontroller, it was necessary

to implement some circuitry removing the analogue carrier wave.

The signal received will be a 13.56MHz wave, with gaps where modulation has been applied by the reader. This is received by a tuned circuit and amplified using a transistor in a common emitter amplifier configuration. The amplified signal is then passed through a low pass filter, which removes the high frequency components, but also affects the definition of the transitions between high and low logic levels. These are then restored by passing the signal through a comparator, which results in a digital signal with the required definition for the microcontroller.

3.4 Modulation circuitry

For the transmission of data from the tag, the microcontroller performs the majority of the modulation. The output of the microcontroller is a ‘0’ or a ‘1’ depending on whether or not modulation is applied. It is necessary to take this signal and change it into the required signal for transmission, which involves increasing the ‘0’ level from 0V to approximately 80% of the maximum amplitude and applying the carrier frequency to the entire signal.

This digital signal can then be used as the trigger for an analogue multiplexer, which has as its two inputs the carrier wave at the two different amplitudes. The output of the multiplexer will be amplified in order to give the signal sufficient power before it is transmitted through the antenna.

3.5 Receiving and demodulation

Due to the very specific requirements for the modulation and rate of data transfer as laid out in the RFID standard, it is necessary for the transmission of the modulated signal to be carefully controlled so that communications between the reader and the tag may be correctly interpreted. For this reason, the portion of the code governing the communications between the reader and tag is written in assembly language, which

allows the total number of clocks required for each command to be observed. This enables the programmer to count the number of clocks which elapse between samples, and ensure that data is transmitted exactly as outlined in the standard.

As this is low-level code, it offers the advantage of enabling relatively simple functional modifications to the operation of the tag. The code governing the transmission and receipt of data can be modified more or less independently of the higher level code without affecting the operation of the circuit, as long as the format in which the assembly code section receives and returns data remains unchanged. This allows the adaptation of the hardware to assess the advantages offered by various RFID standards.

The assembly code will sample continuously while there is a signal from the reader. Sampling is performed at a rate of eight samples per bit, as the modulation period is approximately 2.36 μs , or a quarter of a bit, requiring a sampling period of 1.18 μs to guarantee reconstruction.

Under the MIFARE standard, the transmission of a command from the reader will always be preceded by a Z signal, which is initiated by keying the signal low for one quarter bit period. This means that the tag can listen while the signal is active, but discard the initial string of '1's received before the first '0'. This can then be used as the signal which allows the synchronisation of the remainder of the message within the tag.

The majority of communications sent from the reader to the tag take the form of 7 bit commands, preceded by the Z symbol and followed by a '0'. This means that it is unnecessary to implement data storage or buffering on the tag side of communications, as the command can be decoded as soon as 8 data bits have been received.

Owing to the keying in use, it is easy to detect the occurrence of an error in the transmission of data, as the signal must conform to predefined patterns. For example, if any more than seven consecutive quarter bits are detected as high, it can be assumed that the transmission is ended, as there is no combination of input bits that will generate this modulation pattern. Conversely, more than one consecutive "low" quarter bit will indicate that the reader has moved out of range.

3.6 Modulation and data transmission

For transmission of data, the MIFARE standard defines some error checking processes which make it easier to verify the integrity of the data transmitted by the tag. Owing to the fact that the keying performed by the tag is less complex than that performed by the reader, it is also necessary to implement data checks to verify that the demodulated data is the signal as transmitted by the tag.

The method of data verification in use by the standard is 16 bit cyclic redundancy checking (CRC). The tag must apply the check bits to the data before transmitting, and the reader must be capable of demodulating the signal and checking the correct check bits have been used.

3 DATA REDUCTION

In order for this data extraction method to be effective, it is necessary for it to be both fast and accurate. Since high frequency RFID offers a maximum data rate of about 106 kbit/s, it is necessary to ensure that the maximum amount of data is limited to minimise transfer time, especially considering the fact that the system will be expected to deal with machines which have been operating over several years and that many PCs are rarely, if ever, powered off. If thermal data recording is performed in a continuous and linear way, the space requirements for data storage also increase linearly.

As the amount of data to be transferred increases, so does the probability that an error will occur in the data transfer. Although the cyclic redundancy checking will ensure that data is not incorrectly interpreted, it is obviously desirable to minimise errors in order to reduce the total length of time required for the transfer of the complete set of data.

While it is therefore necessary to reduce the amount of data to be stored, it is highly undesirable to simply reduce the resolution of the temperature data samples, as this is likely to result in the loss of valuable thermal data. For this reason, methods of reducing the volume of data being stored without affecting the integrity of the data itself were examined.

4.1 Rainflow cycle counting

While the absolute temperature can have a serious effect on the reliability of electronic components, temperature change over time is also a factor, as changing temperature has the effect of causing the expansion and contraction of metallic elements, and wherever two different materials are connected their different rates of expansion can cause structural damage. For this reason it is important to monitor the temperature cycles that a PC goes through.

Rainflow cycle counting operates by monitoring the temperature of the PC and, instead of recording the temperature directly, counting similar cycles and grouping them together in bins. With a bin identifying the mean temperature, amplitude and period of the temperature change, it is possible for each possible temperature cycle to be accounted for. The bins in use are of a predefined size which does not change regardless

of the amount of data being collected. This means that it is possible to accurately predict the exact length of time for all of the recorded data to be transmitted from a tag to a reader.

4 PCI CARD

It is important for the system to be easily incorporated into a wide variety of computer architectures. Within a PC, there are a variety of possible configurations of hardware, owing to the number of different manufacturers and possible applications of PCs. However, the PCI bus is a standard feature of the vast majority of PCs, and has proven to be very resistant to the changes that computer architectures are constantly undergoing.

While the PCI card has changed somewhat since it was introduced, compatibility has been maintained between the current standard and previous ones. This allows a prototype developed on PCI card to be tested in a very wide range of PCs, and also suggests that the PCI bus is likely to be similarly robust to future change, and that it will remain a feature of PCs for the foreseeable future.

For these reasons, the circuitry for the storage of thermal data, and reduction and modulation of the data is based on a PCI card. The card interfaces with the PCI bus of a PC, which communicates with the processor via the southbridge of the motherboard. A further advantage of the PCI card as a prototype development platform is that one end of the card is external to the PC, allowing the location of the RFID antenna on the outside of the case, preventing magnetic interference from the metal case.

5 CONCLUSION

The primary goal of developing this technology is influencing manufacturers to include it in the design of new Personal Computers. By incorporating the technology as a standard feature, it will be possible for the handling of PCs at end of life to be controlled based largely on the reusability of the individual PC, rather than using the age of the PC as the only metric.

By introducing the ability to quickly and accurately determine the remaining value in a PC, it becomes viable to separate PCs depending on their optimal end of life destinations. The reduction in the cost of assessing the reusability introduces the potential for a greater number of computers to be reused.

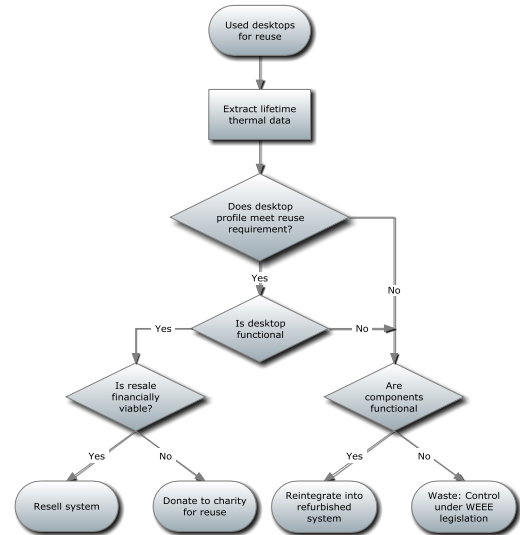


Figure 3: End of life flow diagram for desktops

As well as increasing the number of computers which are reused, since the system could optimise the end of life performance of PCs in which it is implemented, the addition of this system in the manufacture phase of a PC could become a requirement of eco-labels. This would allow the eco-labels to address the difficult end-of-life phase of a computer's life cycle.

One advantage of the inclusion of a system of this type in eco-labelling schemes is that, with the standardisation of requirements for reuse to include thermal monitoring, minimum thresholds can be introduced in order for electronics to be defined as suitable for reuse. This will have the effect of effectively labelling all sub-standard electronics as waste, thereby preventing their illegal export. One possible decision tree for end of life desktops is given in Figure 3.

6 REFERENCES

- [1] J.-D. M. Saphores, H. Nixon, O. A. Ogunseitani, and A. A. Shapiro, "California households' willingness to pay for 'green' electronics," *Journal of Environmental Planning and Management*, vol. 50, pp. 113 - 133, 2007.
- [2] "Energy Star Program Requirements for Computers, Version 5.0," 2008.
- [3] "Federal Acquisition Regulation; FAR Case 2006-030, Electronic Products Environmental Assessment Tool (EPEAT)," 2009.
- [4] S. Sahni, A. Boustani, T. Gutowski, and S. Graves, "Reusing Personal Computers - Good or Bad for the Environment?," in *Sustainable Systems and Technology, 2010. ISSST '10. IEEE International Symposium on*, 2010.
- [5] E. D. Williams and Y. Sasaki, "Energy analysis of end-of-life options for personal computers: Resell, upgrade, recycle," *2003 Ieee International Symposium on Electronics & the Environment, Conference Record*, pp. 187-192, 382, 2003.
- [6] J. Puckett, S. Westervelt, R. Gutierrez, and Y. Takamiya, "The Digital Dump - Exporting Re-use and Abuse to Africa," ed, 2005.

- [7] S. Hickey and C. Fitzpatrick, "Tackling adverse selection in secondary PC markets," *Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses*, pp. 347-352, 480, 2007.