Can a combination of software and hardware be used to breach RFID-enabled doors without the use of predetermined access code?

Author
Dominic Bonaker

Supervisor
Dr Chaminda Hewage

April 2017

In partial fulfillment for the degree of
BSc (Hons.) Software Engineering
# Page of contents

Title page.................................................................................................................1  
Page of contents.......................................................................................................2  
1.0 Abstract..............................................................................................................3  
2.0 Acknowledgment.................................................................................................3  
3.0 Introduction.........................................................................................................4  
4.0 Issues with current implementation....................................................................4  
5.0 Current implemented systems.........................................................................6  
6.0 Aims and objectives.........................................................................................7  
7.0 Literature review...............................................................................................8  
    7.1 Use of RFID within the medical industry supply chain.............................8  
    7.2 RFID Technologies and Application.........................................................9  
8.0 Methodology....................................................................................................11  
9.0 Project planning...............................................................................................11  
10.0 Requirements and analysis...........................................................................12  
    10.1 Brute force attacks..................................................................................12  
    10.2 Iterative methods....................................................................................13  
    10.3 Rainbow tables......................................................................................13  
    10.4 Encryption..............................................................................................13  
    10.5 Arduino..................................................................................................15  
    10.6 MFRC522..............................................................................................15  
11.0 Questionnaire.................................................................................................16  
12.0 Design............................................................................................................18  
13.0 Implementation testing...................................................................................19  
14.0 Code................................................................................................................21  
15.0 Limitation........................................................................................................34  
16.0 Results and discussion....................................................................................35  
17.0 Application.......................................................................................................36  
18.0 Conclusion.......................................................................................................37  
19.0 References.......................................................................................................37  
20.0 Glossary...........................................................................................................38  
21.0 Appendices.....................................................................................................39  
    21.1 Calculation..............................................................................................39  
    21.2 Code Libraries.........................................................................................39  
    21.3 Questionnaire Results............................................................................67  

1.0 Abstract

RFID (Radio Frequency IDentification) is a technology that is rapidly increasing its exposure to the mass market both in the public and private sector. This rapid growth of understanding and implementation has lead to various issues being overlooked due to the convenient nature of the technology. A specifically overlooked area has been the security aspect of this technology. There are still significant flaws that have yet to be addressed by both the manufacturers of the components and the organisations that are implementing this technology into everyday life. The current awareness of this technology and the risk thereby associated with this technology are generally unknown by the users. Organisations are typically reluctant to disclose such information in fear of potential security breaches to the systems. Regardless, there are currently known ways to breach such systems, however, in some circumstances, these require expensive technology and extensive knowledge of the technology. As the technology is relatively new it can be said that these issues will be resolved in due course over the coming years. Alarmingly, these systems are currently being implemented into our everyday lives without the awareness of these issues. RFID is making ways in the financial sector in form of contactless payment cards, public transport, in the form of contactless passes and within buildings, to safeguard important documentation or geographical location from unauthorised access. This study takes a different approach to breaching these RFID-enabled systems by using an iterative method of brute force attacking to gain area access without stating an access code prior. The research hereby conducted concludes that it possible to preform a brute force attack on RFID transponders using an iterative method in theory, however, there are limiting factor in the brute force process such as interference of the systems and duration of the attack. This success does indicate the current flaws within currently implemented systems and identifies that there is an opportunity for further breaches of a similar nature.

2.0 Acknowledgments

I would like to thank all the people who have contributed in any way to the work described in this dissertation. First and foremost, I would like to thank Dr Chaminda Hewage for his constant support throughout this process and his creative ideas that have helped influence this project. During this project, he has facilitated me to create a project this is completely out of my comfort zone and helped me to produce a prototype with various areas for expansion. I would like to thank Dr Ambikesh Jayal for encouraging me to further my studies and to continue to develop my love of technology and everything that it influences. He has helped me make decisions for plans that will fundamentally affect my future. I greatly appreciate the help and advice I received from Catherine Tryfona alongside Dr Paul Angel and Nigel Jones during the more stressful stints of my studies. Their insights and support have helped me tremendously and for that, I cannot thank them enough. I would like to thank Stuart McNeil for his encouragement and support over the many years in which he has lectured me. His approach to lecturing and student care has inspired me to potentially one day become a lecturer myself. Stuart was always available to discuss ideas that I had and would give them his honest and candid opinion regardless of any emotional attachment I may have had. I would also like to thank the Centre of entrepreneurship at Cardiff Metropolitan University. They have taught me valuable skills that I deem essential to becoming a rounded and skilled individual. Finally, I would like to acknowledge friends and family who supported me during my study. I would especially like to thank my Mum and Dad for their support, not only financially, but
emotionally also, without them; I would not be the person I am today.

3.0 Introduction

RFID is a technology that has been present in current technologies for some time now. The concept of RFID engineering was first introduced in 1864 when James C Maxwell predicted the existence of electromagnetic waves. These electromagnetic waves, of which microwaves are part, which lead Maxwell for the prediction of existence of these electromagnetic waves. Following this, Heinrich Hertz demonstrated the existence of these electromagnetic waves by constructing an apparatus involving Leyden Jars and a wrought iron point spark. This experiment conducted by Hertz fundamentally proved the existence of these waves, confirming the theories proposed by Maxwell in 1864. The first known application of RFID is said to be alongside radar technology in the height of the Second World War. RF transponders (tags) were implemented into friendly planes and interrogators (readers) were placed in radar towers creating systems called the IFF (Identification of Friend of Foe). This system enabled the radar operators to determine if the planes that were approaching were friendly or that of the enemy. This detection processes became a valuable tool especially in darker periods of the day where visibility was impaired. This development of RFID alongside radar has been said that it “saved lives during the Blitz bombardments” (Miles, 2008)

The development of the technology made it easier for the commercial markets to start using it within their industries. One of the first applications within the commercial sector was when RFID tags were deployed in retail stored in the 1970s to prevent theft of goods. This process worked by having tags embedded on labels or applied within boxes to which an alarm would be triggered if these tags were not deactivated prior to the item leaving the store. The employee would be required to manoeuvre the tag over the reader in which deactivated it at the time of purchase. This would ensure that the tag did not trigger the alarm upon leaving the store.

In more current situations this technology is currently being used in similar ways in some of the largest organisation in the world, including such companies as Wal-Mart, Target and Carrefour. These companies are all starting to use this technology not only for theft prevention but also for purposes of, stock rotation and management, logistics and advertising. These applications are constantly changing as the technology and knowledge around the area becomes more understood and more implementation applications are created.

It is clear that this technology’s roots are deep within history and has advanced considerable in this timeframe. However, as with most technology, advancement is subject to time, and with development in other areas, this can have a fundamental effect on the growth and understanding of this technology.

4.0 Issues with current implementations

Within the RFID sector, unfortunately, there are already some known issues with the technology. These issues are the fundamental reasons why this technology isn’t advancing at a rate of its potential. Regardless of these issues being known, this does not help solve them, as they are issues that come with the technology. The essential mechanics of the systems are often the cause of the issue in organisational settings.

One of the most prominent issues with RFID is the interference between systems. This is when two or more systems, all communicating at the same wavelength, come into close proximity. The electromagnetic waves become distorted when communicated near another device on the similar wavelength and become unreadable by the interrogator as a result. This
issue becomes more and more apparent when an organisation starts to implement this technology for different applications. An organisation could have area management systems in place using access cards, the same company could also have some form of logistical tracking using a RFID method and also another system for stock management, all on the same wavelength. If these tags or readers were to come into a close proximity whilst trying to read the information on one of the tags there could be a collision of electromagnetic waves and invalid reading.

To mitigate such an issue it is sometimes recommended that the organisation try to vary its wavelengths for different implementations. This would then allow the systems to work on a dedicated wavelength that would be specific to that implementation. Furthermore, careful consideration should be made in the placement of these systems to allow the greatest amount of space between them to avoid this cross communication of waves.

Another issue that RFID is currently facing is the standardisation of systems across the world. As the technology is so widely applicable to many situations in various countries this has lead to a diverse set of standards from country to country. While having a technology that can be diversified in many different aspects and can be applied to various circumstances can be invaluable, this has lead to inconsistencies among the standards of this technology. Some countries have set guidelines in what they deem as acceptable uses of the technology only allowing certain wavelengths. Moreover, some countries have reserved specific wavelengths for military or broadcasting purposes. As certain countries are the more prolific at their production of this technology, such as China, the types of tags and readers can often be predetermined to function under the Chinese standards. This then becomes detrimental as in situations where products using the RFID technology are not correctly being read from or written to, as there is a breakdown in communication due to mismatched wavelengths. This being said, it is critical that organisation consider the use of this technology within their products and what types of standards the recipient may have, in order to fully maximise the potential. “To reach its full potential, RFID must be used throughout the supply chain.” (Hardgrave et al., 2008)

Furthermore, another issue that is currently slowing the advancement of RFID is the reliability aspect of the technology. This reliability becomes more of an issue when used with the supply chain for applications such as stock management and product tracking. The issue of reliability can become an issue when shipping certain items. For example when tags are applied to certain materials such as metal surfaces, this can sometimes affect the read reliability of a tag. As a result this presents the engineers deploying the systems, the critical objective for the optimal tag placement. This placement will allow the tag to be read every time it is passed by a reader but also not to expose the tag in such a way that it could be damaged during transport or tampered with, which could effect integrity of the information on the tag. “Extensive testing is the key to successful deployments” (Menges, 2006). Testing the systems in situ will allow the organisations to test the current implementation of the system allowing them to make any required changes to enhance the performance of the implementation.

Finally, the issue that could be deemed as the biggest limiting factor to this technology is the security aspect of it. Security within the technology has been a huge area of contention especially when the technology began being implemented in contactless payment cards. A known issue has been that scammers have used card readers concealed in bags are scanning the cards through pockets of the victims. This fundamental security flaw could be a reason for the lack of development within this area. Technology providers are reluctant to release information associated with issues related to their technology as they feel it will either spark outrage among consumers resulting in a decline of use or will inform and encourage other scammers to replicate the scam. There are already known methods of stealing information
from these cards whether they are of the contactless payment variety or cards that grant access to buildings. As the technology advances more of security flaws will be addressed, however, if the consumer does not update the systems when these security patches are released then they are still susceptible to these attack/scams.

It is apparent there are still various issues associated with this technology however as with most technology these issues get resolved over time as the technology advances. These issues, however, are not stopping the technology from being implemented, yet it should still be considered the sustainability of growth and application that could be affected if these issues are not resolved.

5.0 Current implemented systems

As discussed many organisations are currently using RFID in everyday situations. Some of the largest companies are implementing these systems to help boost performance or minimise waste through their processes. Companies are making the investment in the RFID technology despite some areas of uncertainty within the technology. The application for RFID within organisation and consumer situations is endless and the limitations of these are likely to be resolved as the technology advances over the coming years.

An area in which RFID technology is being used frequently is the supply chain. The supply chain is the provision of goods to retailers or manufacturers. The supply chain is using this technology to track products throughout transit, monitor levels of stock, and in some circumstances, to measure the quality of good such as temperature sensitive items. When an item is shipped it will start the supply chain, this is when tags containing information such as; quantity, location, UID (Unique IDentification) number and item information are applied to the item being shipped. This can be useful when trying to determine the content of the shipment without opening the item that could be environmentally sensitive. The shipment would then be transported to its next location where the item would pass a scanner, this would confirm that the item had to be received by the recipient. If the items within the shipment are individually tagged then this could provide insight to both the sender and recipient if the shipment has been tampered with or any item within the shipment had been lost during transit. “In reality the preliminary uses of RFID in the supply chain are merely the tip of the proverbial iceberg” (Miller et al., 2008)

The retail industry has also adopted this technology rapidly. This industry is using RFID to not only enhance supply chain functionality but is being used for monitoring sales of certain products and in advertising and marketing strategies. The biggest implementation within this sector has ben for the purpose of prevention of theft of goods. Previously retailers would have to invest in bulky “collars” to apply to products or pins that can potential damage the item upon insertion or removal. Now, this process of applying these tags to the product can be as simple as applying adhesive stickers.

However, RFID is no only present within retail environments. Many institutions ranging from government buildings to universities are using RFID to monitor and grant access to specific areas of the facility. Furthermore, in these environments RFID is often being used to monitor attendance and measure time spent on certain facilities. RFID access cards are becoming more commonplace within larger organisations where area access may be restricted in some way. For example, certain gymnasiuims or leisure centres use RFID access cards to allow members to access the facility. If a person is not a member and does not have an access card then they would not be allowed to access the facilities.

The financial industry has also been a huge adopter of the technology by implementing the technology in contactless payment cards. This allows payments to be made without inserting
the card into any reader and does not require a PIN number to be entered. This implementation of RFID within bank cards has been a huge success for the financial industry as the technology is being seen in more and more of the cards across various countries and being supplied by various banks and building societies. Contactless payments have given a level of convenience to the consumer allowing for swift payment transactions. This has also benefited businesses as less time is spent processing card payments at the time of checkout. These contactless payment methods have started to become more present in smartphones as of recent. Most of the major smartphone providers such as Apple, Samsung and Google are starting to implement this technology in their latest models of smartphones. This technology is enabling people to pay with their phones in a similar fashion to how a contactless card payment would work. The phone is held to a card reader and then the device will ask for some form of verification whether that be of the biometric kind or in the form of a passcode. As a security measure, there have been certain caps on spending in a single transaction. These caps vary from country to country and are constantly being changed to fit the demand of the consumers. At present the United Kingdom has a £30 limit per transaction whereas Taiwan does not have a limit on spending, however in certain circumstances where transactions are of a higher value signatures may be required as a method of verification. It is clear that there are currently large sectors using this technology in the everyday application regardless of the risks stated prior. In the eyes of the organisation and consumers using the technology, the benefits of the technology outweigh the risks associated with it. These current implementations are showing huge promise for the development of this technology as the usage goes up more research and development will be done within this area, improving the reliably, security and range of application in which it can be applied.

6.0 Aims and objectives

The aim of this project is to produce a piece of software and/or hardware that can be used to breach an RFID-enabled door without using a predetermined access code. This prototype will use an iterative method of brute force attack starting with 0 values in all of the relevant sectors and add 1 to the sector and test this code against the door. Then this will loop until all the potential codes have been tested. For this project, the main objective is to determine whether an iterative method of writing to RFID tags is possible. This will be fundamental to the project as the iterative method will be the chosen method for the breach of RFID-enabled doors. If the iterative method of writing to the tag is deemed possible then the next objective would be to determine if the writer can be used to breach an RFID-enabled door. Secondly, following the objectives stated prior, the next objective would be to derive a timeframe for the entire process to be completed. This would then be used to determine if the process is feasible for this application and if the process can be completed within a reasonable timeframe. A tertiary objective would be to discover any other risks or security issues with this technology throughout this research. This could be in various forms either logical or physical. However, the majority of the focus will be on the primary and secondary objectives to identify this underpinning security flaw.

- To produce a piece of software and/or hardware that can be used to breach a RFID enabled door without using a predetermined access code.
- Determine whether an iterative method of writing to RFID tags is possible.
- Determine if the writer can be used to breach an RFID-enabled door.
- Formulate a timeframe for the entire process to be completed.
The aims and objective of this project are clear and the focus is on the design and development of a prototype to prove a theory of iterative writing to a RFID tag. This tag should then be able to be used to unlock an RFID-enabled door. As both software and hardware are being developed congruently it is essential that both these elements work together without any issues that could jeopardise the functionality of the prototype.

7.0 Literature Review

The purpose of this literature review is to identify current implementations of RFID within varying industries. Moreover, specific uses will be identified to how the companies are using RFID and the results they are experiencing. Careful consideration will be made to the issues related to RFID and the processes that are currently being implemented to mitigate these issues. The biggest issues currently facing the use of RFID are primary the security element of RFID as these systems are being used to safeguard information that can be of high sensitivity and/or high value. Another issue that RFID systems are currently facing is implementation. Implementing RFID systems can be a problematic process, as most systems require various elements to become functional within their environment.

7.1 Use of RFID within the medical industry supply chain.

Until recently, supplies for medical experimentation have always been difficult to make “readily available for whenever they are needed but still kept secure with their purchase recorded and tracked remotely”(Collins, 2005). Recently more and more laboratories have been increasing their use of RFID to maintain a constant stream of medical and chemical supplies. “More than 100 laboratories around the world are using the PromegaExpress RFID system”(Collins, 2005). This indicated that within the same year of development the system was being implemented worldwide for similar uses within laboratories. The benefits of the new system being implemented within one Texas-based laboratory are said to be a “greater convenience”(Tripathi, 2005), “reduced theft and shrinkage”(Tripathi, 2005) and finally “increased efficiency”(Tripathi, 2005).

Although the benefits have been said to aid the productivity of the laboratory. This has come with a cost especially at the beginning of the implementation. Firstly there were “issues getting readings from tags on items packaged in metallic foil”(Thomas, 2005), this is an issue among many RFID tags as the metal interferes with the antenna of the transponder confusing the signal of the transponder sometimes resulting in errors whilst reading or a transponder that becomes completely obsolete. “Tags were problematic to begin with”(Thomas, 2005). This was an unexpected issue that faced Terso, the company in charge of leading the development and implementation of the system. Although this issue was resolved Terso would not comment on how this issue was rectified. Terso issued 150 “RFID keycard carrying a unique six-digit ID number”(Collins, 2005). This indicated that the data stored on these cards have little to no security measured associated with them and this sets them up for a potential breach of security in the form of cloning, skimming and/or manipulation. Terso reports that it is currently using 2.4 GHz for the PromegaExpress, the system used for managing the supply chain of the medical stock, however, they have yet to divulge the specifications of the RFID key cards. If they are only being used to store the “six-digit ID number” and with the fact that the company purchasing the system from Terso would have to pay for the 150 RFID key cards, it can be assumed that the key cards would be of the lower frequency of 125 KHz. These would be classed as Low frequency (LF) transponders. These would be chosen firstly as they are cheaper than the 2.4 GHz tags, secondly the LF transponders are only capable of
storing a smaller amount of data when compared to the Ultra-High frequency (UHF) 2.4 GHz tags, as these tags are only storing a six-digit ID number, and LF tag would be more than capable of storing this small data. Due to this assumption, the key cards are susceptible to be comprised, as they make no mention of any form of security or encryption. Studies or statement that the key cards have any security or encryption is lacking. As these ID numbers are “unique” there should be more security measures in place to mitigate the risks associated with the lack of security of LF transponders.

This study determined that whilst the system in which is implemented may be secure and without flaws, smaller systems or subsystems can be easily overlooked and the security compromised relatively effortlessly. Findings demonstrate that cost, storage capacity and usage can all influence an organisation's choice in the selection of RFID solutions. Furthermore using RFID in organisations can increase productivity and reduce costs, however, this might jeopardise the security of the system and organisation.

7.2 RFID Technologies and Application

Within the RFID industry, there are still significant issues with regards to performance and integration. As stated prior RFID is still very much in an “infancy” stage of development. This being said the range for advancement is vast and very much a realisation within the industry. “There are so many other components to the system in which RFID participates, and, for RFID to really blossom, every component of the system must blossom.” (Sarma, 2008) As there are so many elements to RFID systems, these facilitate the scope for development in all areas. Both physical and logical developments are still possible at this stage. RFID hardware currently seems to be more widely available compared to software. This gives a great opportunity for the progression of software within this area. “There is great scope for research in software systems for RFID” (Sarma, 2008). Making advances in software could lead to the further advances in hardware also.

RFID is starting to show more and more prominence within other industries such as fashion such as Levis Jeans, groceries such as Mark and Spencer’s and health products with Gillette. As evidence of the usefulness of RFID in different industries, Gillette found a 19% increase in promotional sales due to the use of RFID (Murphy, 2005). All these companies are currently implementing RFID to track stock. They are using RFID solution to combat counterfeiting and improve on-shelf availability. More recently, development in RFID has increased the capabilities in terms of memory size. EPC generation II-compliant tags with extra memory have been announced that they will be available soon (Scholz-Reiter et al., 2008). Most manufacturers offer tags with a maximum memory size up to 1kbit. However, the maximum memory available today is 64kbytes. This gives a huge amount of scope to hardware manufacturers for development in the future.

On the contrary, to these developments, key elements are still being overlooked in regards to security, implementation and so on.

Firstly, standards for RFID vary dramatically. Demographics are playing a huge part in the disadvantage to RFID implementation. Throughout the world, from country to country, there are “different standards for RFID” (Hardgrave, 2006). This differentiation in global standards can make it difficult for systems to interact, as they will be designed for a specific location and manufactured to a certain standard.

In regards to LF (Low Frequency) transponders, as the technology gains more exposure to the mass market, the manufacturer of “Low-cost” RFID tags are becoming more of the norm. Unfortunately, these “low-cost RFID transponders are limited with respect their maximum gate count (as the chip size influences transponder cost)” (Juels, 2004). These low-cost tags are more commonly the tags with the greatest range of scope for security breaches. As they
have cheaper chips, therefore they cannot physically store the same amount of data as a more expensive tag. This lack of storage available on tags results in a tag without the capacity to implement the appropriate security measures to ensure a secure RFID solution.

This could be an explanation as to why HF (High Frequency) RFID tags and readers are becoming more of the norm within current society.

With the increased implementation of RFID, especially low-cost solutions, this “makes the devices susceptible to physical attacks.” (Feldhofer et al., 2004). As stated prior without advances in all elements of the RFID implementation, in this case being the physical security, the systems can start to show huge holes for exploitation, rendering the security system useless and leaving the asset under this security viable for a breach.

Another issue with the establishment of RFID systems faced by project managers “is the lack of diagnostic tools” (Sarma et al., 2001). This makes decision making in regards to RFID suitability incredibly difficult, as there are few logical methods of characterising RF environments and performance on specific products. This lack of tools can often result in a significant amount of guesswork when deploying these new systems within current working environments.

In the manufacturing process, the EPC GenII chip has been selected for a large proportion of the tags available on the market today. It is used to drive all EPC tags in use today (Sarma, 2005). This leads onto another issue with RFID, specifically being the security thereof. EPC GenII does not support cryptography, and few existing products have cryptographic functions. This limits this type of tag to the security methods that it can actually facilitate.

Finally, “it is important to note that all devices and/or systems that operate at the same or similar frequencies have the potential to interfere with each other” (Menges, 2006). This can make the selection of the frequency of RFID system increasing difficult if other systems are currently deployed. When selecting the frequency of the desired systems careful consideration must be taken as each system has its benefits and its limiting factors.

The use of brute force attacks is a relatively new concept to RFID security. The most common type of data breach for RFID is skimming; this involves reading the card and copying the information from it. However, using a brute force method of card emulation would not require the need for the skimming process. This often involves a close proximity to the target card. If a brute force method were to be successful, the need for the card would be redundant as the brute force attack could emulate the data on the card. “Many brute-force attacks are based on pre-compiled lists of usernames and passwords, which are widely shared” (Owens, 2008). This method of using “pre-complied lists” would speed up the process of the brute force, however; further investigations must take place in the investigation of the correlation between data held on various cards.

In summary, it is apparent that the development of all RFID systems is occurring rapidly. All wavelengths are reaping the benefits of increased exposure and an increasing understanding of the systems. However, there is still a huge amount of this technology that is yet to be explored. Moreover, they are still essential elements of these systems that need to be advanced, such as security, to better increase the chances of progression in other areas of the technology.

There is a vast scope for development in both the software and hardware aspects of this. This development will be crucial to sustain the rate of growth within the commercial industry and to increase the exposure to this technology.

The future of RFID is still yet to be decided. If progression maintains this inertia the possibilities for this technology is endless. There are talks of physical hardware developments with tags that have actuators directly attached to them whereas currently, the tags trigger solenoids and suchlike (IEEE, 2000). Having a physical actuator hardwired to the tag would give the tags greater flexibility to the situations in which they can be placed.
“The market for RFID technology is growing rapidly” (Miles et al., 2008). This statement defiantly rings true from the gathered research. RFID is becoming a greater asset to everyday life and therefore should be developed further with greater security measure in place.

8.0 Project Methodology

For this product, an interpretative research philosophy will be adopted involving an inductive research strategy and quantitative data gathering. This is the methodology that has been selected for this project as it is deemed to yield the best and most accurate result given the subject matter.

An interpretative research philosophy is the method of research concerning human and their interaction within society. This method will be used when conducting the questionnaire as part of this project. The basis of the questionnaire is to gather information from the general public to gain an insight into their comprehension of RFID and the risks associated with it. Using an unknown sample group will mitigate any bias that may be associated with specific social groups such as knowledge or social status.

The inductive element of the research will be conducted in the validation of the theory that has been put forward. Throughout this project, the theory will be developing as more information is discovered. Also, this theory will be influenced by the experiments that will be conducted, which will either prove or disprove the theory put forward.

For quantitative data gathering, this will be in the form of a questionnaire as alluded to previously. The sample size of this questionnaire will be medium in size and aims to gather around 30 to 50 participants in which patterns can be formed in the results. The questionnaire aims to be short and concise only focusing on the exact information required for this study. No information involving personal details or social status will be asked, as this information is not deemed important in the compounding of the results.

From this questionnaire, a purposeful sampling technique will be used to combine the gathered results to define patterns in the given information. The researcher will use their best judgment to draw conclusions on the research and use this information to define the level of understanding of the general public.

The reason as to why a questionnaire was selected as part of this research project is that a questionnaire is a standardised method of gathering information. Moreover, the information that is gathered is directly comparable making the process of drawing conclusions simpler and more accurate.

Finally, a documentary analysis has been conducted in the form of a literature review (section 7.0). This gives the insight into the calibre of work already conducted and identifies an area of improvement that may be required. Using a documentary analysis also enables researchers to discover to methodologies that the author may have used. This information can be invaluable when it comes to conducting similar research.

To conclude, the methodologies defined for this project will be the backbone of the research conducted. This will enable the researcher to focus on deriving clear and concise results from the experiments conducted. Furthermore, using a questionnaire will provide an unbiased and informed overview of the knowledge and understanding of RFID within the general public.

9.0 Project Planning

It is critical that this project adheres to a strict time schedule as there is a known deadline in which the project must be completed. Furthermore, it is essential that this project have ample time towards the end as to allow time for procedures such as proofreading and printing. A Gantt chart (Fig1.1) has been produced to clearly outline the timescale for each of the tasks
that are associated with this project.

![Gantt chart depicting schedule of project.](image)

Fig. 1.1. Gantt chart depicting schedule of project.

The plan allows for 30 days for the questionnaire to be answered, this should be a sufficient amount of time as the questionnaire will be short, consisting of no more than 10 questions. The Publish questionnaire task evolves the initial publication of the questionnaire and the waiting period for the results to be gathered. After these 30 days, the questionnaire will be closed and no more answers will be considered. This allows for sufficient time to process the questionnaire result and to draw conclusions of the participant’s results.

To conclude, adhering to this plan will be paramount if the project is to be completed on time. Moreover, the plan allows for copious amounts of testing throughout, if some of the stages within this plan overrun then these essential testing periods maybe reduce, potentially jeopardising the legibility of the findings.

10.0 Requirement Analysis

As part of the research element of this project, other areas of research have been conducted in more specific areas. This gives greater insight into all aspects that the project may involve throughout. Specialised areas have to be explored for the feasibility and connection with the technology used within this project.

10.1 Brute force attacks

Brute force attacks are the methods of hacking by the means of trying many passwords or passcodes in the hope that eventually the system will guess the correct code. The attack systematically checks all possible passwords and passcodes until a correct code is found. This will be relevant in this experiment, as the prototype will sequentially write all possible codes in an iterative manner. The duration of the brute force attack will be dictated by the security held on the tag and length of code that is trying to be guessed by the prototype. This method of the brute force attacking is suitable for this project, as this type of attack does not require any predetermined information.

In summary, this method of hacking will be implemented within the prototype, as it does not require any information to begin the process. Furthermore, providing the brute force method is deployed correctly all the potential codes will be tested systematically without missing a
single code possibility.

10.2 Iterative Adding Methods

An iterative adding method is when a single value is added and is then looped multiple times to achieve the desired value. This method of the addition will work congruently with the brute force method by trying all the potential values. If the code that is proposed by the brute force method is found to be incorrect then an iterator will be triggered adding a single value to the code. This code will then be tested and this entire process will be looped until a correct code is found. By adding singular values it can be ensured that no potential sequence of code will be missed and all codes will be tried systematically. This method of iteration will be implemented within the prototype, alongside the brute force method to test all potential sequences of access codes. Using an iterative method ensures that all the potential codes are tried providing the iterator is started with a value of 0.

10.3 Rainbow Tables

A rainbow table is a method of reversing hash values and hash functions with the use of precomputed tables. Rainbow tables are invaluable tools used by hackers to derive plaintext from cryptographic hashes. Working on a premise of using some predetermined information and a method of reverse engineering they systematically compute possibilities and decide on the most probable result, in many cases the plain text, unhashed password. In regards to this project, a similar method of rainbow tables could be used by gathering information from existing systems and using this information to narrow down the more common and more likely code combinations. Similar to a dictionary attack of the use of other gathered information, however dictionary attacks are often unsuitable due to validation methods such as limiting the number of attempts of data entry. As of current research, there is no evidence that shows that there are common code combinations that are frequently used. This would make an attack that uses rainbow tables or a form of dictionary attack difficult as there is little information available about the formulation of codes on RFID transponders. Rainbow tables or a derivative thereof will be considered in this project if it is deemed applicable and if there is a direct benefit with their implementation.

10.4 Encryption

Encryption is the method of converting information into a form of code. The main purpose of encryption is to prevent unauthorised access to the information. Encryption has become common practice among many technologies in recent years as the threat and value of data increases.
Fig.1.2. Statistics of encryption within varying industries. (Gemalto, 2015)

Fig.1.2 shows the sheer amount of information that has been lost or stolen in 2015. This vast amount of information that has been lost or stolen would be available to be sold onto the black market or used further by the hackers. “Only 4% of breaches were “Secure Breaches” were the encryption was used and the stolen data was rendered useless” (Gemalto, 2015). This shows the lack of encryption methods currently being used within current systems.

In regards to RFID, some methods of encryption have been developed to safeguard the information held on the tags. Also, some effort has been made to the security of the readers, being both hardware and software.

Firstly, some blocks of the tags are “Read-Only” meaning that no information can be written to them as a result. This is usually the first block, as this block is used to store the tags UID (Unique IDentification) number. This block is usually formulated by the manufacturer of the tag and is sometimes printed on the tags for reference. Furthermore, the 4th block in every sector is a key block. The first 6 and last 6 bytes of this 4th block are reserved for authentication purposes. To determine the authenticity of the tag the reader will read the bytes within the 4th block of each sector. This property of the tag that is read only can act as an assurance that the first block is unable to be altered.

RFID tags use a system of ciphering to more efficiently store the information and to also make the information on the tags readable by the system, as most readers will not understand plain text information. An algorithm is used to convert plain text into cypher text with is then stored onto the card. The same algorithm is then used to decipher the cypher text back into plain text. The software supporting the reader completes this deciphering process.

Information is stored in the EEPROM module of the tag. This information usual consists of the given information such as access codes, UID and a cryptographic key. The EEPROM must store this information even when the supply of power is lost. The EEPROM receives a pulsed voltage to write to the relevant sectors within the tag and stores this information until it is overwritten.

In more recent developments of the technology, methods of data encryption have started to be implemented in more expensive chips. AES (Advanced Encryption Standard) has been introduced to encrypt the data held on RFID tags. The AES module “calculates strong cryptographic authentication” (Feldhofer et al., 2004). These AES modules are designed to function under “low-power and low die size restrictions” (Feldhofer et al., 2004). As to be expected with these RFID tags that use the AES encryption methods, they have a significant price increase compared to tags that do not implement this security measure. Currently, online market stores such as eBay® are selling 5 tags at £2.29. Alternatively, a company called ID Management Systems® are retailing 5 tags with the AES
module embedded for £55.38. Moreover, as the cost of these tags is considerable higher than the tags without the AES module, in most situation companies with forgo this more expensive tag and select the tags that are cheaper by do not have this AES module. This then creates a lack of standardisation among tags if some are using the AES capabilities and some are not. This can create inconsistencies within the system and result in some tags or readers becoming unusable.

10.5 Arduino

Arduino is a microcontroller board used to prototype electronics. Arduino boards are inexpensive, with many I/O (Input / Output) ports for various applications. For this project, an Arduino Uno will be used, as this is a common board that is often used for rapid prototyping. The Arduino also allows for Serial output, which is useful to demonstrate processes and gives feedback to the user. Arduino Uno uses an ATmega328 chip for the computation of the board. This chip is programmable and can be easily removed from the board and implemented into other projects. The Arduino Uno allow for modules and sensors to be added to the board to give the board increased functionality.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance / Clock speed</td>
<td>20 MIPS (Millions of instructions per second) at 20 Mhz</td>
</tr>
<tr>
<td>CPU Type</td>
<td>8-bit AVR</td>
</tr>
<tr>
<td>Maximum operating frequency</td>
<td>20 Mhz</td>
</tr>
<tr>
<td>Flash memory</td>
<td>32kB</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1kB</td>
</tr>
<tr>
<td>Pin count</td>
<td>28-pin</td>
</tr>
<tr>
<td>I/O Pins</td>
<td>26</td>
</tr>
<tr>
<td>USB interface</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Fig.1.3. ATmega328 specifications.

Alongside the module that will be used for this project other components will be used to better demonstrate the process of the prototype during operation. LEDs (Light Emitting Diodes) will be added to the prototype as visual confirmation that the prototype is functioning correctly, also to signify that the sequence has been completed. Furthermore, a piezo buzzer will be added, which will give a short beep when a sequence has been completed. Finally, an LCD screen will be added to display current processes of the prototype.

10.6 MFRC522

The MFRC522 module is the selected module for this project. This module allows the Arduino to read and write to RFID-enabled tags. This module works within the High Frequency (HF) range with a working frequency of around 13.56Mhz. This is a typical frequency among HF devices. The module has 8 pins of which 7 need to be connected to the Arduino.
11.0 Questionnaire

As part of this study, an online questionnaire was published to gather insight into the perception of RFID and its security. As the questionnaire was completely anonymous there is no way to identify the type of people that participated in the questionnaire. Knowing this, there is no way to correlate any other indicating factors such as age or economic or social background.

To begin the questionnaire the participants were asked their knowledge of RFID. The participants were asked if they knew what RFID is and if so, to elaborate on their knowledge. The results were unsurprising and were concluded to $3:1.375 \approx 3:1.375$ stating that they did not know what RFID was.

This statistic shows that there is still very little knowledge of RFID even after many years of the technology being available. This technology is being implemented within various organisations and in different products and yet people are still not aware of the technology they may be using on a daily basis. From the answers in which the participants knew what RFID was, the general consensus was that the main uses were in the cards such as “contactless bank cards”.

Following this, the participants were asked if they use any form of RFID on a daily basis. The results are reflective of the question prior as 70% of the participants stated that they use a form of RFID for 0 to 1 times a day. However, as identified in the previous question there is little knowledge of RFID so it could be said that people are using RFID systems without the awareness that they are using the technology. As RFID is the technology that is often referred to in different terms such as “contactless” this could affect the understanding of the participants using these systems. This question also uncovered that the other 30% used the technology more than 2 times a day. From this 30% of results, 5% used this technology more than 10 times a day. This 5% that use RFID more than 10 times a day are clearly aware of the uses of RFID and the benefits. A point should be made that in certain organisations or companies RFID systems are mandatory to access specific areas within a building or to retrieve certain resources. In most situations, these systems are in the form of RFID access cards to allow them to gain access to the building acting as a security measure. Furthermore, some companies use these RFID cards to act as a method of signing into account on workstations.

The participants were then asked about their use of Apple Pay and Android Pay. These payment methods alongside Samsung Pay are the most popular types of mobile payments. In a recent survey, it was found that 61% of US consumers are now replacing cash transactions with mobile payments. (ACG, 2016). This study shows that the use of these methods is a growing element within current society. In reference to the questionnaire that was conducted it was found that 42.5% of the participants use a method of mobile payments. This usage
statistic indicates that there is a significant usage of this mobile payment technology, however, there is still little awareness surrounding the technology involved with the processes.

The questionnaire then asked if the participants own any of a selected list of items that all use some form of RFID. The list contained; Contactless bankcard, Oyster card, iff card, Card to gain access to buildings or facilities and other. It should be said that some of the options are geographically specific and are only obtainable by living in certain locations. Furthermore, consideration must be made to the willingness of the participants to relinquish information relating to financial processes, such as the use of bankcards. It was identified from the questionnaire that the most used RFID enable items that people use are contactless bank cards, access cards and then oyster cards. 87.5% of the participants stated that they own a contactless bank card. “There are a total of 102.7m contactless cards in issue in the UK.” If this was compared to March 2014 where only 43.3m contactless cards were issued. (Theukcardsassociation.org.uk, 2017). This growth in a short amount of time can indicate the growth of the technology, especially within the financial industry. Following bank cards, access cards were the second most used item found within this questionnaire. Many companies are using or starting to implement access card for their employees for security and accountability reasons. Using contactless access cards are often used to secure specific areas that may require prior authorisation to access. This authorisation can be assigned to the card enabling the employee access to that area or to a specific resource that may be of a confidential nature.

The next question asked was if the participants knew of any risks associated with contactless payment methods. The questionnaire found that 47.5% of the people asked said they didn’t know of any risks associated with the use of contactless payment methods. This is a staggering statistic considering that 30% of the participants use RFID on a frequent daily basis. This statistic indicates perfectly the lack of information on the security issues related to this technology, yet people are still using the technology completely unaware of the risks associated with it. From this question, 52.5% of the participants said they did know of a risk(s) associated with this technology. They were asked to give examples of the risks they believe were associated with contactless payment methods. The two most common associated risks were fraud/identity theft and skimming. Almost half of the risks were in relation to fraud and/or identity theft. This was explained as “Loss of card” or “If card gets stolen”. As “Contactless transactions do not require verification of the cardholder’s PIN.” (Emms et al., 2013). This can leave them unprotected for fraudulent use if lost or stolen as the thief would not require any other information such as a PIN number. The second most common reply to this question was skimming, however, the term skimming wasn’t used. The replies from participants were answers such as “Readers can be used to gain information from the bank card” and “People accessing your card through your wallet or purse.” These are typical examples of skimming. This indicated that people are familiar with the concept of contactless theft of information, however, are unfamiliar with the terminology and how the skimming process works. This is an indicator of the general awareness of the security of RFID and shows that there is some awareness of the risks associated however there is still a certain level of unfamiliarity with other associated risks.

Finally, the participants were asked if their place of work currently implements any form of RFID systems. As the type of people that participated in the questionnaire is unknown, some allowance should be made as some of the participants may be unemployed or may be unaware if a system is currently in place. It was found that 62.5% of the participants answered no to this question showing that their current employer does not currently use any type of RFID system. However, 37.5% of the participants replied, yes, implying that there is currently some form of RFID system within their workplace. This indicates that there is currently a lot of
companies that are using a form of a RFID system.
In conclusion, it has been identified that there is a general understanding of some of the security concerns associated with RFID yet there is still an area of understanding that is lacking. There seems to be an overall understanding of the uses of RFID and the situations in which it is currently being used, however, it is apparent that there is still some distance to go for this general understanding to ensure there is a consistent and coherent knowledge among both users and organisations of the risks and potential exploits that could jeopardise the effectiveness of the systems.

12.0 Design

For the design of the project, careful consideration had to be made in order to make this prototype viable for the situation. The prototype needed to be small, handheld and portable making it suitable to any RFID-enabled door.
Research that has been conducted has returned no results stating that any other research in which an iterator has been used to breach an RFID-enabled door. There is some development within the brute force area using Perl scripts, however, this research is primarily focused on the discovery of common access codes and default tag information. (Grunwald, n.d.)
First, the ATmega chip was removed from the Arduino as to reduce the size of the prototype. This meant that ports and I/O headers didn’t waste valuable space as the components required for this project could be attached directly to the chip. A small breadboard was used for this prototype as it made swapping out components faster than if they were soldered together. Also being built on a breadboard meant that all the components could be used in different situations if the project required without a lengthy process of desoldering the components. A wiring diagram has been produced to illustrate the prototype and the connections required (Fig.1.5). This diagram illustrates the connections needed for all the components such as the LCD screen, ATmega chip, RC522 module the piezo and the LEDs.

![Wiring diagram for prototype.](image)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>MFRC522 Module</th>
<th>LCD w/ I2C Backpack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance / Clock speed</td>
<td>13.56 MHz</td>
<td>100 kHz</td>
</tr>
<tr>
<td>CPU Type</td>
<td>RC522</td>
<td>8-bit PCF8574T</td>
</tr>
<tr>
<td>Maximum operating frequency</td>
<td>13.56 MHz</td>
<td>100 kHz</td>
</tr>
<tr>
<td>Flash memory</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>EEPROM</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Pin count</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>I/O Pins</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Interface Type</td>
<td>Serial UART</td>
<td>I2C Interface</td>
</tr>
<tr>
<td></td>
<td>SPI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I2C Bus Interface</td>
<td></td>
</tr>
<tr>
<td>I2C Address</td>
<td>N/A</td>
<td>0x3F, 0x40</td>
</tr>
</tbody>
</table>

Fig.1.6. MFRC522 Module and LCD specifications.

The final prototype was able to fit into a small project box making it easily portable. The prototype was also lightweight, only weighing a total of 138g. Moreover, the benefits of using the project box are that it conceals the project. This protects the prototype during use but also shields the prototype from view, this could be important if the prototype was being used covertly.

As the project did not have a serial output, it felt necessary to add indicators to the prototype to show functionality. This was done in the form of 2 LEDs and a piezo buzzer. One LED was used to show that the prototype was correctly writing to the tag and the other LED would be illuminated once the loop had been completed. The piezo would also be triggered at this stage of completion.

Finally, the LCD screen was added to display to the users that the prototype is running correctly and display current stage of progress.

13.0 Implementation Testing

This stage of implementation testing is when the prototype is used in the situation in which is intended. In this situation, a simulation was created with another reader RFID module was set up to act as an RFID-enabled access door. This simulation used the same RFID module as the prototype to ensure that the tag could be written by prototype and read by the reader. A cardboard door was created with the reader placed behind it to replicate the RFID enabled door within a building (Fig.1.7).
Fig. 1.7. RFID door simulation.

Fig. 1.8. Completed sequence by prototype, LED lit as indicator to user.
Upon testing the prototype in the simulation one key limitation was discovered that limited
the functionality of the prototype. This limitation was the interference between the 2 modules
(RC522 (A) & RC522 (B)). When the modules were brought into a close proximity they
would both cease to read and/or write. This limiting factor will be discussed further in a
following section.

Another limitation was discovered at this stage, this was the speed in which the brute force
attack could occur. To complete all the possibilities at the current rate would take a significant
amount of time $\approx 712555942097919992$ millennia This fundamentally lead to a shorter code
being produced designed to test the prototype and the theory in a shorter amount of time. This
code was tested in the same way, as the longer code would be.

The prototype completed the loop, with all successful writes the tag. This code focuses on
writing into block 4 in sector 1. This test demonstrated that this method of iterative writing to
a tag is possible. The shorter code that was produced uses the 15 and 16 bytes for the
destination of the iteration and is complete when all possibility of codes are tested. However,
in the longer code all the bytes within sector 1 block 4 would be tested up the value of 255 for
each byte.

14.0 Code

As discussed prior, two sets of codes were created for this project. The first developed was
the complete code, which will run until all code possibilities have been exhausted. The second
code produced, uses the same method of iteration but only for bytes 15 and 16 to shorten the
process. This shorted code was used to test the concept and functionality of the prototype
without the limitations influencing the results. Both sets of code have been uploaded onto
separate ATmega chips so that either code can be easily implemented without writing or
deleting code unnecessarily.

```c
#include <SPI.h> // RC522 Module uses SPI protocol
#include <MFRC522.h>

#include "Wire.h"
#include "LCD.h" //Libaries associated with LCD
#include "LiquidCrystal_I2C.h"

#define COMMON_ANODE // Define common anode of LEDs
#define LED_ON HIGH
#define LED_OFF LOW

#define greenLED 6
#define blueLED 5

#define SS_PIN 10 // Define communication pins
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance

MFRC522::MIFARE_Key key;

byte C1 = 0; // RFID sector bytes
byte C2 = 0;
```
byte C3 = 0;
byte C4 = 0;

byte D1 = 0;
byte D2 = 0;
byte D3 = 0;
byte D4 = 0;

byte E1 = 0;
byte E2 = 0;
byte E3 = 0;
byte E4 = 0;

byte F1 = 0;
byte F2 = 0;
byte F3 = 0;
byte F4 = 0;

LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7); //I2C address of LCD
// setup of LCD initialization of LCD parameters
void setup() {
    pinMode(greenLED, OUTPUT);
    pinMode(blueLED, OUTPUT);
    digitalWrite(greenLED, LED_OFF);
    digitalWrite(blueLED, LED_OFF);  //Priming LEDs

    //Protocol Configuration
    Serial.begin(9600);    // Initialise serial communications with PC
    SPI.begin();           // SPI protocol
    mfrc522.PCD_Init();    // Initialise MFRC522 Hardware

    lcd.begin (20,4);
    lcd.clear();
    // LCD Backlight ON
    lcd.setBacklightPin(3,POSITIVE);
    lcd.setBacklight(HIGH);

    lcd.home (); // go home on LCD
}

//write to tag, if successful, loop and preform again
void loop () {
    byte sector       = 1;
    byte blockAddr    = 4;
    byte dataBlock[]  = {

byte trailerBlock = 7;
byte buffer[18];
byte size = sizeof(buffer);

// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent())
    return;

// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial())
    return;

MFRC522::StatusCode status; //Define Status Codes

// Autentication of A and B blocks
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, trailerBlock, &key, &(mfrc522.uid));

mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_B, trailerBlock, &key, &(mfrc522.uid));

mfrc522.MIFARE_Write(blockAddr, dataBlock, 16); //Wrtie to PICC

if(F3 == 2){
    completedSeq();
}

} else {

    if(status == MFRC522::STATUS_OK){
        F4++;
        if(F4 == 255){
            F4=0;
            F3++;
            lcd.clear();
        } else {
            if(F3 == 255){
                F4=0;
                F3=0;
                F2++;
                lcd.clear();
            }
        }
    }
}
digitalWrite(blueLED, LED_ON);
digitalWrite(blueLED, LED_OFF);

lcd.setCursor (0,0);
lcd.print(C1);
lcd.setCursor (4,0);
lcd.print(C2);
lcd.setCursor (8,0);
lcd.print(C3);
lcd.setCursor (12,0);
lcd.print(C4);

lcd.setCursor (0,1);
lcd.print(D1);
lcd.setCursor (4,1);
lcd.print(D2);
lcd.setCursor (8,1);
lcd.print(D3);
lcd.setCursor (12,1);
lcd.print(D4);

lcd.setCursor (0,2);
lcd.print(E1);
lcd.setCursor (4,2);
lcd.print(E2);
lcd.setCursor (8,2);
lcd.print(E3);
lcd.setCursor (12,2);
lcd.print(E4);

lcd.setCursor (0,3);
lcd.print(F1);
lcd.setCursor (4,3);
lcd.print(F2);
lcd.setCursor (8,3);
lcd.print(F3);
lcd.setCursor (12,3);
lcd.print(F4);

// Stop encryption on PCD
mfrc522.PCD_StopCrypto1();

//Writing of array to tag
void dump_byte_array(byte *buffer, byte bufferSize) {
  for (byte i = 0; i < bufferSize; i++) {
    Serial.print(buffer[i] < 0x10 ? " 0" : " ");
    Serial.print(buffer[i], DEC);
  }
}
void completedSeq()
{
    tone(7,20,7500);
    for (byte IC = 0; IC < 3; IC++) {
        digitalWrite(blueLED, LED_OFF);
        digitalWrite(greenLED, LED_ON);
        delay(500);
        digitalWrite(greenLED, LED_OFF);
        delay(500);
    }
}

The shortened code will loop until the 15th sector (F3) = 2. In the process the code will designate the area in which is desired to be written to then checks if the tag is available to be written too, followed by an authentication process of blocks. An if statement is then looped to check if the desired code combination is reached, in this instance the desired code has been set to 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0. When the if statement is fulfilled then it will call a function called completedSeq() which activates the piezo and the LED to indicate that the desired code has been reached.

#include <SPI.h> // RC522 Module uses SPI protocol
#include <MFRC522.h>

#include "Wire.h"
#include "LCD.h" //Libaries associated with LCD
#include "LiquidCrystal_I2C.h"

#define COMMON_ANODE // Define common anode of LEDs
#define LED_ON HIGH
#define LED_OFF LOW

#define greenLED 6
#define blueLED 5

#define SS_PIN 10 // Define communication pins
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance

MFRC522::MIFARE_Key key;

byte C1 = 0; // RFID sector bytes
byte C2 = 0;
byte C3 = 0;
byte C4 = 0;

byte D1 =0 ;
byte D2 =0 ;
byte D3 =0 ;
byte D4 =0 ;
byte E1 = 0;
byte E2 = 0;
byte E3 = 0;
byte E4 = 0;

byte F1 = 0;
byte F2 = 0;
byte F3 = 0;
byte F4 = 0;

LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7); //I2C address of LCD
// setup of LCD initialization of LCD parameters

void setup() {

    pinMode(greenLED, OUTPUT);
    pinMode(blueLED, OUTPUT);
    digitalWrite(greenLED, LED_OFF);
    digitalWrite(blueLED, LED_OFF);  //Priming LEDs

    // Protocol Configuration
    Serial.begin(9600);    // Initialise serial communications with PC
    SPI.begin();           // SPI protocol
    mfrc522.PCD_Init();    // Initialise MFRC522 Hardware

    lcd.begin (20,4);
    lcd.clear(); // LCD Backlight ON
    lcd.setBacklightPin(3,POSITIVE);
    lcd.setBacklight(HIGH);

    lcd.home (); // go home on LCD
}

//write to tag, if successful, loop and preform again

void loop () {

    byte sector       = 1;
    byte blockAddr    = 4;
    byte dataBlock[]  = {
        C1, C2, C3, C4, //  1,  2,   3,  4,
        D1, D2, D3, D4, //  5,  6,   7,  8,
        E1, E2, E3, E4, //  9, 10,  11, 12,
        F1, F2, F3, F4  // 13, 14,  15, 16
    };
}
byte trailerBlock = 7;
byte buffer[18];
byte size = sizeof(buffer);

// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent())
    return;

// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial())
    return;

MFRC522::StatusCode status; //Define Status Codes

// Authentication of A and B blocks
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, trailerBlock,
                        &key, &(mfrc522.uid));

mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_B, trailerBlock,
                        &key, &(mfrc522.uid));

mfrc522.MIFARE_Write(blockAddr, dataBlock, 16); //Write to PICC

if(F4 == 255 && F3 == 255 && F2 == 255 && F1 == 255
  && E4 == 255 && E3 == 255 && E2 == 255 && E1 == 255
  && D4 == 255 && D3 == 255 && D2 == 255 && D1 == 255
  && C4 == 255 && C3 == 255 && C2 == 255 && C1 == 255){
    completedSeq();
}

if(status == MFRC522::STATUS_OK){
    F4++;
    if(F4 == 255){
        F4=0;
        F3++;
        lcd.clear();
    }
    if(F3 == 255){
        F4=0;
        F3=0;
        F2++;
        lcd.clear();
    }
    if(F2 == 255){
        F4=0;
        F3=0;
    }
F2=0;
F1++;
lcd.clear();
}
    if(F1 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4++;  
lcd.clear();
}
    if(E4 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3++;
lcd.clear();
}
    if(E3 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2++;
lcd.clear();
}
    if(E2 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1++;  
lcd.clear();
}
    if(E1 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4++;
lcd.clear();
}
if(D4 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3++;  
lcd.clear();
}
if(D3 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2++;  
lcd.clear();
}
if(D2 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D1++;  
lcd.clear();
}
if(D1 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4++;  
lcd.clear();
}
if(C4 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3++;  
lcd.clear();
}
if(C3 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3=0;
C2++;  
lcd.clear();
}
if(C2 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3=0;
C2=0;
C1=0;
lcd.clear();
} 
if(C1 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3=0;
C2=0;
C1=0;
D4++;
lcd.clear();
}
if(D4 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3=0;
C2=0;
C1=0;
D4=0;
D3++;
lcd.clear();
}
if(D3 == 255){
F4=0;
F3=0;
F2=0;
F1=0;
E4=0;
E3=0;
E2=0;
E1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3=0;
C2=0;
C1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3=0;
C2=0;
C1=0;
D4=0;
D3=0;
D2=0;
D1=0;
C4=0;
C3=0;
C2=0;
C1=0;
D4=0;
D3=0;
D2=0;
D1=0;
lcd.clear();

if(D1 == 255)
{
    completedSeq();
}

digitalWrite(blueLED, LED_ON);
digitalWrite(blueLED, LED_OFF);

lcd.setCursor (0,0);
lcd.print(C1);
lcd.setCursor (4,0);
lcd.print(C2);
lcd.setCursor (8,0);
lcd.print(C3);
lcd.setCursor (12,0);
lcd.print(C4);

lcd.setCursor (0,1);
lcd.print(D1);
lcd.setCursor (4,1);
lcd.print(D2);
lcd.setCursor (8,1);
lcd.print(D3);
lcd.setCursor (12,1);
lcd.print(D4);

lcd.setCursor (0,2);
lcd.print(E1);
lcd.setCursor (4,2);
lcd.print(E2);
lcd.setCursor (8,2);
lcd.print(E3);
lcd.setCursor (12,2);
lcd.print(E4);

lcd.setCursor (0,3);
lcd.print(F1);
lcd.setCursor (4,3);
lcd.print(F2);
lcd.setCursor (8,3);
lcd.print(F3);
lcd.setCursor (12,3);
lcd.print(F4);

} // Stop encryption on PCD
mfc522.PCD_StopCrypto1();}
void dump_byte_array(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        if (buffer[i] < 0x10) Serial.print(" 0");
        Serial.print(buffer[i], DEC);
    }
}

void completedSeq()
{
    tone(7,20,7500);
    for (byte lC = 0; lC < 3; lC++) {
        digitalWrite(blueLED, LED_OFF);
        digitalWrite(greenLED, LED_ON);
        delay(500);
        digitalWrite(greenLED, LED_OFF);
        delay(500);
    }
}

For the longer code, in which will write and check all possible code combination, the same coding structure is used. However, the if statement is longer as there is more criterion in which much be fulfilled. This entails that all the bytes are tested within the selected block.

The language that was used for the prototype was C++, this was dictated by the Arduino IDE (Integrated Development Environment) as this is the language that both the software and Arduino board run on. C++ is a general-purpose high-level language, sharing features of many other object-oriented programming languages. C++ has been a massive influence on languages such as C#, Perl and Java which are now, alongside C++ very popular languages for their specific applications (Meyerovich et al., 2013). The Arduino IDE is how the code is then uploaded to the chip through the Arduino board. This chip can then be removed from the board and implemented onto a breadboard. This allows the chip to be programmed and reprogrammed for debugging purposes.

In conclusion, using an Arduino for the prototype alongside the powerful IDE supplied alongside the Arduino has made it possible to use different codes without the need for reprogramming. Using separate chips for the different codes means that either code option can be selected and then implemented quickly without the need to write directly to the chip embedded in the prototype. Using C++ as the language for the prototype facilitates a level of flexibility due to the easy IDE and the object-oriented nature of the language. Using two separate codes enables the prototype to be tested in shorter periods of time, allowing the prototype to be debugged at an early stage in its development.

15.0 Limitations

During the development and testing stages, various limiting factors associated with the technology became apparent. These limiting factors fundamentally affect the feasibility of the prototype as they show a significant flaw within the technology in its current implementation. The first of the limitations is the interference issues. This is when one or more readers or tags are placed in a close proximity to each resulting in collisions of electromagnetic waves. This is a known issue within the in technology as has been a limiting factor for some time. Whilst there are engineers producing algorithms and hardware to prevent this crossover of communication, anti-collision technology is yet to be perfected so collisions are an often-
occurring incident. Anti-collision algorithms are usually implemented within the firmware of the reader and generally functions by trying to differentiate the different waves that it is receiving and tries to process them separately. There are varying levels of success with this; however, anti-collision algorithms are primary focused more towards the transponder element. “When only one tag answers, the reader successfully recognises the tag. When more than one tags answer, a collision occurs and the reader can’t recognise the tags.” (Shih et al., 2006).

In the simulation, the 2 modules (RC522(A) & RC522(B)) began to interfere with each other and the transponder resulting in both the modules to cease functionality. This is a massive flaw within the technology and why it is suggested that readers are spaced as far away from each other as possible to prevent issues such as collision or interference. “Effective site surveys are required prior to the implementation of RFID application in order to understand the current environment in which the RFID technology will be installed.” (Bennett et al., 2008).

The second limiting factor is the processing speed of the ATmega328 chip. The speed in which this chip operates isn’t the problem; the issue lies with the sheer amount of possible code combinations that the prototype must try. In the prototype, a single chip was used to test the combinations one by one in line with a typical brute force attack. At the current speed of the singular chip is would take the prototype around 2.247116419e28 seconds to complete all the possibilities, ≈ 7.1255942097919992e17 millennia (See section 21.1). This length of time to run such an experiment is not feasible and as a result, a shorter code has been created to demonstrate the theory and the development as of proof of concept. In regards to quickening the process, there are currently advances in technology such as supercomputing and quantum processing. These are often computers with a significantly greater computational power than a single computer of a single chip. Using similar processing power of these could reduce the amount of time required completing all the combinations, however, these technologies are expensive and are not designed for this particular use so significant modification would have to be made to make it relevant for this purpose.

Finally, a limiting factor with the prototype is the fact that it can only write to a single block at a time. As there can be up to 15 sectors mostly containing 3 writable blocks this increases the number of possibilities in which the assess code could be required to be read from. There is no method of identifying which sector the reader will be reading from so this identifies another limitation within the prototype.

Whilst there may be some significant limiting factor to this technology, as it develops these issue will be resolved. This, in turn, should increase the feasibility of the prototype and render is more useful as the technology improves.

16.0 Results and Discussion

Concluding on the experiment, the results that could be gathered were compiled and used for such applications such as mathematically equations for timeframe calculations. The results that were gathered were only available from the shortened code. As discussed prior the longer could not be run within a reasonable amount of time due to a large number of possible combinations.

The first objective was to produce a piece of software and/or hardware that can be used to breach an RFID-enabled door without using a predetermined access code. This objective has not been met, as the prototype that has been produced is incapable of working near other readers due to interference limitation. Controversially, the prototype does satisfy part of this objective, as the prototype is a produced piece of hardware with custom software that in theory should be able to breach an RFID-enabled door if given enough time to complete the
The experiment implementing the shortened code concluded that it is indeed possible to write in an iterative manner to an RFID tag. This addressed the second of the objectives to determine whether this was possible to write in such a manner to RFID transponders. The prototype successfully can write to the transponder using an iterator without missing potential code possibilities.

The third objective is to determine if the writer can be used to breach an RFID-enabled door. Similarly to objective one, this has been proven that it is not possible to use the writer (prototype) to breach an RFID-enabled door due to the limiting factor of interference between the modules and the transponders.

The final objective was to formulate a timeframe in which it would take for the entire process to be completed. These calculations have been completed (See section 21.1) and indicate that using a brute force attack on an RFID-enabled door would not be feasible due to the sheer amount of potential possibilities $\approx 16^{3255}$.

There is some area of discussion around the starting point in which the prototype would commence the brute force attack. The prototype could be programmed to start at a different point in the brute force attack enabling it to find the desired code in a smaller amount of time. However, there would be no guarantee that the code required may not have already been overlooked. For example, if the prototype was to begin at the half way point of the process. The code that is desired may be in the first half of the brute force attack. This would result in a prototype that never reaches the desired code.

17.0 Application

The application of this development is only restricted be the limiting factors associated with the technology. Both the prototype produced and the technology as a whole has the potential to be expanded in various situations.

In regards to the prototype, when the technology advances in respect of the interference issues, the prototype will be able to be tested in situ. However, this does not solve the issue of the timeframe in which it takes to complete the breach. “In 1965 Moore observed that integrated circuit complexity evolved exponentially” (Schaller, 1997). Moore was able to plot this evolution on a graph and discovered that there was typically a doubling of a number of transistors on IC (Integrated Circuits) chip approximately every two years. This then leads onto the prototype that has been produced. If the chip used within the prototype were enhanced, the amount of time that would be required by the prototype would be reduced, following the exponential curve proposed by Moore. This makes the prototype more feasible as the time it would require to complete the entire process would be dramatically quicker.

In regard to the technology as a whole, the possibilities for their implementation are truly endless if the limiting factors are considered and addressed. Issues such as collision can simply be avoided by spacing out the integrators allowing for the greatest amount of distance between them. However, issues such as the interference of tags and readers are a significant issue that must be addressed sharply. Moreover, if these limiting factors are addressed and resolved the applications for this technology could be greatly diverse and the technology could become a common commodity.

The limitations stated prior may possibly be addressed with further postgraduate study within this area. Moreover, future undergraduate students may use this research within their future projects.

Furthermore, this research is planned to be progressed into a conference paper as to be validated by experts within this area. The goal would be for this paper to be accepted at a conference and published publically.
It is clear that this technology has limitless application within current real-world applications, however, the progression is being restricted by the limiting factors that are associated with the technology.

18.0 Conclusion

The stages of development, implementation and testing have all been relevant to the conclusion of this project. The theory in question has been thoroughly integrated and after an extensive research period and testing stage conclusions have been made. Firstly, it is indeed possible to use an iterative method to write to RFID transponders. Furthermore, in theory, it is also viable to implement a method of brute force attack. However, due to limitations of the technology, it isn’t yet possible to fully test the prototype for its real-world application.

As for the technology, the limiting factors are also key arguments in the progression of this technology, however, some of these factors can be avoided or mitigated depending on the situations in which the technology is implemented.

The research hereby conducted proves that there is a scope for this technology within a current real-world application. Moreover, the prototype produced demonstrates that an iterative method of writing to RFID transponders is possible, yet the limiting factors such as interference and collisions are at present limiting the feasibility of this prototype.

19.0 References


20.0 Bibliography


### 21.0 Appendices

#### 21.1 Calculation

\[ x = 16^{255} = 11235582092890004495315224631 \]

Length of time required for the prototype to go from 0 to 255 in byte 16 \( \approx 45 \) seconds.

\[ y = 255 \div 45 \text{ seconds} = 5.6 \text{ possibilities tested a second} = 5 \text{ complete possibilities tested a second.} \]

\[ \frac{x}{y} = 2247116419000000000000000 \text{ seconds to complete all possibilities.} \]

\[ 22471164190000000000000000000 \text{ seconds into millennia} = 712555942097919872 \text{ millennia} \]

#### 21.2 Code Libraries

MFRC522.h
/**
 * MFRC522.h - Library to use ARDUINO RFID MODULE KIT 13.56 MHZ WITH TAGS
 * SPI W AND R BY COOQROBOT.
 * Based on code Dr.Leong (WWW.B2CQSHOP.COM)
 * Created by Miguel Balboa (circuitito.com), Jan, 2012.
 * Rewritten by Søren Thing Andersen (access.thing.dk), fall of 2013 (Translation to English,
 * refactored, comments, anti collision, cascade levels.)
 * Extended by Tom Clement with functionality to write to sector 0 of UID changeable
 * Mifare cards.
 * Released into the public domain.
 *
 * Please read this file for an overview and then MFRC522.cpp for comments on the specific
 * functions.
 * Search for "mf-rc522" on ebay.com to purchase the MF-RC522 board.
 *
 * There are three hardware components involved:
 * 1) The micro controller: An Arduino
 * 2) The PCD (short for Proximity Coupling Device): NXP MFRC522 Contactless Reader IC
 * 3) The PICC (short for Proximity Integrated Circuit Card): A card or tag using the ISO
 * 14443A interface, eg Mifare or NTAG203.
 *
 * The microcontroller and card reader uses SPI for communication.
 * The protocol is described in the MFRC522 datasheet:
 *
 * The card reader and the tags communicate using a 13.56MHz electromagnetic field.
 * The protocol is defined in ISO/IEC 14443-3 Identification cards
 * -- Contactless integrated
circuit cards -- Proximity cards -- Part 3: Initialization and anticollision".
 * A free version of the final draft can be found at
 * http://wg8.de/wg8n1496_17n3613_Ballot_FCD14443-3.pdf
 * Details are found in chapter 6, Type A – Initialization and anticollision.
 *
 * If only the PICC UID is wanted, the above documents has all the needed information.
 * To read and write from MIFARE PICCs, the MIFARE protocol is used after the PICC has
 * been selected.
 * The MIFARE Classic chips and protocol is described in the datasheets:
 * 1K: http://www.mouser.com/ds/2/302/MF1S503x-89574.pdf
 * 4K: http://datasheet.octopart.com/MF1S7035DA4,118-NXP-
 Semiconductors-datasheet-11046188.pdf
 * The MIFARE Ultralight chip and protocol is described in the datasheets:
 * Ultralight C:
 *
 * MIFARE Classic 1K (MF1S503x):
 * Has 16 sectors * 4 blocks/sector * 16 bytes/block = 1024 bytes.
 * The blocks are numbered 0-63.
* Block 3 in each sector is the Sector Trailer. See
  http://www.mouser.com/ds/2/302/MF1S503x-89574.pdf sections 8.6 and 8.7:
  * Bytes 0-5: Key A
  * Bytes 6-8: Access Bits
  * Bytes 9: User data
  * Bytes 10-15: Key B (or user data)
  * Block 0 is read-only manufacturer data.
  * To access a block, an authentication using a key from the block's sector must
    be performed first.
  * Example: To read from block 10, first authenticate using a key from sector 3
    (blocks 8-11).
  * All keys are set to FFFFFFFFh at chip delivery.
  * Warning: Please read section 8.7 "Memory Access". It includes this text: if the
    PICC detects a format violation the whole sector is irreversibly blocked.
  * To use a block in "value block" mode (for Increment/Decrement operations)
    you need to change the sector trailer. Use PICC_SetAccessBits() to calculate the bit patterns.
  * MIFARE Classic 4K (MF1S703x):
    * Has (32 sectors * 4 blocks/sector + 8 sectors * 16 blocks/sector) * 16
      bytes/block = 4096 bytes.
    * The blocks are numbered 0-255.
    * The last block in each sector is the Sector Trailer like above.
  * MIFARE Classic Mini (MF1 IC S20):
    * Has 5 sectors * 4 blocks/sector * 16 bytes/block = 320 bytes.
    * The blocks are numbered 0-19.
    * The last block in each sector is the Sector Trailer like above.
  *
  * MIFARE Ultralight (MF0ICU1):
    * Has 16 pages of 4 bytes = 64 bytes.
    * Pages 0 + 1 is used for the 7-byte UID.
    * Page 2 contains the last check digit for the UID, one byte manufacturer
      internal data, and the lock bytes (see http://www.nxp.com/documents/data_sheet/MF0ICU1.pdf section 8.5.2)
    * Page 3 is OTP, One Time Programmable bits. Once set to 1 they cannot revert
      to 0.
    * Pages 4-15 are read/write unless blocked by the lock bytes in page 2.
  * MIFARE Ultralight C (MF0ICU2):
    * Has 48 pages of 4 bytes = 192 bytes.
    * Pages 0 + 1 is used for the 7-byte UID.
    * Page 2 contains the last check digit for the UID, one byte manufacturer
      internal data, and the lock bytes (see http://www.nxp.com/documents/data_sheet/MF0ICU1.pdf section 8.5.2)
    * Page 3 is OTP, One Time Programmable bits. Once set to 1 they cannot revert
      to 0.
    * Pages 4-39 are read/write unless blocked by the lock bytes in page 2.
    * Page 40 Lock bytes
    * Page 41 16 bit one way counter
    * Pages 42-43 Authentication configuration
    * Pages 44-47 Authentication key
*/
#ifndef MFRC522_h
#define MFRC522_h

#include <Arduino.h>
#include <SPI.h>

// Firmware data for self-test
// Reference values based on firmware version
// Hint: if needed, you can remove unused self-test data to save flash memory

// Version 0.0 (0x90)
// Philips Semiconductors; Preliminary Specification Revision 2.0 - 01 August 2005; 16.1 self-test
cnst byte MFRC522_firmware_referenceV0_0[] PROGMEM = {
  0x00, 0x87, 0x98, 0x0f, 0x49, 0xFF, 0x07, 0x19,
  0xBF, 0x22, 0x30, 0x49, 0x59, 0x63, 0xAD, 0xCA,
  0x7F, 0xEE, 0x4E, 0x03, 0x5C, 0x4E, 0x49, 0x50,
  0x47, 0x9A, 0x37, 0x61, 0xE7, 0xE0, 0xC6, 0x2E,
  0x75, 0x5A, 0xED, 0x04, 0x3D, 0x02, 0x4B, 0x78,
  0x32, 0xFF, 0x58, 0x3B, 0x7C, 0xE9, 0x00, 0x94,
  0xB4, 0x4A, 0x59, 0x5B, 0xFD, 0xC9, 0x29, 0xDF,
  0x35, 0x96, 0x98, 0x9E, 0x4F, 0x32, 0x30, 0x32, 0x8D
};

// Version 1.0 (0x91)
// NXP Semiconductors; Rev. 3.8 - 17 September 2014; 16.1.1 self-test
cnst byte MFRC522_firmware_referenceV1_0[] PROGMEM = {
  0x00, 0xC6, 0x37, 0xD5, 0x32, 0xB7, 0x57, 0x5C,
  0xC2, 0xD8, 0x7C, 0x4D, 0x27, 0x89, 0x5C, 0xDE,
  0x9D, 0x3B, 0xA7, 0x00, 0x21, 0x5B, 0x89, 0x82,
  0x51, 0x3A, 0xEB, 0x02, 0x0C, 0xA5, 0x00, 0x4D,
  0x7C, 0x84, 0x4D, 0xB3, 0xCC, 0x2D, 0x1B, 0x81,
  0x5D, 0x48, 0x76, 0xD5, 0x71, 0x61, 0x21, 0xA9,
  0x86, 0x96, 0x83, 0x38, 0xCF, 0x9D, 0x5B, 0x6D,
  0xDC, 0x15, 0x5A, 0x3E, 0x7D, 0x95, 0x3B, 0x2F
};

// Version 2.0 (0x92)
// NXP Semiconductors; Rev. 3.8 - 17 September 2014; 16.1.1 self-test
cnst byte MFRC522_firmware_referenceV2_0[] PROGMEM = {
  0x00, 0xEB, 0x66, 0xBA, 0x57, 0x5C, 0x32, 0xB7, 0x57, 0x5C,
  0xC2, 0xD8, 0x7C, 0x4D, 0x27, 0x89, 0x5C, 0xDE,
  0x9D, 0x3B, 0xA7, 0x00, 0x21, 0x5B, 0x89, 0x82,
  0x51, 0x3A, 0xEB, 0x02, 0x0C, 0xA5, 0x00, 0x4D,
  0x7C, 0x84, 0x4D, 0xB3, 0xCC, 0x2D, 0x1B, 0x81,
  0x5D, 0x48, 0x76, 0xD5, 0x71, 0x61, 0x21, 0xA9,
  0x86, 0x96, 0x83, 0x38, 0xCF, 0x9D, 0x5B, 0x6D,
  0xDC, 0x15, 0x5A, 0x3E, 0x7D, 0x95, 0x3B, 0x2F
};

// Clone
// Fudan Semiconductor FM17522 (0x88)
cnst byte FM17522_firmware_reference[] PROGMEM = {
  0x00, 0x6D, 0x78, 0x8C, 0xE2, 0x8A, 0x0C, 0x18,
class MFRC522 {
public:
    // MFRC522 registers. Described in chapter 9 of the datasheet.
    // When using SPI all addresses are shifted one bit left in the "SPI address byte" (section 8.1.2.3)
    enum PCD_Register {
        // Page 0: Command and status
        CommandReg = 0x01 << 1, // starts and stops
        ComIEnReg = 0x02 << 1, // enable and disable
        DivIEnReg = 0x03 << 1, // enable and disable
        ComIrqReg = 0x04 << 1, // interrupt request bits
        DivIrqReg = 0x05 << 1, // interrupt request bits
        ErrorReg = 0x06 << 1, // error bits showing the error status of the last command executed
        Status1Reg = 0x07 << 1, // communication status
        Status2Reg = 0x08 << 1, // receiver and transmitter status bits
        FIFODataReg = 0x09 << 1, // input and output of 64 byte FIFO buffer
        FIFOLevelReg = 0x0A << 1, // number of bytes stored in the FIFO buffer
        WaterLevelReg = 0x0B << 1, // level for FIFO underflow and overflow warning
        ControlReg = 0x0C << 1, // miscellaneous control registers
        BitFramingReg = 0x0D << 1, // adjustments for bit-oriented frames
        CollReg = 0x0E << 1, // bit position of the first bit-collision detected on the RF interface
        // for future use
        0x0F // reserved
    };

    // Page 0: Command and status
    // for future use
    0x00 // reserved

    // Page 1: Command
    // for future use
    0x10 // reserved

ModeReg
modes for transmitting and receiving
= 0x11 << 1,  // defines general
data rate and framing
TxModeReg
= 0x12 << 1,  // defines transmission
RxModeReg
= 0x13 << 1,  // defines reception data
rate and framing
TxControlReg
= 0x14 << 1,  // controls the logical behavior of
the antenna driver pins TX1 and TX2
TxASKReg
= 0x15 << 1,  // controls the setting of
the transmission modulation
TxSelReg
= 0x16 << 1,  // selects the internal
sources for the antenna driver
RxSelReg
= 0x17 << 1,  // selects internal receiver
settings
RxThresholdReg
= 0x18 << 1,  // selects thresholds for the
bit decoder
DemodReg
= 0x19 << 1,  // defines demodulator
settings
//
0x1A       // reserved
for future use
//
0x1B       // reserved
for future use
MfTxReg
= 0x1C << 1,  // controls some
MIFARE communication transmit parameters
MfRxReg
= 0x1D << 1,  // controls some
MIFARE communication receive parameters
//
0x1E       // reserved
for future use
SerialSpeedReg
= 0x1F << 1,  // selects the speed of the
serial UART interface

// Page 2: Configuration
//
0x20       // reserved
for future use
CRCResultRegH
= 0x21 << 1,  // shows the MSB and
LSB values of the CRC calculation
CRCResultRegL
= 0x22 << 1,

0x23       // reserved
for future use
ModWidthReg
= 0x24 << 1,  // controls the ModWidth
setting?
//
0x25       // reserved
for future use
RFCfgReg
= 0x26 << 1,  // configures the receiver
gain
GsNReg
= 0x27 << 1,  // selects the
cconductance of the antenna driver pins TX1 and TX2 for modulation
CWGsPReg
= 0x28 << 1,  // defines the conductance
of the p-driver output during periods of no modulation
ModGsPReg = 0x29 << 1, // defines the conductance of the p-driver output during periods of modulation
TModeReg = 0x2A << 1, // defines settings for the internal timer
TPrescalerReg = 0x2B << 1, // the lower 8 bits of the TPrescaler value. The 4 high bits are in TModeReg.
TReloadRegH = 0x2C << 1, // defines the 16-bit timer reload value
TReloadRegL = 0x2D << 1,
TCounterValueRegH = 0x2E << 1, // shows the 16-bit timer value
TCounterValueRegL = 0x2F << 1,

// Page 3: Test Registers
// 0x30 // reserved
for future use
TestSel1Reg = 0x31 << 1, // general test signal configuration
TestSel2Reg = 0x32 << 1, // general test signal configuration
TestPinEnReg = 0x33 << 1, // enables pin output driver on pins D1 to D7
TestPinValueReg = 0x34 << 1, // defines the values for D1 to D7 when it is used as an I/O bus internal test bus
TestBusReg = 0x35 << 1, // shows the status of the test
AutoTestReg = 0x36 << 1, // controls the digital self-test version
VersionReg = 0x37 << 1, // shows the software and AUX2
AnalogTestReg = 0x38 << 1, // controls the pins AUX1
TestDAC1Reg = 0x39 << 1, // defines the test value for TestDAC1
TestDAC2Reg = 0x3A << 1, // defines the test value for TestDAC2
TestADCReg = 0x3B << 1 // shows the value of ADC I and Q channels
  // 0x3C // reserved
for production tests
  // 0x3D // reserved
for production tests
  // 0x3E // reserved
for production tests
  // 0x3F // reserved
for production tests
};

// MFRC522 commands. Described in chapter 10 of the datasheet.
enum PCD_Command {

PCD_Idle = 0x00, // no action,
cancels current command execution
PCD_Mem = 0x01, // stores 25
bytes into the internal buffer
PCD_GenerateRandomID = 0x02, // generates a 10-byte
random ID number
PCD_CalcCRC = 0x03, // activates
the CRC coprocessor or performs a self-test
PCD_Transmit = 0x04, // transmits data
from the FIFO buffer
PCD_NoCmdChange = 0x07, // no command
change, can be used to modify the CommandReg register bits without affecting the command,
for example, the PowerDown bit
PCD_Receive = 0x08, // activates the
receiver circuits
PCD_Transceive = 0x0C, // transmits data
from FIFO buffer to antenna and automatically activates the receiver after transmission
PCD_MFAuthent = 0x0E, // performs the
MIFARE standard authentication as a reader
PCD_SoftReset = 0x0F // resets the MFRC522
};

// MFRC522 RxGain[2:0] masks, defines the receiver's signal voltage gain factor (on
the PCD).
// Described in 9.3.3.6 / table 98 of the datasheet at

enum PCD_RxGain {
    RxGain_18dB = 0x00 << 4, // 000b - 18 dB, minimum
    RxGain_23dB = 0x01 << 4, // 001b - 23 dB
    RxGain_18dB_2 = 0x02 << 4, // 010b - 18 dB, it seems
    RxGain_23dB_2 = 0x03 << 4, // 011b - 23 dB, it seems
    RxGain_33dB = 0x04 << 4, // 100b - 33 dB, average,
    RxGain_38dB = 0x05 << 4, // 101b - 38 dB
    RxGain_43dB = 0x06 << 4, // 110b - 43 dB
    RxGain_48dB = 0x07 << 4, // 111b - 48 dB, maximum
    RxGain_min = 0x00 << 4, // 000b - 18 dB, minimum,
    RxGain_avg = 0x04 << 4, // 100b - 33 dB, average,
    RxGain_max = 0x07 << 4 // 111b - 48 dB,
    RxGain_min
}
// Commands sent to the PICC.
enum PICC_Command {
    // The commands used by the PCD to manage communication with several
    PICCs (ISO 14443-3, Type A, section 6.4)
PICC_CMD_REQA = 0x26,  // REQuest command, Type A. Invites PICCs in state IDLE to go to READY and prepare for anticollision or selection. 7 bit frame.

PICC_CMD_WUPA = 0x52,  // Wake-UP command, Type A. Invites PICCs in state IDLE and HALT to go to READY(*) and prepare for anticollision or selection. 7 bit frame.

PICC_CMD_CT = 0x88,  // Cascade Tag. Not really a command, but used during anti collision.

PICC_CMD_SEL_CL1 = 0x93,  // Anti collision/Select, Cascade Level 1
PICC_CMD_SEL_CL2 = 0x95,  // Anti collision/Select, Cascade Level 2
PICC_CMD_SEL_CL3 = 0x97,  // Anti collision/Select, Cascade Level 3
PICC_CMD_HLTA = 0x50,  // HaLT command, Type A. Instructs an ACTIVE PICC to go to state HALT.

The commands used for MIFARE Classic (from http://www.mouser.com/ds/2/302/MF1S503x-89574.pdf, Section 9)

Use PCD_MFAuthent to authenticate access to a sector, then use these commands to read/write/modify the blocks on the sector.

PICC_CMD_MF_AUTH_KEY_A = 0x60,  // Perform authentication with Key A
PICC_CMD_MF_AUTH_KEY_B = 0x61,  // Perform authentication with Key B
PICC_CMD_MF_READ = 0x30,  // Reads one 16 byte block from the authenticated sector of the PICC. Also used for MIFARE Ultralight.
PICC_CMD_MF_WRITE = 0xA0,  // Writes one 16 byte block to the authenticated sector of the PICC. Called "COMPATIBILITY WRITE" for MIFARE Ultralight.

PICC_CMD_MF_DECREMENT = 0xC0,  // Decrements the contents of a block and stores the result in the internal data register.
PICC_CMD_MF_INCREMENT = 0xC1,  // Increments the contents of a block and stores the result in the internal data register.
PICC_CMD_MF_RESTORE = 0xC2,  // Reads the contents of a block into the internal data register.
PICC_CMD_MF_TRANSFER = 0xB0,  // Writes the contents of the internal data register to a block.

The commands used for MIFARE Ultralight (from http://www.nxp.com/documents/data_sheet/MF0ICU1.pdf, Section 8.6)

The PICC_CMD_MF_READ and PICC_CMD_MF_WRITE can also be used for MIFARE Ultralight.

PICC_CMD_UL_WRITE = 0xA2  // Writes one 4 byte page to the PICC.

// MIFARE constants that does not fit anywhere else
enum MIFARE_Misc {
    MF_ACK = 0xA,  // The MIFARE Classic uses a 4 bit ACK/NAK. Any other value than 0xA is NAK.
MF_KEY_SIZE = 6 // A Mifare Crypto1 key is 6 bytes.

// PICC types we can detect. Remember to update PICC_GetTypeName() if you add more.
// last value set to 0xff, then compiler uses less ram, it seems some optimisations are triggered
enum PICC_Type : byte {
    PICC_TYPE_UNKNOWN,
    PICC_TYPE_ISO_14443_4, // PICC compliant with ISO/IEC 14443-4
    PICC_TYPE_ISO_18092, // PICC compliant with ISO/IEC 18092 (NFC)
    PICC_TYPE_MIFARE_MINI, // MIFARE Classic protocol, 320 bytes
    PICC_TYPE_MIFARE_1K, // MIFARE Classic protocol, 1KB
    PICC_TYPE_MIFARE_4K, // MIFARE Classic protocol, 4KB
    PICC_TYPE_MIFARE_UL, // MIFARE Ultralight or Ultralight C
    PICC_TYPE_MIFARE_PLUS, // MIFARE Plus
    PICC_TYPE_TNP3XXX, // Only mentioned in NXP AN 10833 MIFARE Type Identification Procedure
    PICC_TYPE_NOT_COMPLETE = 0xff // SAK indicates UID is not complete.
};

// Return codes from the functions in this class. Remember to update GetStatusCodeName() if you add more.
// last value set to 0xff, then compiler uses less ram, it seems some optimisations are triggered
enum StatusCode : byte {
    STATUS_OK, // Success
    STATUS_ERROR, // Error in communication
    STATUS_COLLISION, // Collision detected
    STATUS_TIMEOUT, // Timeout in communication.
    STATUS_NO_ROOM, // A buffer is not big enough.
    STATUS_INTERNAL_ERROR, // Internal error in the code.
    STATUS_INVALID, // Invalid argument.
    STATUS_CRC_WRONG, // The CRC_A does not match
    STATUS_MIFARE_NACK = 0xff // A MIFARE PICC responded with NAK.
};

typedef struct {
    byte size; // Number of bytes in the UID. 4, 7 or 10.
    byte uidByte[10];
}
byte sak; // The SAK (Select acknowledge) byte returned from the PICC after successful selection.
} Uid;

// A struct used for passing a MIFARE Crypto1 key
typedef struct {
    byte keyByte[MF_KEY_SIZE];
} MIFARE_Key;

// Member variables
Uid uid; // Used by PICC_ReadCardSerial().

// Size of the MFRC522 FIFO
static const byte FIFO_SIZE = 64; // The FIFO is 64 bytes.

// Functions for setting up the Arduino
MFRC522();
MFRC522(byte resetPowerDownPin);
MFRC522(byte chipSelectPin, byte resetPowerDownPin);

// Basic interface functions for communicating with the MFRC522
void PCD_WriteRegister(byte reg, byte value);
void PCD_WriteRegister(byte reg, byte count, byte *values);
byte PCD_ReadRegister(byte reg);
void PCD_ReadRegister(byte reg, byte count, byte *values, byte rxAlign = 0);
void setBitMask(unsigned char reg, unsigned char mask);
void PCD_SetRegisterBitMask(byte reg, byte mask);
void PCD_ClearRegisterBitMask(byte reg, byte mask);
StatusCode PCD_CalculateCRC(byte *data, byte length, byte *result);

// Functions for manipulating the MFRC522
void PCD_Init();
void PCD_Init(byte resetPowerDownPin);
void PCD_Init(byte chipSelectPin, byte resetPowerDownPin);
void PCD_Reset();
void PCD_AntennaOn();
void PCD_AntennaOff();
byte PCD_GetAntennaGain();
void PCD_SetAntennaGain(byte mask);
bool PCD_PerformSelfTest();

// Functions for communicating with PICCs
StatusCode PCD_TransceiveData(byte *sendData, byte sendLen, byte *backData, byte *backLen, byte *validBits = NULL, byte rxAlign = 0, bool checkCRC = false);

StatusCode PCD_CommunicateWithPICC(byte command, byte waitIRq, byte *sendData, byte sendLen, byte *backData = NULL, byte *validBits = NULL, byte rxAlign = 0, bool checkCRC = false);

StatusCode PICC_RequestA(byte *bufferATQA, byte *bufferSize);

StatusCode PICC_WakeupA(byte *bufferATQA, byte *bufferSize);

StatusCode PICC_REQA_or_WUPA(byte command, byte *bufferATQA, byte *bufferSize);

StatusCode PICC_Select(Uid *uid, byte validBits = 0);

StatusCode PICC_HaltA();

///////////////////////////////////////////////////////////////////////
// Functions for communicating with MIFARE PICCs
///////////////////////////////////////////////////////////////////////
StatusCode PCD_Authenticate(byte command, byte blockAddr, MIFARE_Key *key, Uid *uid);

void PCD_StopCrypto1();

StatusCode MIFARE_Read(byte blockAddr, byte *buffer, byte *bufferSize);

StatusCode MIFARE_Write(byte blockAddr, byte *buffer, byte bufferSize);

StatusCode MIFARE_Ultralight_Write(byte page, byte *buffer, byte bufferSize);

StatusCode MIFARE_Decrement(byte blockAddr, int32_t delta);

StatusCode MIFARE_Increment(byte blockAddr, int32_t delta);

StatusCode MIFARE_Restore(byte blockAddr);

StatusCode MIFARE_Transfer(byte blockAddr);

StatusCode MIFARE_GetValue(byte blockAddr, int32_t *value);

StatusCode MIFARE_SetValue(byte blockAddr, int32_t value);

StatusCode PCD_NTAG216_AUTH(byte *passWord, byte pACK[]);

///////////////////////////////////////////////////////////////////////
// Support functions
///////////////////////////////////////////////////////////////////////
StatusCode PCD_MIFARE_Transceive(byte *sendData, byte sendLen, bool acceptTimeout = false);

// old function used too much memory, now name moved to flash; if you need char, copy from flash to memory
//const char *GetStatusCodeName(byte code);
static const __FlashStringHelper *GetStatusCodeName(StatusCode code);
static PICC_Type PICC_GetType(byte sak);

// old function used too much memory, now name moved to flash; if you need char, copy from flash to memory
//const char *PICC_GetTypeName(byte type);
static const __FlashStringHelper *PICC_GetTypeName(PICC_Type type);

// Support functions for debugging
void PCD_DumpVersionToSerial();
void PICC_DumpToSerial(Uid *uid);
void PICC_DumpDetailsToSerial(Uid *uid);
void PICC_DumpMifareClassicToSerial(Uid *uid, PICC_Type piccType, MIFARE_Key *key);
void PICC_DumpMifareClassicSectorToSerial(Uid *uid, MIFARE_Key *key, byte sector);
void PICC_DumpMifareUltralightToSerial();

// Advanced functions for MIFARE
void MIFARE_SetAccessBits(byte *accessBitBuffer, byte g0, byte g1, byte g2, byte g3);
bool MIFARE_OpenUidBackdoor(bool logErrors);
bool MIFARE_SetUid(byte *newUid, byte uidSize, bool logErrors);
bool MIFARE_UnbrickUidSector(bool logErrors);

/////////////////////////////////////////////////////////////////////////////////////
// Convenience functions - does not add extra functionality
/////////////////////////////////////////////////////////////////////////////////////
bool PICC_IsNewCardPresent();
bool PICC_ReadCardSerial();

private:
  byte _chipSelectPin;     // Arduino pin connected to MFRC522's SPI slave select input (Pin 24, NSS, active low)
  byte _resetPowerDownPin; // Arduino pin connected to MFRC522's reset and power down input (Pin 6, NRSTPD, active low)
  StatusCode MIFARE_TwoStepHelper(byte command, byte blockAddr, int32_t data);
};
#endif

LCD.h

//------------------------------------------------------------------------------
// Created by Francisco Malpartida on 20/08/11.
// Copyright 2011 - Under creative commons license 3.0:
// Attribution-ShareAlike CC BY-SA
//
// This software is furnished "as is", without technical support, and with no
// warranty, express or implied, as to its usefulness for any purpose.
//
// Thread Safe: No
// Extendable: Yes
//
// @file LCD.h
// This file implements a basic liquid crystal library that comes as standard
// in the Arduino SDK.
//
// @brief
// This is a basic implementation of the LiquidCrystal library of the
// Arduino SDK. This library is a refactored version of the one supplied
// in the Arduino SDK in such a way that it simplifies its extension
to support other mechanism to communicate to LCDs such as I2C, Serial, SR,
The original library has been reworked in such a way that this will be
the base class implementing all generic methods to command an LCD based
on the Hitachi HD44780 and compatible chipsets.

This base class is a pure abstract class and needs to be extended. As reference,
it has been extended to drive 4 and 8 bit mode control, LCDs and I2C extension
backpacks such as the I2CLCDextraIO using the PCF8574* I2C IO Expander ASIC.

The functionality provided by this class and its base class is identical
to the original functionality of the Arduino LiquidCrystal library.

@version API 1.1.0

@author F. Malpartida - fmalpartida@gmail.com

#ifndef _LCD_H_
define _LCD_H_

#if (ARDUINO < 100)
#include <WProgram.h>
#else
#include <Arduino.h>
#endif

#ifdef __AVR__
#include <avr/pgmspace.h>
#endif

#include <inttypes.h>
#include <Print.h>

/*@defined
@abstract Performs a bitwise shift.
@discussion Defines _BV bit shift which is very dependent macro defined by Atmel.

\note The bit shift is performed by the compiler which then inserts the result into the code. Thus, there is no run-time overhead when using _BV().
*/
#endif
#define _BV(bit) (1 << (bit))

/*@defined
@abstract Enables disables fast waits for write operations for LCD
@discussion If defined, the library will avoid doing un-necessary waits. this can be done, because the time taken by Arduino's slow digitalWrite operations. If fast digitalIO operations, comment this line out or undefine the mode.
*/
#endif __AVR__
#define FAST_MODE
#endif

/*@function
@abstract waits for a given time in microseconds (compilation dependent).
@discussion Waits for a given time defined in microseconds depending on the FAST_MODE define. If the FAST_MODE is defined the call will return immeditelly.
@param uSec[in] time in microseconds.
@result None
/*/ inline static void waitUsec ( uint16_t uSec )
{
#ifndef FAST_MODE
    delayMicroseconds ( uSec );
#endif // FAST_MODE
}

/*@defined
@abstract All these definitions shouldn't be used unless you are writing a driver.
@discussion All these definitions are for driver implementation only and shouldn't be used by applications.
*/

// LCD Commands
// ---------------------------------------------------------------------------------------------------------------
#define LCD_CLEARDISPLAY 0x01
#define LCD_RETURNHOME 0x02
#define LCD_ENTRYMODESET 0x04
#define LCD_DISPLAYCONTROLSHIFT 0x08
#define LCD_CURSORSHIFT 0x10
#define LCD_FUNCTIONSET 0x20
#define LCD_SETCGRAMADDR 0x40
#define LCD_SETDDRAMADDR 0x80

// flags for display entry mode
// ---------------------------------------------------------------------------------------------------------------
#define LCD_ENTRYRIGHT 0x00
#define LCD_ENTRYLEFT 0x02
#define LCD_ENTRYSHIFTINCREMENT 0x01
#define LCD_ENTRYSHIFTDECREMENT 0x00

// flags for display on/off and cursor control
// ------------------------------------------------------------------------------------
#define LCD_DISPLAYON 0x04
#define LCD_DISPLAYOFF 0x00
#define LCD_CURSoron 0x02
#define LCD_CURSorOFF 0x00
#define LCD_BLINKon 0x01
#define LCD_BLINKOFF 0x00

// flags for display/cursor shift
// -----------------------------------------------------------------------------------
#define LCD_DISPLAYMOVE 0x08
#define LCD_CURSormove 0x00
#define LCD_MOVERIGHT 0x04
#define LCD_MOVELEFT 0x00

// flags for function set
// -----------------------
#define LCD_8BITMODE 0x10
#define LCD_4BITMODE 0x00
#define LCD_2LINE 0x08
#define LCD_1LINE 0x00
#define LCD_5x10DOTS 0x04
#define LCD_5x8DOTS 0x00

// Define COMMAND and DATA LCD Rs (used by send method).
// -------------------------------------------------------------------
#define COMMAND 0
#define DATA 1
#define FOUR_BITS 2

/*! @defined @abstract Defines the duration of the home and clear commands @discussion This constant defines the time it takes for the home and clear commands in the LCD - Time in microseconds. */
#define HOME_CLEAR_EXEC 2000

/*! @defined @abstract Backlight off constant declaration @discussion Used in combination with the setBacklight to switch off the LCD backlight. */
#define BACKLIGHT_OFF 0
#define BACKLIGHT_ON 255

typedef enum { POSITIVE, NEGATIVE } t_backlighPol;

class LCD : public Print {

public:

virtual void begin(uint8_t cols, uint8_t rows, uint8_t charsize = LCD_5x8DOTS);

};

virtual void begin(uint8_t cols, uint8_t rows, uint8_t charsize = LCD_5x8DOTS);
corner.

This operation is time consuming for the LCD.

```c
@param none
*/
void clear();
```

```c
/*@function
@abstract Sets the cursor to the upper-left corner.
@discussion Positions the cursor in the upper-left of the LCD.
That is, use that location in outputting subsequent text to the display.
To also clear the display, use the clear() function instead.

This operation is time consuming for the LCD.

```c
@param none
*/
void home();
```

```c
/*@function
@abstract Turns off the LCD display.
@discussion Turns off the LCD display, without losing the text currently
being displayed on it.

```c
@param none
*/
void noDisplay();
```

```c
/*@function
@abstract Turns on the LCD display.
@discussion Turns on the LCD display, after it's been turned off with
noDisplay(). This will restore the text (and cursor location) that was on
the display prior to calling noDisplay().

```c
@param none
*/
void display();
```

```c
/*@function
@abstract Turns off the blinking of the LCD cursor.

```c
@param none
*/
void noBlink();
```
/*! 
 * @function
 * @abstract Display the cursor of the LCD.
 * @discussion Display the blinking LCD cursor. If used in combination with
 * the cursor() function, the result will depend on the particular display.
 * 
 * @param none
 * */
 void blink();

/*! 
 * @function
 * @abstract Hides the LCD cursor.
 * @param none
 * */
 void noCursor();

/*! 
 * @function
 * @abstract Display the LCD cursor.
 * @discussion Display the LCD cursor: an underscore (line) at the location
 * where the next character will be written.
 * 
 * @param none
 * */
 void cursor();

/*! 
 * @function
 * @abstract Scrolls the contents of the display (text and cursor) one space
 * to the left.
 * 
 * @param none
 * */
 void scrollDisplayLeft();

/*! 
 * @function
 * @abstract Scrolls the contents of the display (text and cursor) one space
 * to the right.
 * 
 * @param none
 * */
 void scrollDisplayRight();

/*! 
 * @function
 * @abstract Set the direction for text written to the LCD to left-to-right.
 * @discussion Set the direction for text written to the LCD to left-to-right.
 */
All subsequent characters written to the display will go from left to right, but does not affect previously-output text.

This is the default configuration.

```c
@param none
*/
void leftToRight();

/*@!
@function
@abstract Set the direction for text written to the LCD to right-to-left.
@discussion Set the direction for text written to the LCD to right-to-left. All subsequent characters written to the display will go from right to left, but does not affect previously-output text.

left-to-right is the default configuration.

@param none
*/
void rightToLeft();

/*@!
@function
@abstract Moves the cursor one space to the left.
@discussion
@param none
*/
void moveCursorLeft();

/*@!
@function
@abstract Moves the cursor one space to the right.

@param none
*/
void moveCursorRight();

/*@!
@function
@abstract Turns on automatic scrolling of the LCD.
@discussion Turns on automatic scrolling of the LCD. This causes each character output to the display to push previous characters over by one space. If the current text direction is left-to-right (the default), the display scrolls to the left; if the current direction is right-to-left, the display scrolls to the right. This has the effect of outputting each new character to the same location on the LCD.
```
@param none
*/
void autoscroll();

/*!
@function
@abstract Turns off automatic scrolling of the LCD.
@discussion Turns off automatic scrolling of the LCD, this is the default
configuration of the LCD.

@param none
*/
void noAutoscroll();

/*!
@function
@abstract Creates a custom character for use on the LCD.
@discussion Create a custom character (glyph) for use on the LCD.
Most chipsets only support up to eight characters of 5x8 pixels. Therefore,
this methods has been limited to locations (numbered 0 to 7).

The appearance of each custom character is specified by an array of eight
bytes, one for each row. The five least significant bits of each byte
determine the pixels in that row. To display a custom character on screen,
write()/print() its number, i.e. lcd.print(char(x)); // Where x is 0..7.

@param location[in] LCD memory location of the character to create
(0 to 7)
@param charmap[in] the bitmap array representing each row of the character.
*/
void createChar(uint8_t location, uint8_t charmap[]);

#ifndef __AVR__
/*!
@function
@abstract Creates a custom character for use on the LCD.
@discussion Create a custom character (glyph) for use on the LCD.
Most chipsets only support up to eight characters of 5x8 pixels. Therefore,
this methods has been limited to locations (numbered 0 to 7).

The appearance of each custom character is specified by an array of eight
bytes, one for each row. The five least significant bits of each byte
determine the pixels in that row. To display a custom character on screen,
write()/print() its number, i.e. lcd.print(char(x)); // Where x is 0..7.

This method take the character defined in program memory.

@param location[in] LCD memory location of the character to create
(0 to 7)
@param charmap[in] the bitmap array representing each row of the character.
*/

#endif

Usage for flash defined characters:
const char str_pstr[] PROGMEM = {0xc, 0x12, 0x12, 0xc, 0, 0, 0, 0};
*/
void createChar(uint8_t location, const char *charmap);
#endif // __AVR__

/*! 
 @function
 @abstract   Position the LCD cursor.
 @discussion Sets the position of the LCD cursor. Set the location at which subsequent text written to the LCD will be displayed.

 @param      col[in] LCD column
 @param      row[in] LCD row - line.
 */
void setCursor(uint8_t col, uint8_t row);

/*! 
 @function
 @abstract   Switch-on the LCD backlight.
 @discussion Switch-on the LCD backlight. The setBacklightPin has to be called before setting the backlight for this method to work. @see setBacklightPin.
 */
void backlight ( void );

/*! 
 @function
 @abstract   Switch-off the LCD backlight.
 @discussion Switch-off the LCD backlight. The setBacklightPin has to be called before setting the backlight for this method to work. @see setBacklightPin.
 */
void noBacklight ( void );

/*! 
 @function
 @abstract   Switch on the LCD module.
 @discussion Switch on the LCD module, it will switch on the LCD controller and the backlight. This method has the same effect of calling display and backlight. @see display, @see backlight
 */
void on ( void );

/*! 
 @function
 @abstract   Switch off the LCD module.
 @discussion Switch off the LCD module, it will switch off the LCD controller and the backlight. This method has the same effect of calling noDisplay and noBacklight. @see display, @see backlight
 */
void off ( void );

virtual void setBacklightPin ( uint8_t value, t_backlighPol pol ) { };

virtual void setBacklight ( uint8_t value ) { };

virtual void write(uint8_t value);

#if (ARDUINO < 100)
    virtual void write(uint8_t value);
#endif
#else
   virtual size_t write(uint8_t value);
#endif

#if (ARDUINO < 100)
   using Print::write;
#else
   using Print::write;
#endif

protected:
   // Internal LCD variables to control the LCD shared between all derived
   // classes.
   uint8_t _displayfunction;  // LCD_5x10DOTS or LCD_5x8DOTS, LCD_4BITMODE or
   // LCD_8BITMODE, LCD_1LINE or LCD_2LINE
   uint8_t _displaycontrol;   // LCD base control command LCD on/off, blink, cursor
   // all commands are "ored" to its contents.
   uint8_t _displaymode;      // Text entry mode to the LCD
   uint8_t _numlines;         // Number of lines of the LCD, initialized with begin()
   uint8_t _cols;             // Number of columns in the LCD
   t_backlightPol _polarity;   // Backlight polarity

private:
   /*! *
      @function
      @abstract   Send a command to the LCD.
      @discussion This method sends a command to the LCD by setting the Register
      select line of the LCD.

      This command shouldn't be used to drive the LCD, only to implement any other
      feature that is not available on this library.

      @param      value[in] Command value to send to the LCD (COMMAND, DATA or
      FOUR_BITS).
      */
   void command(uint8_t value);

   /*! *
      @function
      @abstract   Send a particular value to the LCD.
      @discussion Sends a particular value to the LCD. This is a pure abstract
      method, therefore, it is implementation dependent of each derived class how
      to physically write to the LCD.

      Users should never call this method.

      @param      value[in] Value to send to the LCD.
      @result     mode LOW - write to the LCD CGRAM, HIGH - write a command to
      the LCD.
      */
#if (ARDUINO < 100)
   virtual void send(uint8_t value, uint8_t mode) { }
#else
   virtual void send(uint8_t value, uint8_t mode) = 0;
#endif

};

#else

// Created by Francisco Malpartida on 20/08/11.
// Copyright 2011 - Under creative commons license 3.0:
// Attribution-ShareAlike CC BY-SA

// This software is furnished "as is", without technical support, and with no warranty, express or implied, as to its usefulness for any purpose.

// Thread Safe: No
// Extendable: Yes

//@file LiquidCrystal_I2C.h
// This file implements a basic liquid crystal library that comes as standard in the Arduino SDK but using an I2C IO extension board.

//@brief
// This is a basic implementation of the LiquidCrystal library of the Arduino SDK. The original library has been reworked in such a way that this class implements the all methods to command an LCD based on the Hitachi HD44780 and compatible chipsets using I2C extension backpacks such as the I2CLCDextraIO with the PCF8574* I2C IO Expander ASIC.

// The functionality provided by this class and its base class is identical to the original functionality of the Arduino LiquidCrystal library.

//@author F. Malpartida - fmalpartida@gmail.com

#ifndef LiquidCrystal_I2C_h
#define LiquidCrystal_I2C_h
#include <inttypes.h>
#include <Print.h>
#include "I2CIO.h"
#include "LCD.h"

class LiquidCrystal_I2C : public LCD
{
 public:
/*!
@method   Class constructor.
@abstract  Initializes class variables and defines the I2C address of the
LCD. The constructor does not initialize the LCD.
@param     lcd_Addr[in] I2C address of the IO expansion module. For I2CLCDextraIO,
the address can be configured using the on board jumpers.
*/
LiquidCrystal_I2C (uint8_t lcd_Addr);
// Constructor with backlight control
LiquidCrystal_I2C (uint8_t lcd_Addr, uint8_t backlighPin, t_backlighPol pol);

/*!
@method   Class constructor.
@abstract  Initializes class variables and defines the I2C address of the
LCD. The constructor does not initialize the LCD.
@param     lcd_Addr[in] I2C address of the IO expansion module. For I2CLCDextraIO,
the address can be configured using the on board jumpers.
@param     En[in] LCD En (Enable) pin connected to the IO extender module
@param     Rw[in] LCD Rw (Read/write) pin connected to the IO extender module
@param     Rs[in] LCD Rs (Reset) pin connected to the IO extender module
*/
LiquidCrystal_I2C( uint8_t lcd_Addr, uint8_t En, uint8_t Rw, uint8_t Rs);
// Constructor with backlight control
LiquidCrystal_I2C( uint8_t lcd_Addr, uint8_t En, uint8_t Rw, uint8_t Rs,
     uint8_t backlighPin, t_backlighPol pol);

/*!
@method   Class constructor.
@abstract  Initializes class variables and defines the I2C address of the
LCD. The constructor does not initialize the LCD.
@param     lcd_Addr[in] I2C address of the IO expansion module. For I2CLCDextraIO,
the address can be configured using the on board jumpers.
@param     En[in] LCD En (Enable) pin connected to the IO extender module
@param     Rw[in] LCD Rw (Read/write) pin connected to the IO extender module
@param     Rs[in] LCD Rs (Reset) pin connected to the IO extender module
@param     d4[in] LCD data 0 pin map on IO extender module
@param     d5[in] LCD data 1 pin map on IO extender module
@param     d6[in] LCD data 2 pin map on IO extender module
@param     d7[in] LCD data 3 pin map on IO extender module
*/
LiquidCrystal_I2C( uint8_t lcd_Addr, uint8_t En, uint8_t Rw, uint8_t Rs,
     uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7 );
// Constructor with backlight control
LiquidCrystal_I2C(uint8_t lcd_Addr, uint8_t En, uint8_t Rw, uint8_t Rs,
     uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7,
     uint8_t backlighPin, t_backlighPol pol);

/*! 
@function
@abstract LCD initialization and associated HW.
@discussion Initializes the LCD to a given size (col, row). This methods initializes the LCD, therefore, it MUST be called prior to using any other method from this class or parent class.

The begin method can be overloaded if necessary to initialize any HW that is implemented by a library and can't be done during construction, here we use the Wire class.

@param cols[in] the number of columns that the display has
@param rows[in] the number of rows that the display has
@param charsize[in] size of the characters of the LCD: LCD_5x8DOTS or LCD_5x10DOTS.
*/
virtual void begin(uint8_t cols, uint8_t rows, uint8_t charsize = LCD_5x8DOTS);

/*! 
@function
@abstract Send a particular value to the LCD.
@discussion Sends a particular value to the LCD for writing to the LCD or as an LCD command.

@param value[in] Value to send to the LCD.
@param mode[in] DATA - write to the LCD CGRAM, COMMAND - write a command to the LCD.
*/
virtual void send(uint8_t value, uint8_t mode);

/*! 
@function
@abstract Sets the pin to control the backlight.
@discussion Sets the pin in the device to control the backlight. This device doesn't support dimming backlight capability.

@param 0: backlight off, 1..255: backlight on.
*/
void setBacklightPin ( uint8_t value, t_backlighPol pol );

/*! 
@function
@abstract Switch-on/off the LCD backlight.
@discussion Switch-on/off the LCD backlight.
The setBacklightPin has to be called before setting the backlight for
this method to work. @see setBacklightPin.

  @param value: backlight mode (HIGH|LOW)
  */
  void setBacklight ( uint8_t value );

private:

  /*! 
     @method 
     @abstract Initializes the LCD class 
     @discussion Initializes the LCD class and IO expansion module. 
   */ 
  int init();

  /*! 
     @function 
     @abstract Initialises class private variables 
     @discussion This is the class single point for initialising private variables. 
   */ 
  void config (uint8_t lcd_Addr, uint8_t En, uint8_t Rw, uint8_t Rs, 
               uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7 );

  /*! 
     @method 
     @abstract Writes an 4 bit value to the LCD. 
     @discussion Writes 4 bits (the least significant) to the LCD control data lines. 
     @param value[in] Value to write to the LCD 
     @param more[in] Value to distinguish between command and data. 
     COMMAND == command, DATA == data. 
   */ 
  void write4bits(uint8_t value, uint8_t mode);

  /*! 
     @method 
     @abstract Pulse the LCD enable line (En). 
     @discussion Sends a pulse of 1 uS to the Enable pin to execute an command or write operation. 
   */ 
  void pulseEnable(uint8_t);
21.3 Questionnaire Results

**RFID usage awareness**

1. Do you know what RFID is? (If Yes, please elaborate) *
   
   Number of participants: 39
   
   - I've heard of it but only recently and when it's about protecting contactless bank cards.
   - No
   - Yes
   - I do not.
- Something that can be scanned to obtain personal info
- Contactless identification, but only got that from the questions asked.
- Yes, something to do with fobs/touch cards.
  - No
  - No
  - No
  - No
  - No
  - No
  - No
  - No
  - The square tags on some consumer goods
  - No not really
  - No
  - No
  - Erm not sure
  - No.
  - Just that there is a risk that people can read and use your credit cards without your knowledge
- Yes. A sort of smart tag, that reads information on an object.

2. Do you use any form of RFID? *

Number of participants: 40

28 (70.0%): 0-1 times a day
10 (25.0%): 2-5 times a day

(0.0%): 6-9 times a day

2 (5.0%): 10+ times a day
3. Do you use Apple Pay® or Android Pay®?

Number of participants: 40

17 (42.5%): yes
23 (57.5%): no

4. Do you own any of the following?*

Number of participants: 40

35 (87.5%): Contactless bank Card
9 (22.5%): Oyster card
5 (12.5%): Iff card
15 (37.5%): Card to gain access to buildings or facilities
2 (5.0%): Other

Answer(s) from the additional field:
- Sign in card
- Don't own any just a debit card

5. Do you know any risks associated with contactless payment methods?*

Number of participants: 40

19 (47.5%): no
21 (52.5%): Other
Answer(s) from the additional field:

- Anyone can use them if they get hold of your card
- Fraud
- With RFID bank cards people can simply pass a reader by and take small payments. Also with a simple reader you can take the details from cards.
- It could possibly be scared by fraudsters using special equipment
- Fraudulent transactions
- If it gets stolen
- Anyone can use it up to £30 if they get hold of it or pick it up
- Security/Identity protection
- Fraud with loss of card.
- Fraud
- Fraudulent charges
- Theft from contactless machines pressed to bags, easier to access account if card is lost
- Identity theft. Money theft
- Scammers have supposedly created a device where they all they need to do is walk past you and scan your bank card. Apparently. Other than that only if any of the cards are stolen it lets anyone use.

- Losing the card gives someone access to a certain amount of money per day.
- Security reasons
- Fraud/Theft. Chicken
- If card gets stolen
- Security but not health risks
- People accessing your card without your knowledge, through a wallet or purse
- Readers can be used to gain information from you bank card.
6. Does your place of work use any form of RFID systems? *
   Number of participants: 40
   15 (37.5%): yes
   25 (62.5%): no