Imperial College London

EE2-PRJ Year Two Project

Group 12
Knock Pattern Recognition Device
Final Report

Constantinos Nicolaou – 00953169

James Arnold – 01071561

Kantapat Tangpoonsinthana – 01096788

Loizos Papachristoforou – 00941676

Lok Hin Wan – 01059636

Zifan Wang – 01077639

Zhongxuan Li – 01050018

Supervisor – Dr. Krystian Mikolajczyk

13th March 2017

Table of Contents

1 Abstract

From those suffering from dexterity problems to someone going to a swimming pool, the KnockLock provides them a convenient yet secure method of locking their doors or belongings. This report furthers the knocking pattern recognition idea from the previous interim report. It describes the technical details of the concept in depth, before carefully analysing and evaluating the product's commercial feasibility and technical viability. The conclusion is positive on both aspects, and thus a prototype is being built to examine the KnockLock's physical implementation, and for demonstration purposes to its potential investors.

2 Introduction

2.1 Technical Problem to be solved

Until now the humble mechanical lock and key has reigned supreme in the domestic environment. However, as technology has developed, consumers now have an unprecedented capacity and desire to conserve energy, reduce domestic drudgery and optimise their living environment. As such, smart devices are now one of the fastest growing sector in the market.

At the same time, the group observed that dexterity impairment is a serious issue, being ranked as the third-most prevalent disability worldwide. Specifically, in the UK, about 3.4 million people, or 28% of all disabled people suffer from a sort of dexterity problem [1]. Although the public have kept calling for action to aid those with dexterity problems, there are few products designed to assist them.

Apart from those suffering from dexterity problems, there are many other scenarios where people may have difficulties with opening a door. For example, someone might want to open the door while carrying a few shopping bags without much struggles.

Therefore, the group believes that people are in need of a more convenient and intuitive lock. Being inspired by these social problems, this project is set to develop a next generation locking mechanism, which not only provides care to people with dexterity issues, but also widely serves ordinary people in their daily lives.

2.2 Social Context

Further detailed investigation produced some interesting results that relates the age of people with the incidence of dexterity impairments.

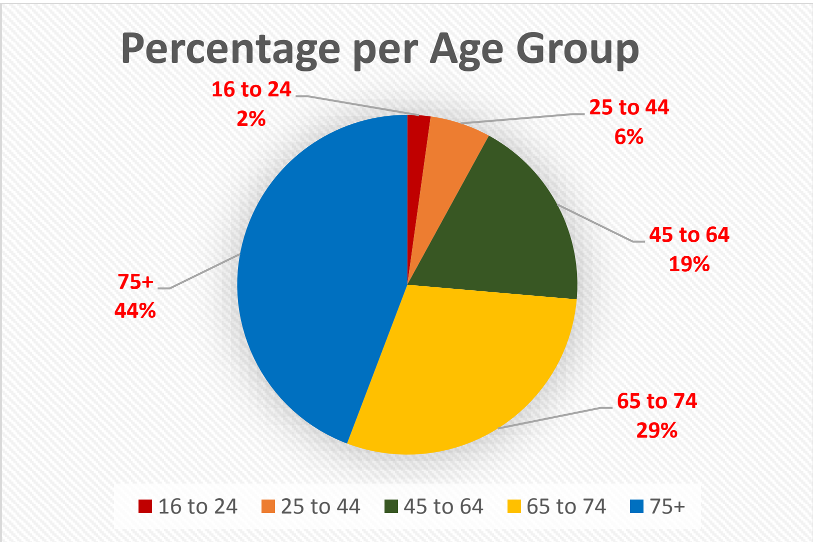

Figure 1 Dexterity Problems in Different Age Groups

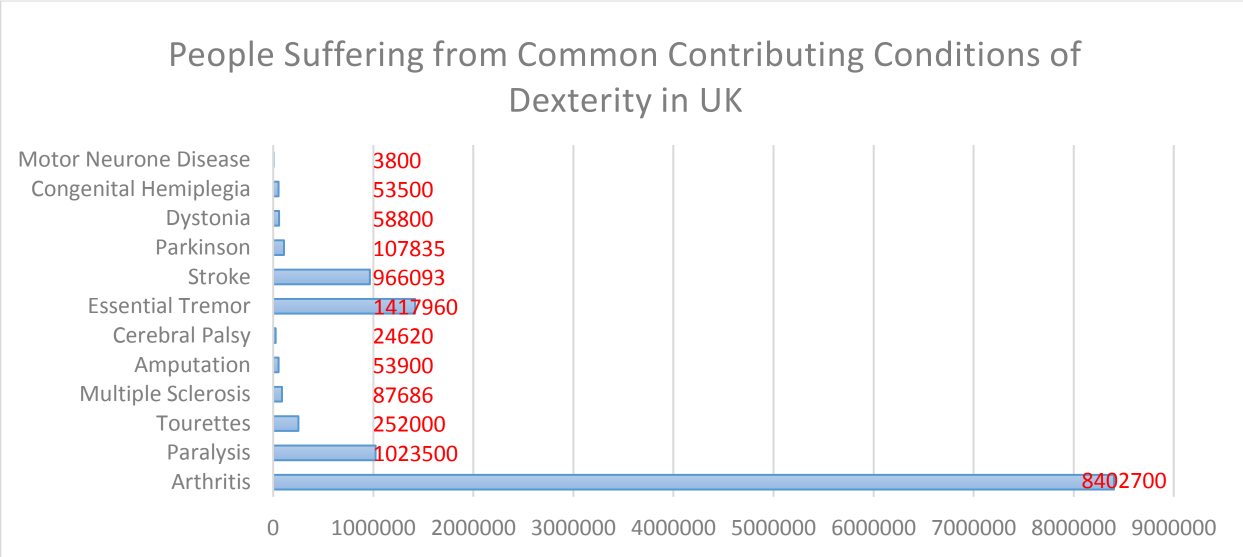

Figure 2 Common Contribution Conditions related with Dexterity Problems in UK

From Figure 1, it can be observed that elderly people are more likely to suffer from dexterity problems, as almost 3 out of 4 sufferers are aged 65 or over. This is because dexterity impairments result from several conditions that mainly occur in old age. Figure 2 provides data of the number of people that suffer from such conditions in UK. Clearly, arthritis is the main contributing condition since more than 8 million people suffer from a form of it. Moreover, approximately 3.5 million people have been subjected to a kind of stroke, experienced essential tremor and/or are partially or fully paralysed. A high percentage of those people are prone to dexterity problems and therefore the impact of those conditions is significant and of high relevancy.

Dealing with the challenges faced by the disabled population remains one of the greatest and most pressing challenges in the current time. There are over 10 million disabled people in Britain, comprising 15% of the population [1]. The effects are devastating. Quality of life can be greatly affected, with only half of disabled people of working age in work. [2] The effects can be particularly pronounced for those with motor impairments, as they require support from others for often even the most basic of everyday tasks. Being able to live an independent life free of impediments is a key factor in increasing quality of life, and access to buildings is one of the key difficulties faced. Easing this process would ease a significant obstacle for disabled people's lives.

Governments of developed countries around the world are placing greater focus on disability issues than ever before. [3] The U.S. Department of Labour recently moved to equate disability with race and gender under Affirmative Action laws. Likewise, Canada recently passed legislation focussed on implementing and enforcing accessibility standards with respect to goods, services, employment and the “built environment.” Any company seeking growth cannot have no response to this huge potential market otherwise it will lose esteem with the increasingly vocal disabled customer base.

Market research analysts have estimated Europe to be the largest market for smart locks during the forecast period. [4] This region currently accounts for more than 53% of the total market share and is envisaged to retain its dominating hold over the market by the end of 2020, owing to the augmented demand for smart locks from non-residential customers like the hospitality industry. This industry will witness a huge demand for smart locks as they must cater to the security and privacy needs of customers to maintain a good brand name.

Smart Locks represents a digital revolution in healthcare and home security. It provides convenience and maintains security. The group’s product, named KnockLock, with its unique, intuitive operation, will perfectly solve people’s concerns while providing a high assurance of security.

2.3 Market Research

The commercial opportunities available to those who are able to meet the needs of disabled people are massive. It is estimated that in 2008 the total market for disability equipment was £1.46 billion [5]. The market is also growing tremendously fast. Sales have increased 92.6% in the last 10 years and the total market size increased by 9.2% last year [6]. In short, there has never been a better time to enter this market, while considerable scope remains for innovation, and before the market inevitably becomes consolidated.

The most convenient and safe locks existing on market are RFID locks and coded locks. Compared to RFID locks, The KnockLock provides much more convenience, as it does not need a physical key or RF card. For instance, The KnockLock can be perfectly applied as a gym locker. Users can set a personalized password when storing their belongings and easily open the locker after training even when they are holding a large bag. In similar places like the supermarket or garage, The KnockLock can easily provide assistance.

The personal locks market is highly diverse and fragmented, making it is hard to find much information about the exact market size. But there is one thing to be sure of, which is that there exists a huge potential market for the next generation's smarter, safer and more intuitive lock.

In terms of cybersecurity, other lock systems on the market use smartphones as a key to the home, verifying identity via Bluetooth or using smartphone apps to save security information. While it is convenient that a physical key is not required, there are some major concerns. As technology develops the concern for privacy increases as well. Digital security systems are prone to hacking and the number of crimes related to hacking bank accounts or smartphone information have increased over time. In the UK, cybercrimes are hugely under reported, causing the full extent of the impact to remain unknown. In 2015, the Office of National Statistics included cybercrimes in the annual Crime Survey for England and Wales. It is estimated that there were 2.46 million cyber incidents and 2.11 million victims of cybercrime in the UK in 2015. However, the actual number of reported cybercrimes are far less than estimated, with only 16,349 cyber-dependent crimes reported and around 700,000 cyber-enabled crimes reported during the same period [7]. If more houses implement this method of security, hacking crimes will increase since hacking other people's phones could potentially grant access to their houses as well. To prevent new methods of crime being utilised in the future, the project will keep all information private to the user only.

As the project progressed, the team realised that the designed mechanism is not just useful as a lock. The innovation in its intuitive control and pattern recording capability allows it to serve as the bases for other products. One possibility is to use it to play a guessing game between children. Alternatively, it can be used for training beginner drummers or other tasks that require the memorisation of complex patterns. However, due to the time constrains of this project, these could not be implemented or demonstrated with the same level of quality like that of the lock. As such these further ideas will be discussed further under the 'Future Work' section.

3 Design Specifications

3.1 Design Criteria

The design aim of the project was to design a device which, upon taking in some sort of pattern input from either a person with limited control over mobility or a person using an electronic locker, would be able to unlock a door. In order to achieve this, the design had to be:

- Simple and intuitive to use – most predicted users would probably lack sophisticated technical knowledge, so being clear to use for everyone is important.
- Secure – since this is a method to enter people's residences/workplaces, it is essential that it doesn't present an easy method for intruders to gain access to these places.
- Relatively inexpensive – this is particularly important to people with dexterity impairments, as they tend to be less wealthy than average.
- Robust – the system may be subjected to large amounts of repeated use, along with a wide range of physical shocks and vibrations, and should be able to withstand the vigour of everyday use. This is particularly true in the context of electronic lockers, which are often subjected to a large degree of physical stress.
- Reliable – the system should be able to work well without issue for a very long period of time. This is particularly true when it is used to gain access to a place of work or residence. A failure would cause significant inconvenience to users of the premises.
- Not physically challenging for someone with limited dexterity

3.2 Conceptual Solution

To implement these criteria, a prototype design has been created. It consists of an array of piezoelectric sensors mounted on a block of wood. These detect vibrations and shocks caused by the user knocking on the door. These vibrations are measured by a microcontroller, which recognises whether a particular pattern (specified by the user) has been knocked. If the pattern is recognised (within set tolerances), the microcontroller will send a signal to an electric door lock, to unlock the door and allow the user access. In order to reset the pattern, a series of buttons and lights will be mounted on the other side of the door, allowing the user to enter a new pattern by knocking again.

Most people have considerable experience knocking doors, and comes naturally to most people, making a knock pattern recognition lock intuitive for the clear majority of people. This and the switch reset system do not require any technical expertise whatsoever, so no additional knowledge is required to use the system. As a physical action, knocking is unchallenging to accomplish, even for those with dexterity impairment. The device can be constructed using off-the-shelf components, all of which are relatively inexpensive, making the system very cheap to produce. In fact, the majority of costs will ultimately come from the installation. The components will all be shielded by protective covering, to guard against accidental strikes to the electronics, and most of the components are electronic, which generally are very reliable. Although the lock is a mechanical moving part, most commercially available units are very robust, with a sufficiently long expected lifespan for reasonable practical use. Electronic components tend not to fail very often at all, so it is predicted that any significant failures will come from either the lock or damage to the wiring. Since these are generally unlikely, it should provide an adequate lifespan for the system.

Security may present an issue, due to variation in sound produced by knocking different regions of a hard door, which could be audible to potential intruders, who could copy the pattern to gain access to the building or locker. To resolve this problem, it is proposed to put some sort of padding on the area of the door to be knocked, to dampen any sound produced and improve security, as well as increasing user comfort.

4 Project Management

The group was divided into three subgroups each working on a portion of the project.

The hardware team, as suggested by its name, oversees the hardware design and implementation of the project. The hardware design is then broken down into three parts, circuit design, breadboard and Perfboard implementation, as well as the mounting of the completed device onto the door prototype. The circuit design process, as it is the most vital and time-consuming portion of hardware, is split further into microcontroller input and output circuit designs.

The software team handles the coding of the microcontroller. The microcontroller needs to monitor the input from piezoelectric sensors constantly, while simultaneously making decisions regarding the correctness of the knocking pattern received.

The webmaster creates and maintains the website put up for this project. The main workload here is to build a structure for the webpages containing all the information about the project. It must be polished enough for easy access, navigation, whilst having adequate visual design to make it pleasing to the eyes of the customers.

Software	FSM Decision Making	Constantinos, Loizos
	Output Logic	
	Programmability	
	Code Optimisation	
	Additional Functionalities	
Hardware	Sensor Input Circuit	James, Joshua, Kantapat
	Output Circuits	
	Circuit Boards Construction	
	Prototype Building	
Website	Architecture	Zhongxuan
	Content Creation	
	Visual Design	
Team Coordination		Zifan
Project	High Level Design	
	Quality Control	
Report Writing		All Members

Table 1 Work Load Distribution

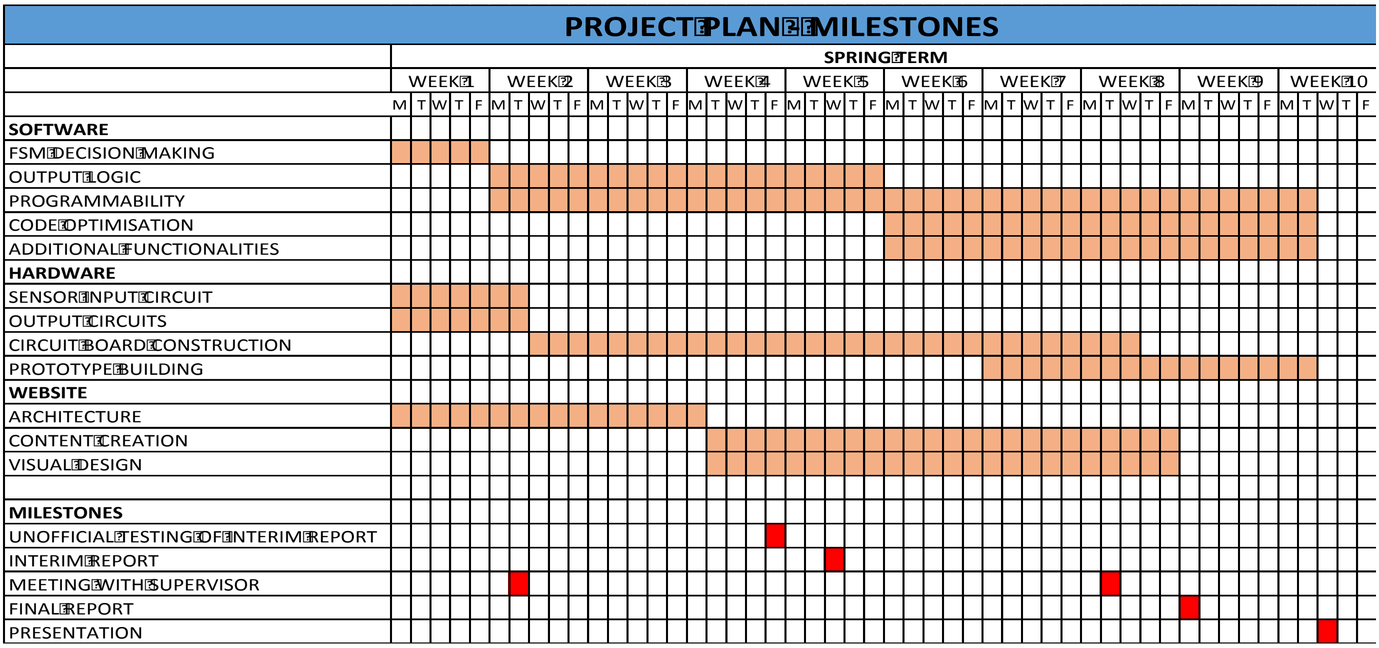

Figure 3 - Project plan and Milestones

5 Design and Implementation

5.1 High Level Design

5.1.1 Block Diagram

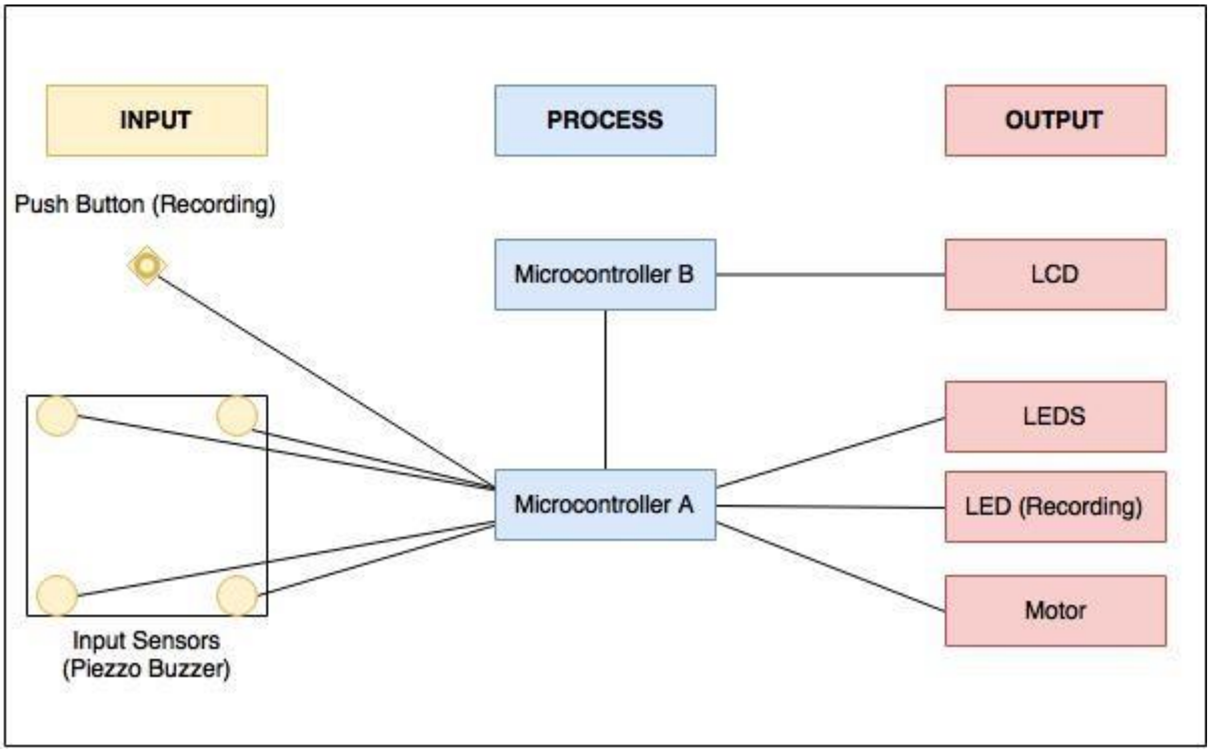

Figure 3 - Block Diagram

This shows the structure of the project's approach in achieving its purpose. As it is noted above, three stages are declared (input, process and output). According to the input received, relevant processing takes place and the expected output produced. The input stage comprises of the piezoelectric sensors and a push button. Piezoelectric sensors detect a knock and the voltage generated by the sensors is amplified into the analogue inputs of the microcontroller. The microcontroller checks if the pattern knocked on the door matches a 6-digit password set by the user, one of the six LEDs light up when each digit is inputted correctly. When the correct password was entered, the microcontroller drives a voltage into the motor which represents the door being unlocked. A push button is used to access the 'record new pattern' function of the product. When the push button is pressed, the microcontroller records the new pattern while indicating the identification of a knock by flashing the Recording LED. Finally, microcontroller A (U2 in figure 4) is connected to microcontroller B (U3 in figure 4) using the method described in detail in section 5.5.2.4. to display relevant messages on LCD screen.

5.1.2 Innovation

The innovation behind this device is the method of inputting data through vibration. The main factor to be concerned about regarding the device is convenience. As tapping on a surface is simple and requires no extra tools, this allows everyone to be able to use it without trouble. The sensors are arranged in a two by two array to form a rhombus, giving the processor four individual voltage readings to use. When each element is arranged far enough apart, by comparing the voltage spikes on each element, it will be possible to detect knocks coming from a three by three grid. This cuts the sensor costs and also raises the complexity of the possible knock patterns. This method of inputting data has the potential to be integrated into other products and will be elaborated on in the future works section 6.6.

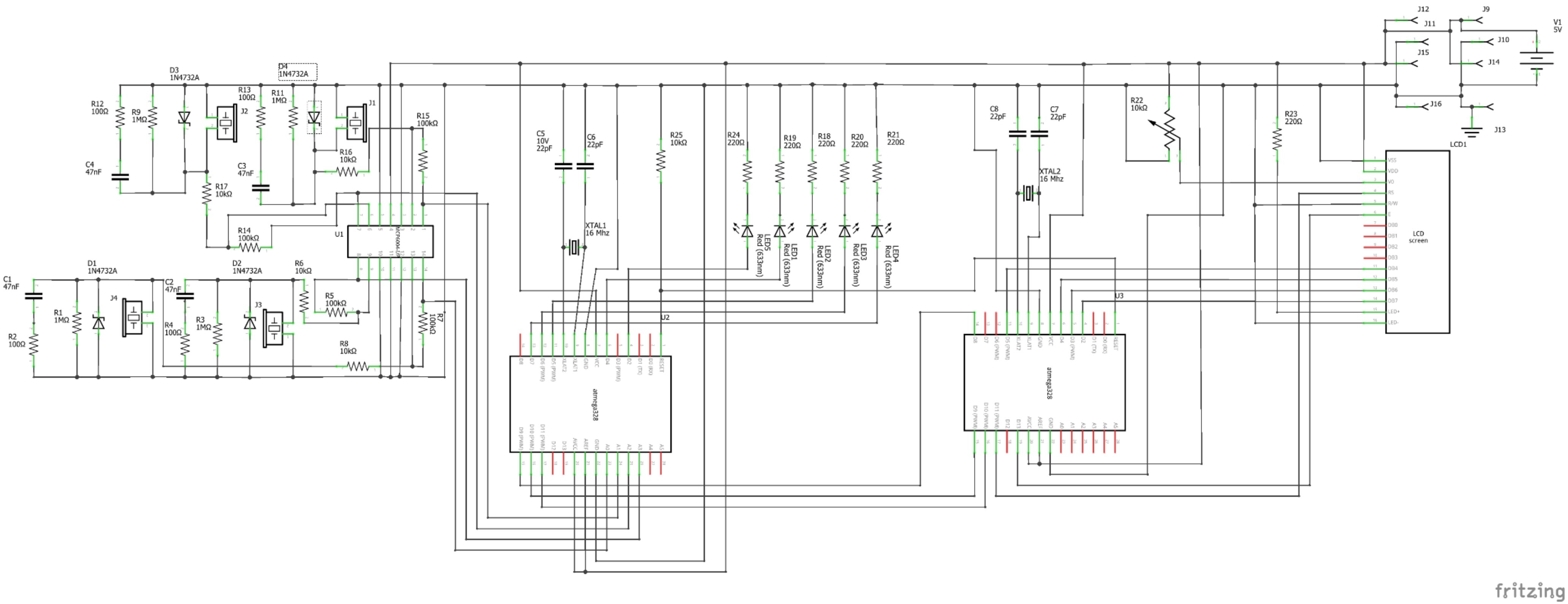

Figure 4: Circuit schematic of the latest test circuit.

Figure 4 shows the overall layout of the current design (at the time of writing). The sensing and amplification circuits are shown on the left of the diagram, with the two microcontrollers in the middle. U2 receives and processes the input signals, and sends them to U3, which displays a message on the LCD screen LCD1 (on the right of the diagram). The operation of the circuit is explained in detail later in the report. By the time the circuit is demonstrated, it is expected that it will be mounted on Perfboard and an electromagnetic lock (with appropriate powering circuit) will be added. The MCP6004 is an IC containing four op-amps, used to amplify the input signals from the Piezo sensors, (the J components). U2 and U3 are Atmel Atmega328 microcontrollers, programmed using the Arduino IDE, and LCD1 is a Winstar WH1602B 16x2 LCD display board. A YwRobot 5V breadboard power supply is plugged in to the headers in the top-right corner to provide power to the circuit.

5.2 Input Systems

5.2.1 Piezoelectric Sensor Inputs

5.2.1.1 Components Chosen

The piezoelectric sensor was chosen as it can detect a deformation caused by the user knocking, by converting it into an electrical signal. This ABT-441-RC Piezo Element was picked due to its high maximum voltage rating and its durability, being able to function after minor damages. [8]

5.2.1.2 Challenges Faced

After some initial tests, the group found out that, if the piezoelectric sensors are mounted behind the plywood block, knocking on the wood will only generate a peak of about 100-500mV. Even if is banged as hard as possible, the peak will still be at around 1V, rather than the 3-4V peak generated when tapping on the sensor directly.

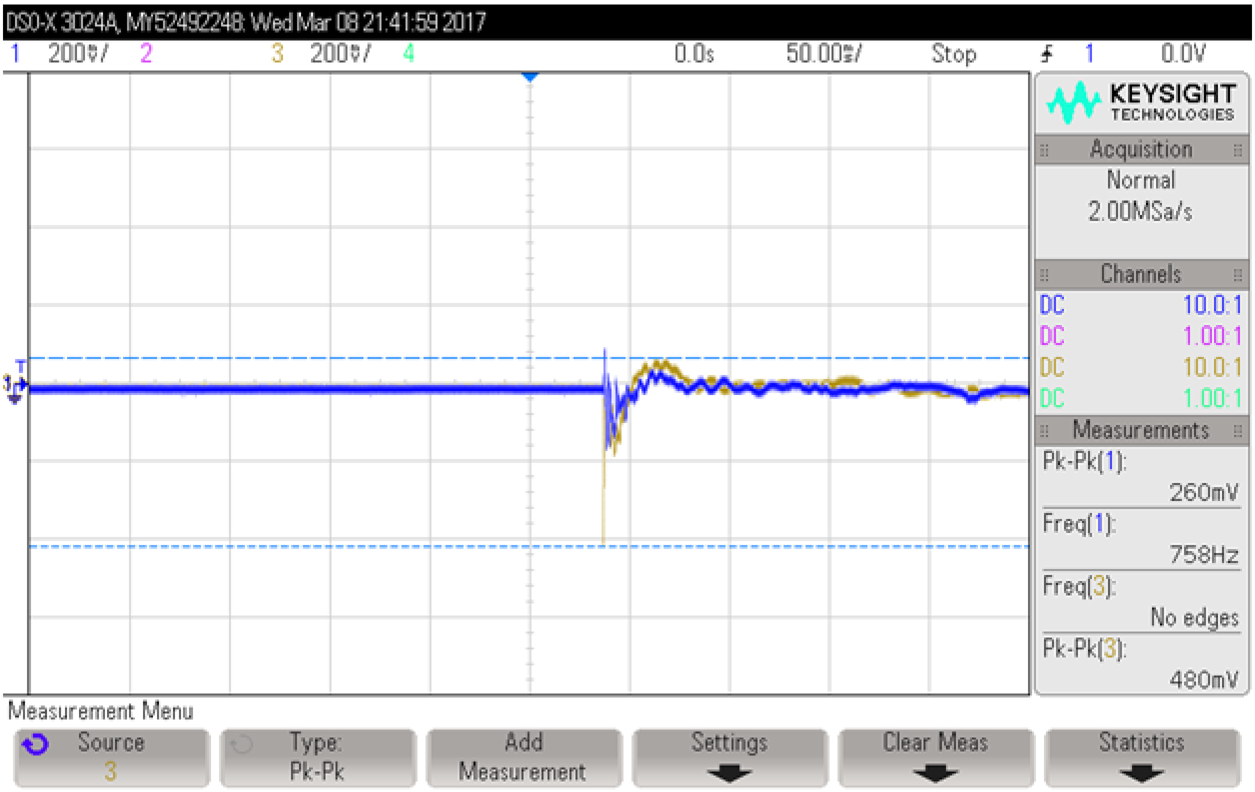

Figure 5. An oscilloscope screenshot showing the signal from two neighbouring sensors when only one is directly knocked on.

The yellow trace is the voltage detected by the sensor closest to the knock, whereas the blue trace is the voltage detected by a neighbouring sensor. It can be seen from the above screenshot that the difference in peak to peak voltage generated is roughly 300mV. This is quite low, when it is considered that the microcontroller has a supply voltage of 5V and a 10 bit device will have 5mV steps. With such a small margin for error, it was decided that the input signal should be manipulated and amplified, so that the probability of making a wrong decision on the position of the knock will be much smaller.

When looking into which amplifier should be used, there were several considerations for this particular project. There are several methods to amplify the signal, including using BJTs or MOSFETs as a switch to make it into a digital signal, using an operational amplifier, using logic gates, etc. After discussion, the group has come to a conclusion that the circuit should have a small power consumption, as well as occupying a small area on the board. Therefore, the operational amplifier (op amp) would be the optimal solution. However, with an op amp, there are also more criteria to consider.

First of all, the whole circuit would be running at 0 to 5V, and therefore, it might not be suitable to use a dual supply op amp. However, single supply op amps have quite a few limitations. For example, the Texas Instrument LM series op amps were looked into. In their specification, they required the input voltage to be within the range of -0.3V and V_{cc}. This was quite problematic because the voltage generated would be more negative than -0.3V. As a result, the op amp the ultimately adopted, the MICROCHIP MCP6004-I/P, can take an input voltage of -1V of V_{ss}, which is usually grounded, meaning it can take at the minimum of -1V. This can be easily achieved by clamping the input pin with a Zener diode to limit the voltage to -0.7V.

5.2.1.3 Implementing the Solution

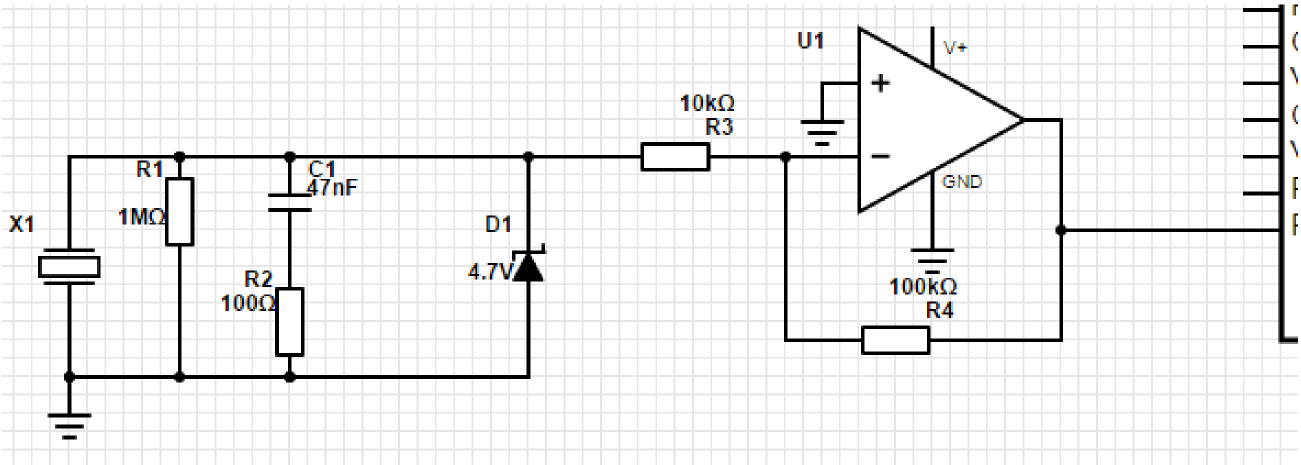

Figure 6. The circuit diagram for the input of one of the sensors

The output voltage signals from the piezo sensors are far from ideal. It is short, rapidly changing, and can potentially generate a voltage anywhere between 30mV and 12V. As such, a method had to be devised to manipulate this input, for it to be read by the microcontroller accurately and reliably.

To smooth out this signal, an RC circuit was implemented, capable of removing most of the high frequency components from the Piezo output. An RC circuit with R = 1MΩ and C = 47nF was implemented to smoothen the sharp signal detected from the sensor, the discharging capacitor allowing the period of the signal to be controllable by changing the value of R, varying the time constant. A small resistor of 10Ω was also added in series with the capacitor to limit the current spikes when the voltage source is turned on. A Zener diode was also used here to limit the voltage generated by the piezoelectric element so it doesn't not exceed 4.7V. This is to ensure that the capacitor does not overheat when the voltage across it is higher than its rated value. It also protects the microcontroller from receiving a voltage higher than its input capacity.

After the circuit was built, it was mounted to a sheet of wood which represents a door. A problem that was faced was that when the piezo sensors were mounted directly to the knocked surface, they all detected similar signal amplitudes regardless of the knocked location. This was solved by using Blu-tack as a medium between each sensor and the surface. This dampens the knock impact and thus, reduces voltage produced on sensors whose locations are not directly at where the user knocks the surface.

Since the knocked surface is on the opposite side of the door as the circuit, the voltages generated by the piezo sensors were greatly reduced to a range 0.2V to 1.3V depending on the force of the knocks. Thus, the amplifier was used to ensure that the microprocessor can clearly determine where the door was knocked. An MCP6004-I/P chip [9] was used since it has 4 op-amps, one for each sensor, and the voltages generated by the sensors were within its maximum input voltage rating.

Because the op amp is single-supplied, the output voltage is limited to within 0 to 5V. With the help from the amplifier, the output signal becomes cleaner with an easily distinguishable peak. The amplification also makes it easier for the microcontroller to determine on which sensor the user has knocked on. This can be seen on the following screenshot, where the magenta line (signal 2) is generated by the sensor located at the knock, and the blue line (signal 1) is from a neighbouring sensor. Yellow and green are the voltages after being amplified by the inverting amplifier.

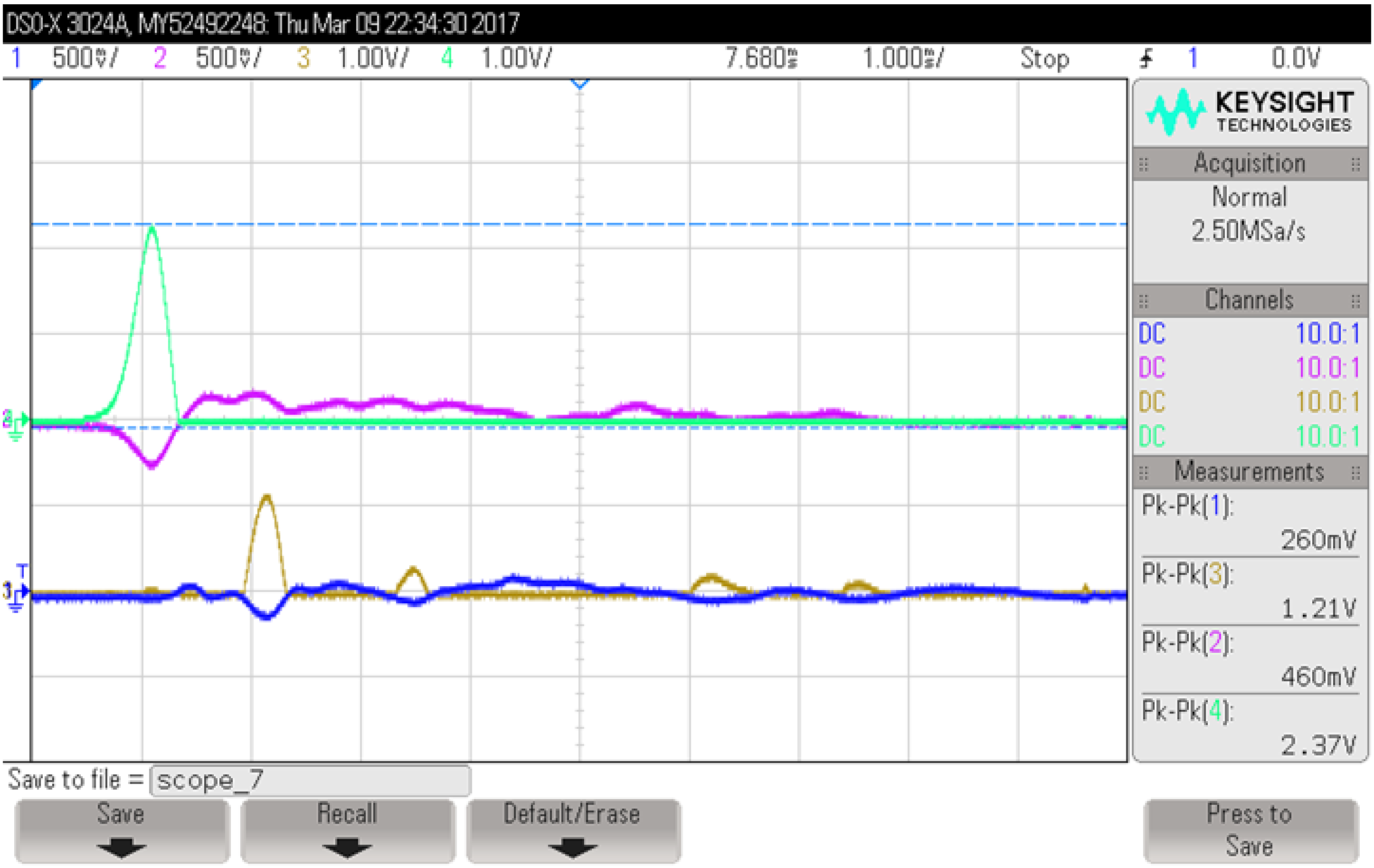

Figure 7. The screenshot of the oscilloscope straight after a knock. Note the difference between the final inputs, signal (3) and (4).

5.3 Output Systems

5.3.1 Light Emitting Diode (LED) Output

The LEDs are used as an indicator to the status of the mechanism. The single red LED near the power socket indicates the power supply is connected and is providing usable voltage. The line of LEDs can be read to determine the state of which the microcontroller is in. These are extremely useful in the debugging process, and they provide a good visual cue to prompt inputs from the user. For security reasons, however, not all state LEDs will be present in the final, commercialised product.

The circuit is extremely simplistic. The diodes do not require much power, and as such can be powered directly from the output signal of the microcontroller. The output pins on the microcontroller work as the supply lines for the LEDs. From the data sheet, the voltage drop and forward current needed for them to switch on can be extracted.. With these values, suitable resistors can be selected to be connected in series, between the LEDs and the ground rail. [10]

Due to the simplicity of this output, it is very resistant to random errors in the circuit or code. As such it is valuable as a backup and debugging output for the project. In case something goes faulty within the system, it is very likely that the LEDs retain some usability. This provided great insights about fixing and improving the system.

5.3.2 Motor Output

To be physically capable of unlocking a door, the device needs to have an output to actuate the locking mechanism. This can be achieved, for example, by using a motor, an electromagnetic switch, or a relay circuit, etc. In this project, for simplicity sake, the group is going to use a servo motor to represent the output.

The specific servo motor (Parallax Inc. 140Ma) was chosen for its low voltage requirement at 4 to 6 V, and its adequate torque output. A low voltage requirement is necessary since the circuit supply rail will be at 5V, and it would take a relatively costly power circuit to shift this higher to supply a higher rated motor. The chosen motor provides enough torque to quickly rotate a reinforced locking bar.

Servo motors are used to control position and speed very precisely, which in the project, the position has to be controlled. The servo motor that was used (Parallax Inc. 140Ma) takes in the input voltage signal which drives the motor. The signal will be a PWM output signal from the microcontroller (3.3V-5V), and is aimed to control the position of the servo shaft. In order to hold its position, the servo needs to receive a pulse every 20ms with a width of 1.5ms.

The servomotor used can turn 90° in either direction, i.e. maximum movement can be 180°. A pulse is sent every 20 milliseconds, and the width of the pulse determines the position of the shaft. The angle of the shaft is mapped from 0 to 180 degrees with respect to the pulse width. This way, when the motor operates, it can move at full power and have a quicker response, without being affected by the PWM control. In this case, a pulse of 1ms will move the shaft anticlockwise at -90°, a pulse of 1.5ms will move the shaft at the neutral position and a pulse of 2ms will move the shaft clockwise at +90°.

The rotation of the shaft can then be used to lock and unlock the door depending on the structure of the door it will be installed on. If there is a hole for a locking bar, then this rotation can be easily converted to a linear motion with a 'half' of a crank. A crank is an arm attached at a right angle to the rotating shaft. It is a simple mechanism that can convert circular motion into reciprocating motion, exemplified by a hand crank on a manual pencil sharpener. Since 180 degrees of rotation need to be created, 'half' of a crank is required. Otherwise, a simple hole can be cut off in the shape of a sector of a circle. Then this servomotor can carry a solid piece of material that fits directly in the hole, thus locking the door. [11]

5.3.3 Liquid Crystal Display (LCD)

The LCD provides a user-friendly feedback to indicate the status of the product. This is especially important in recording mode, as it might be difficult to know with confidence what has been recorded in the target pattern. As such, a LCD that displays when recording starts and ends is extremely helpful to provide more confidence to the user in the product's accuracy and responsiveness.

The Winstar WH1602B-NYG-JT is chosen for it is the only LCD under £5.00 that provides a 32-character output. The exact functionality will be described under section 5.5.2.5. [12]

5.4 Processing Systems

5.4.1 Microcontrollers

For the microcontroller, it was decided that the ATmega328 would be used for the project. First, the group would require a microcontroller that has a decent amount of built-in memory because they are needed to store the user knock pattern, which the program can be used to compare with the input pattern. ATmega328 has plenty of high endurance non-volatile memory segments available, including a 32KB Flash Memory, 1KB EEPROM, and a 2KB SRAM, which are sufficient for this project. On top of that, it is required to have at least 4 input pins that consist of an Analogue-to-Digital Converter (ADC), as well as a few output pins for the LEDs, LCD monitor, and the motor. ATmega328 has more than enough of ADCs for the project.

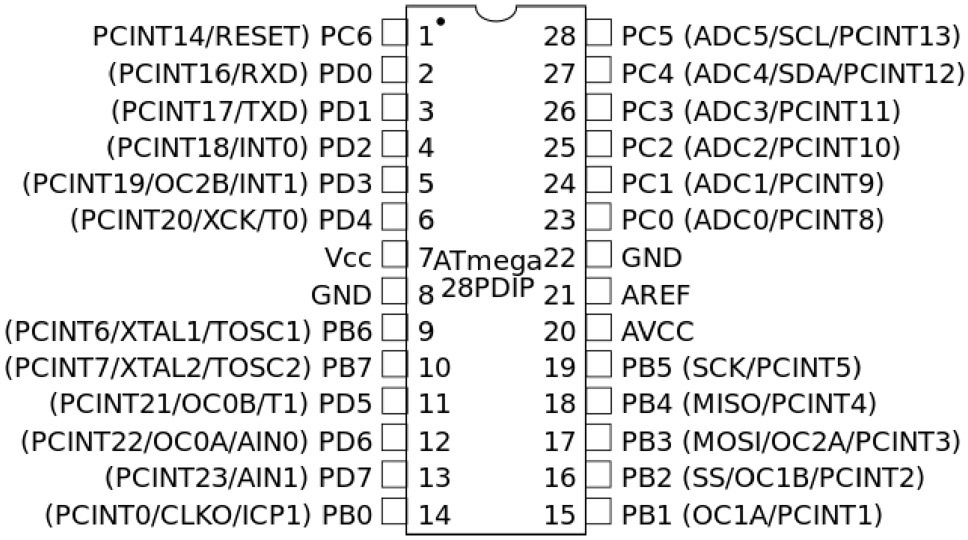

Figure 8: Pin Layout of ATmega328 in 28-PDIP [13]

Another main criterion for the microcontroller is how the difficulty to program the chip. Since the group do not have much time to complete the project, time is also a main consideration. From experience from last year, it was decided to avoid buying a microcontroller that would need to be programmed in assembly language or require a specific programming suite. As most of the group member have had experience with using Arduino from last year's EEBug project, it was decided that the Atmel AT series microcontroller would be used. Within the AT series, there are concerns as well on how to complicated would it be to load the program into the chip; for ATtiny, the most popular microcontroller used in the EEBug project last year, was notoriously difficult when it comes to loading a program into it. In comparison, ATmega328 is a more commonly used microcontroller and is therefore more convenient in that sense.

Furthermore, ATmega328 has a sophisticated architecture that allows it to achieve up to 16MIPS (million instructions per second) at 16MHz. It has an interrupt control, which allow the group to be more flexible when it comes to choosing between polling or interrupting. It has a low power consumption because it can switch between different modes when idle. It also has a QTouch Library available, which 'provides a simple to use solution to realize touch sensitive interfaces', and this is also very useful for the required purposes.

Overall, ATmega328 fulfilled all the selection criteria, and also provided many other perks that were helpful and convenient in this project. [14]

5.5 Software Implementation

The Logic System consists of a set of instructions the microcontrollers have been programmed to follow to obtain and process the analogue input signals from the piezoelectric sensors and provide digital output signals to the three output components.

5.5.1 Microcontroller A

Principally, the microcontroller acts as a finite state machine (FSM). It takes in the voltage signal input from the piezoelectric sensors, and changes the output signals digitally based on current and previous inputs.

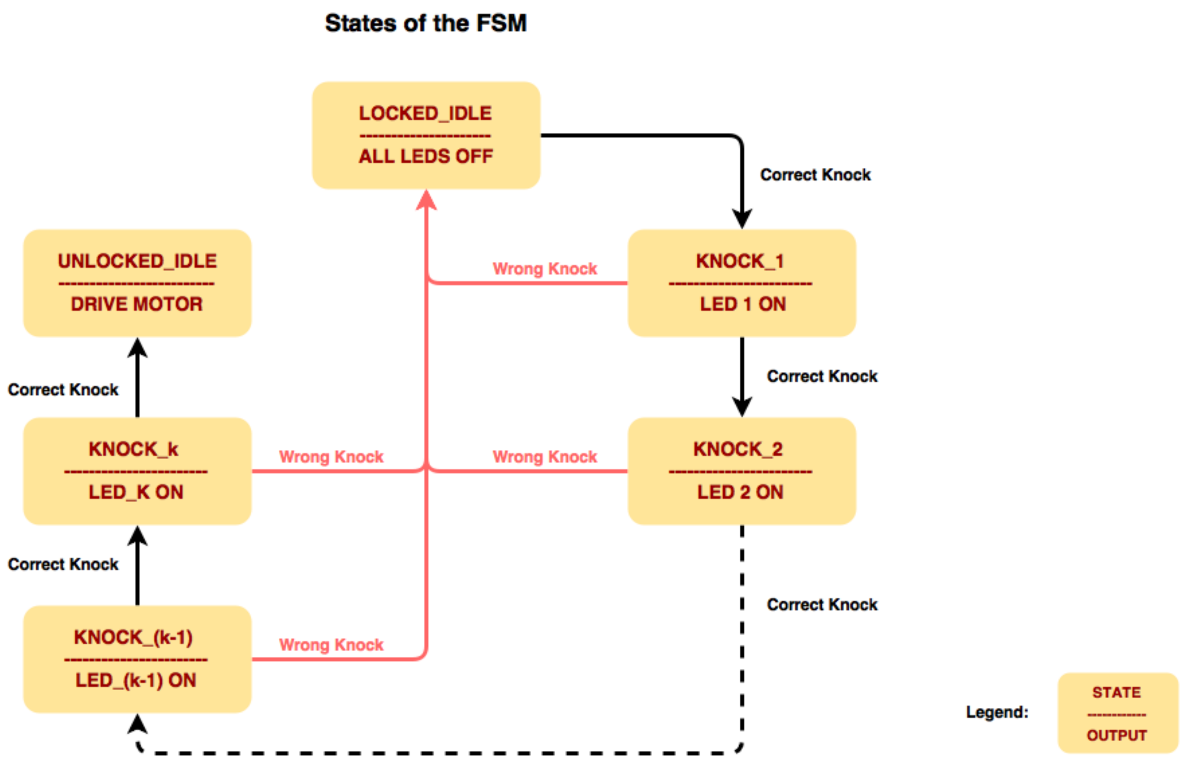

Figure 9: State Transitions of the Finite State Machine

In this simplified block diagram, state changes caused by a correct input are shown as black arrows. All other connections in red simply point from every input state to LOCKED_IDLE, used when an incorrect input is received. To explain with an example, suppose the correct knocking pattern is /a, b, c, d, ..., n/, then when an /a/ is received at LOCKED_IDLE, the state will change to KNOCK_1. If a /b/ is decoded at KNOCK_1, then the FSM will move on to KNOCK_2, and the rest follows. When the final /n/ is read by the microcontroller, and when the FSM is at KNOCK_k, then it will reach UNLOCKED_IDLE, at the same time sending out an output signal to the motor and LEDs to indicate the opening of the door.

However, if for example, a /d/ is read at KNOCK_2, instead of the expected /c/, the FSM will reset itself back to UNLOCK_IDLE. This restarts the process as the input knocking is wrong. To enter the door successfully, the user would now have to re-enter the knocking code again, from the beginning. This is made as such for security purposes.

5.5.2 Code Explanation using pseudo-codes

In this section the main functions of the programming part of the product are explained in detail. For this reason, pseudo-codes are used to make the explanation of this part as understandable as possible. Please refer to the appendix for the full code. It is commented and formatted for ease of reading.

5.5.2.1 Loop Function Part I – Processing Input

A sequential array is created with the target pattern

For example, setting this array to [2, 2, 3], creates a target pattern of length 3. The user needs to knock on the second sensor twice before unlocking the door with a knock on the third sensor.

A chain of else-if statements follows to process these values.

```
If input reading of first sensor is greater than the threshold defined
       hit identified is true
hit spotted at location 1

else if input reading of second sensor is greater than the threshold defined
    ……

else if input reading of fourth sensor is greater than the threshold defined
       hit identified
hit spotted on location 4
```

As explained previously in the sensor input section, the voltage amplitude is directly related to the position of which sensor is being knocked. Therefore, by comparing every input voltage with a pre-set voltage threshold, the location of the knock can be determined.

5.5.2.2 Loop Function Part II – Decision Making in the Finite State Machine

The finite state machine (FSM) is used to determine the correctness of the output. It is broken into two parts, one to control the state transitions, and one to govern the output when different states are reached.

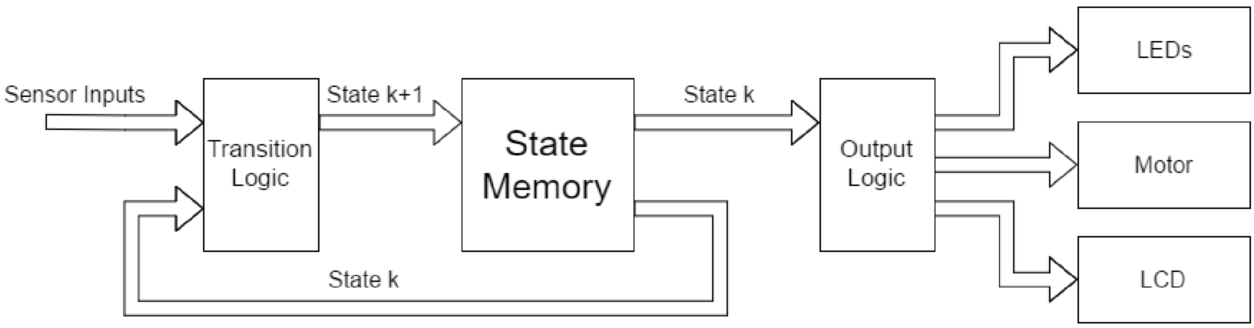

Figure 10: Working Principle of the Finite State Machine

This diagram shows how the finite state machine works. A switch-case statement is used to define the transition logic.

```
//FSM - Transition of States

Case switching based on state value

       case LOCKED_IDLE:
              go to state KNOCK_1

       case KNOCK_1:
              if hit is identified{
                     if hit location matches with value of array Sequence{
                            set hit to false
                            go to state KNOCK_2
                     }else{
                            set hit to false
```

```
                        go to state LOCKED_IDLE
                }
        ……
        case KNOCK_k:
                if hit is identified{
                        if hit location matches with value of array Sequence{
                                set hit to false
                                go to state UNLOCKED_IDLE
                        }else{
                                set hit to false
                                go to state LOCKED_IDLE
                        }
                }
        case UNLOCKED_IDLE:
                unlock door
end case
```

LOCKED_IDLE is the default case. The FSM falls back to this state whenever an unexpected input or calculation happened. In this state the lock is shut, and all inputs are cleared.

Most other states are KNOCK_i states. These states have two possible next states, so there are logic statements to determine which state to switch to. Following the code, it can be seen when the detected hit position is the same as the i^{th} value in the sequence array, the state is shifted to the next KNOCK_i+1 state. However, if it is different from the expected i^{th} value in the array, the state will fall back to LOCKED_IDLE, effectively resetting the system.

The other part of the FSM is the LED output logic.

```
//FSM - Output of States
        Case switching based on state
                case LOCKED_IDLE:
                        LED_1 off
                        LED_2 off
                        LED_3 off
                        LED_4 off
                case KNOCK_1:
                        LED_1 ON
                        LED_2 off
                        LED_3 off
                        LED_4 off
                case KNOCK2:
        ……
```

This is another switch-case statement. It takes the state value and outputs the relevant signals. This code snippet is the LED output logic. It is relatively simple, setting the different lights high or low to switch them on and off. As explained under the LED section, this then indicates the state of which the machine is in, making it easy to debug and check the correctness of the FSM design.

5.5.2.3 Recording new pattern

The product is capable of recording a new target pattern defined by the user, replacing the one existing. For this purpose, a function is created which is enabled when the recording button (a push button) is pressed. The first address of pattern is passed using pointers to the function and the change of the data is also implemented in a similar manner. A led (LED_R) flashes to indicate that a knock is identified.

```
function RecordPattern (pointer that point to the first element of array passed ){
        initialise length to 0
        while length is smaller than size of array
                read values from sensors
                If input reading of first sensor is greater than the threshold defined{
                        hit identified
                        save hit location as 1
                }else if input reading of second sensor is greater than the threshold defined{
                        ……
                }else if input reading of fourth sensor is greater than the threshold defined{
                        hit identified
```

```
                    save hit location as 4
            }
        }
        if hit is identified{
            pointer value = location spotted
            increase length by 1
            increase address of the pointer
            LED_R off
            delay for 0.5 seconds
            LED_R on
        }
}
```

5.5.2.4 Communication between Microcontrollers

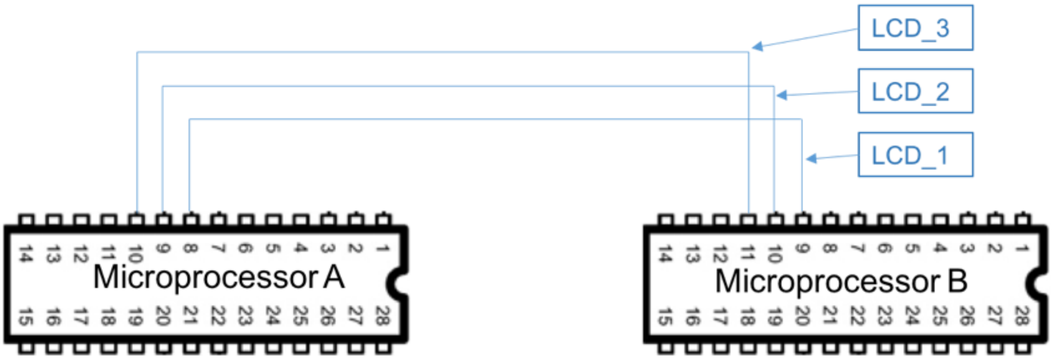

Figure 11: Interconnection

As the number of digital pins on a single microcontroller is insufficient, a second microcontroller is used purely for the operation of the LCD screen. The messages to be displayed on the LCD screen are directly related to the state in which the FSM of Microcontroller A is in. To achieve that, there must be a way of communicating between the two microcontrollers. Besides wiring up the pins, part of the code is dedicated in creating functions that provide a digital output of binary digits 0 or 1 to the interconnected pins that are used as inputs for Microcontroller B.

The pseudo-code below, is an example of such a function. Six of these functions were created corresponding to each transition the FSM goes through. Functions that have all three pins set to low or set to high are avoided, since they can create interferences and lead to unexpected results.

```
#define LCD_1 = pin 8;
#define LCD_2 = pin 9;
#define LCD_3 = pin 10;
void function message1{
        LCD_1 = 0;
        LCD_2 = 0;
        LCD_3 = 1;
}
```

5.5.2.5 Microcontroller B

Microcontroller B is directly involved in printing out messages on the LCD. Using the binary coded message produced by Microcontroller A as inputs, Microcontroller B decodes it using the following combination table:

Combination Number	Binary Code (LCD_1 | LCD_2 | LCD_3)	Output Message
A	001	"PLEASE KNOCK"
B	010	"KNOCK IDENTIFIED"
C	011	"DOOR UNLOCKED"
D	100	"WRONG KNOCK"
E	101	"START RECORDING NEW PATTERN"
F	110	"RECORDING ENDED"

Table 2 - Communication between the two ATMega328

The following diagram shows how the LCD works in pseudo-code.

```
Read combination
if combination received is A {display “PLEASE KNOCK”}
else if combination received is B {……}
     ……
else if combination received is F {display “RECORDING ENDED”}
```

Each input pin is assigned an alphabet letter for simplicity reasons. The main function runs in a loop checking each time the binary digit of either A, B or C. If the combination of A, B and C represent either one of the combinations shown on the combination table then the LCD outputs the corresponding message. The LCD can display up to 16 digits on a line, so the message output should carefully be chosen in order to fit within those 16 digits. If not, then the LCD can be programmed to display the remaining message on its second 16 digit line using the appropriate command. Similarly, to the LEDs, the LCD screen indicates the state of which the machine is in, making it easy to debug and fix any unexpected errors.

5.6 Physical Implementation

5.6.1 Prototyping the Door and Circuit Boards

A plank of wood was required to build the prototype for the product. It should be as close to a normal wooden door as possible. However, a small piece of wood suitable for demonstration was not readily available on any of the preferred stores from the Electrical and Electronics Department. A compromise was made to resolve this issue.

Two pieces of 600 mm × 300 mm × 6 mm were chosen to build the prototyping base. They were cut into four pieces first and then secured by four screws in the corners, to achieve minimum disturbance to its audio properties.

A standard external door (Green) is of size is 1981 mm × 610 mm × 44 mm. For convenience, the prototyping board is in the shape of a square and in the scale of 1:2 of the original. The square is of size 610 mm × 610 mm × 44 mm (Yellow), and the prototype is roughly half of that, at 300 mm × 300 mm × 24 mm (Blue). [15]

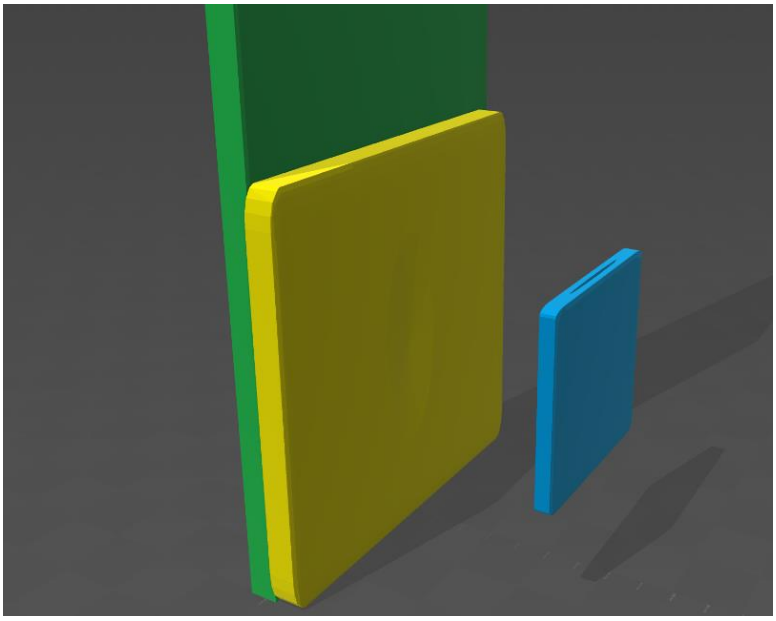

Figure 12: from left to right:
Dimensions of a real external door, the same door cut into a square piece, and the prototype board at scale of 1:2

Due to time constrains it was not possible to implement the circuit in the desired PCB form. Therefore, the next best thing was chosen, to solder the circuit onto Perfboard.

5.6.2 Power Supply

The system needs to be provided with a stable direct current supply, as this is required by the microcontroller, the LCD, and the DC actuator. Considering the installation process of the finished product, it should be natural and easily achievable for the system to be run under mains electricity supply. Battery supplied devices require a lot more power optimisation in the system for it to be more useful than a hassle to maintain, and therefore this may only be possible as a future extension to the project. It widens the application scenarios possible for the lock. This is discussed under the 'Future Work' section.

As such, the current system is supplied first by an adaptor that converts 230V mains electricity to 9V DC supply. It is then followed by a breadboard power supply unit that steps down this from 9V DC to a 5V and 3.3V DC supply, as used by the logic circuits.

5.6.3 Industrial design, ergonomics and manufacturing considerations

If the device were to be made into an actual product, it would need to be standardized for different consumers around the word. It can be assumed that the device can be mounted and can work on every type of door. Now, the circuit needs to be standardized, so that it is compatible with the mains power supply of different countries. The UK and most countries in Europe, Asia, South America and Australia use a residential voltage of 220-230V with a frequency of 50Hz while in North America, a residential voltage of 120V with a frequency of 60Hz is used. The product should be designed to work with a 230V power supply since running a device designed for 120V to a 230V supply is more dangerous than having insufficient supply on a device, although it not guaranteed to be risk free. To sell the device in North America, another branch of the device with different resistor and capacitor values would need to be made.

The device needs to be ergonomic for all users, especially for people with dexterity impairments. Before anything else, the device must be simple to install. The finished product should be one solid, cuboid shaped object with the sensors fixed on the inner surface of a plastic cover. The plastic cover between the sensor and the door needs to be as thin as possible to reduce vibration loss. The device will be connected to a detachable LCD which will be placed in front of the door either by wire or wirelessly. The disadvantage of a wired connected LCD is that a hole would have to be drilled through the door to connect the LCD to the device however with wireless connections, batteries can be used to supply the LCD which raises the cost. The whole circuit should be fixed on a printed circuit board without any user interaction.

6 Future Work

6.1 Batteries

Although the device is planned to be supplied by the mains, there are some advantages of being supplied by batteries. Using batteries as a power supply requires little maintenance, and can be readily replaced. However, since the microcontroller, the LCD, and the DC actuator needs to be supplied with voltage continuously, the longevity of the battery is a strong concern.

Another concern was that the locking device should still be able to function when the mains power source stops supplying voltage. In situations such as blackouts that may occur more frequently in some areas of the world, many electric lock systems would automatically unlock and the user would be prone to intruders. Having only an external power supply may therefore be insufficient for some users. As such, given more time, an uninterruptible power supply system, or at least a rechargeable battery system should be in place to prevent such unwanted unlocks.

6.2 PCB

All components of the circuit should be mounted on a printed circuit board rather than having them placed loosely on a breadboard. As a product, the device should come in as few pieces as possible so that most users can easily install it. In addition, since the device would be mount on the surface of a door, it should be as thin and low-profile as possible for not ruining the aesthetics.

6.3 Wireless Connection

At the moment, the knocking is done on a fixed position, on the surface of a door. However, for future work, the whole sensor unit may be allowed to be removable from the door. Bluetooth or Wi-Fi can be used to connect the unit in the form of a portable tablet to its microcontroller. This added option will benefit people who may have trouble reaching all positions of the sensors such as being on wheelchairs, or people who wish to carry the tablet around treating it as a key.

6.4 Electric Lock Release

An electric lock system would provide another level of security to a user due to its variety of authentication methods such as passcodes, a security key in many different forms, biometrics, or radio frequency identification tags. This gives the user more options to their method of security. The device isn't solely targeted towards people with dexterity impairments; many consumers who are physically able to use a key or memorize passcodes would want a more complex unlocking method, since the original wall of security can be broken just by others watching and memorising the knock pattern. At the same time, biometric methods such as retina or finger print scan can provide more security to many people with dexterity impairments and not introduce more inconvenience to them.

6.5 Reset Password Key

Since the device does not send or receive any information from the internet, there is no method of retrieving the knock pattern once the user forgets it, although it provides advantages, such as being unaffected by hacking, and requiring less labour cost for technicians to open doors for the user. A physical key that allows the user to re-enter a new knock pattern would be useful when the user wanted to change their knock pattern or couldn’t remember their current pattern.

6.6 Implementing the Knock Pattern Recognition method on other products

The innovation of the pattern recognition can be implemented on more than just lock systems. One possible market that this idea could be applied to is children’s toys for learning. The product would be memory game for children to remember a pattern, each input to the pattern could correspond to a musical note to create a short verse of a song. Another possible product that can use the vibration detection idea is a keyboard without buttons. This would allow users to type without needing to apply force from their fingers and would be especially useful for those who are unable to do so.

7 Conclusion

The project aims to provide people a convenient yet secure method of locking their doors and belongings. To achieve this, the group has developed the KnockLock concept. After a significant amount of building, testing and experimenting, a working prototype was successfully completed, showing that the concept is viable. As such, it is expected that the demonstration will be successful and attractive to its potential investors.

8 References

[1] "Family Resources Survey (FRS) Disability prevalence estimates," the UK Government, London, 2007-2008.

[2] "Labour Force Survey," the UK Office for National Statistics, London, Jan - March 2009.

[3] "How is the welfare budget spent?," Office of National Statistics, London, 2016.

[4] "SMART HOME M2M MARKET IN THE US," Technavio, 2015-2019.

[5] "Equipement for the Disabled," Key Note Limited, 2014.

[6] "Disabled Equipment Portfolio Analysis," Plimsoll, 2009.

[7] "Cyber crime data," National Crime Agency, 2015.

[8] MULTICOMP, "The datasheet for MULTICOMP ABT-441-RC Piezo Element," [Online]. Available: http://www.farnell.com/datasheets/1662482.pdf.

[9] Microchip, "The datashet of MCP6001/1R/1U/2/4," [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/21733j.pdf.

[10] KingBright, "10mm SOLID STATE LAMP," [Online]. Available: http://www.farnell.com/datasheets/1701185.pdf.

[11] "The datasheet of Parallax Standard Servo (#900-00005)," Parallax Inc, [Online]. Available: http://docs-europe.electrocomponents.com/webdocs/123f/0900766b8123f8d7.pdf.

[12] L. Winstar Display Co., "WH1602B-NYG-JT#," [Online]. Available: https://www.rapidonline.com/pdf/57-2224.pdf.

[13] Mediaquark, "ATmega328P 28-PDIP.svg," Wikimedia Commons.

[14] "ATmega328/P Datasheet Complete," Atmel, 2016.

[15] "Doornmore.com," Door & More Inc., [Online]. Available: http://www.doornmore.com/help/what-is-the-standard-size-for-residential-homes.html. [Accessed 08 03 2017].

9 Appendix

9.1 Minutes of Meeting

Meeting: 7

Date: 7/2/2017 **Time:** 13:30 **Location:** Computer Lab

Meeting Called by: Group

Type of Meeting: 2nd Report Review

Facilitator: -

Note taker: Constantinos

Attendees James, Joshua, Constantinos, Loros, Jack, Kevin, Eric

Agenda Topic Report Check After Dr Perea's comments

Discussion We went over the ~~comments~~ feedback from Dr Perea and ~~fixed~~ corrected some of the mistakes.

Conclusions: Most of report was corrected. Just needed some finishing details which were agreed to be put by the end of the day so report can be submitted on Wednesday.

Action Items:	Person Responsible:	Deadline:
Finish Report	All group	7/2/2017
Submit Report	Eric	8/2/2017

Time of Meeting: 1 hour **Next meeting:** 10/2/2017

Meeting: 8

Date: 10/2/2017 **Time:** **Location:** Computer Lab

Meeting Called by: Group Meeting

Type of Meeting: Weekly meeting

Facilitator: -

Note taker: Constantinos

Attendees James, Joshua, Constantinos, Loizos, Jack, Eric, Kevin

Agenda Topic Split workload of project.

Discussion

With the interim report done we can concentrate on developing the product so the main point of discussion is how do we split the work now.

Conclusions:

Kevin as webmaster took responsibility for designing the webpage. Constantinos and Loizos were assigned the code design. James, Joshua and Jack took responsibility for circuit design. Eric as a leader will help in all 3 areas.

Action Items:	Person Responsible:	Deadline:
Code design	Constantinos and Loizos	22/2/2017
Webpage (template)	Kevin	22/2/2017
Circuit design	James Joshua, Jack	22/2/2017

Time of Meeting: 30' **Next meeting:** 22/2/2017

Meeting: 9

Date: 22/2/2017 **Time:** 14:00 **Location:** Computer Lab.

Meeting Called by: Group

Type of Meeting: Check progress

Facilitator: –

Note taker: Constantinos

Attendees Constantinos, Loizos, Eric, Jack, Joshua, James, Kenn

Agenda Topic Check progress on project and discuss report

Discussion Regarding the ~~a~~report since it ~~was~~ is mainly technical we thought it would be better to do some preliminary testing first and then write it. In terms of progress, webpage template done and most of code done

Conclusions: Finish webpage and code asap and do some testing ~~const take measurements~~ so ~~report~~ we can start writing the report.

Action Items:	Person Responsible:	Deadline:
Finish code	Loizos, Constantinos	1/3/2017
Finish Website	Kenn	1/3/2017
Testing for sensors and circuit	Joshua, James, Jack	1/3/2017

Time of Meeting: 30' **Next meeting:** 1/3/2017

Meeting: 10

Date: 1/3/2017 **Time:** 14:00 **Location:** Electronics Lab

Meeting Called by: Group

Type of Meeting: Work on the product

Facilitator: -

Note taker: Constantinos

Attendees Constantinos, Lorizos, James, Joshua, Jack, Eric, Kevin

Agenda Topic Work on project

Discussion

After the code was presented we thought it needed a bit of optimisation but definitely seemed to work as expected. We started testing the sensors however they ~~same~~ results didn't satisfy us.

Conclusions:

We ordered more sensors to have them for further testing. The code seemed to work, just needs a bit more optimisation. Eric said he would start writing report ~~[illegible]~~

Action Items: | **Person Responsible:** | **Deadline:**

Action Items	Person Responsible	Deadline
Optimise code	Lorizos, Constantinos	7/3/2017
Report writing	Eric	7/3/2017
Order new sensors and continue testing circuit.	Joshua, James, Jack	7/3/2017
Finalise some details of website	Kevin	7/3/2017

Time of Meeting: 1h **Next meeting:** 7/3/2017

Meeting: 11

Date: 7/3/2017 **Time:** 15:00 **Location:** Computer Lab

Meeting Called by: Group

Type of Meeting: Progress ~~meeting~~ check

Facilitator: –

Note taker: Constantinos

Attendees Constantinos, Loizos, Eric, Kevin, Jack, James, ~~Eamo~~ Joshua

Agenda Topic Progress on report.

Discussion

Most of report was done. Code worked fine but we decided to add some extra parts to make our product more impressive. Circuit design is at the final stages.

Conclusions:

We decided as a group to finish report by Wednesday afternoon.

Action Items:	**Person Responsible:**	**Deadline:**
Work on ~~pro~~ product development and code	Constantinos, Loizos	10/3/2017
Report writing	Eric, Jack, Joshua, James	8/3/2017
Finish website	Kevin	13/3/2017

Time of Meeting: 30' **Next meeting:** 10/3/2017

Meeting: 12

Date: 9/3/2017 **Time:** 12 30 **Location:** level 11

Meeting Called by: Group- Supervisor

Type of Meeting: Supervisor

Facilitator: -

Note taker: Constantinos

Attendees Constantinos, Lizos, Joshua, Joe, James, Yann, Eric

Agenda Topic final Report and final product.

Discussion

The supervisor gave us advice regarding the final report and some areas which we didn't do properly in the interim report. ~~We also~~ ~~[illegible]~~ Overall though he seemed satisfied with the progress so far.

Conclusions:

We decided to concentrate on final report and correct some of the ~~[illegible]~~ points the supervisor noted.

Action Items:	**Person Responsible:**	**Deadline:**
finish report	Everyone	13/03/2017

Time of Meeting: 1h **Next meeting:** 13/03/2017 to submit report.

9.2 Spending

Supplier	*Code*	*Description*	*Each*	*Quantity*	*Total*
Onecall	1675548	ABT-441-RC Piezo Element	£0.447	4	£1.788
EEEDStore	LC0035	6mm laser-grade birch plywood	£2.98	2	£5.96
Onecall	1675548	ABT-441-RC Piezo Element	£0.447	10	£4.47
Rapid	73-4538	Breadboard Power Supply Module	£2.60	1	£2.60
Onecall	2060973	AC/DC Power Supply	£6.60	1	£6.60
Rapid	57-2224	16x2 LCD Screen	£4.69	1	£4.69
RS	781-3058	Servomotor	£9.37	1	£9.37
Rapid	73-4281	ATMEGA328-PU Microcontroller	£2.28	2	£4.56
EEEDStore	SD0030	Zener Diode 1.3W BZX85 4.7V	£0.26	5	£1.30
EEEDStore	SD0115	Green LED 5mm 2.1V	£0.08	5	£0.40
EEEDStore	SD0120	Red LED 5mm, 2.1V	£0.12	5	£0.60
RS	734-6748	Push Button	£1.10	3	£3.30
Onecall	1605571	Operational Amplifier	£0.486	2	£0.972
EEEDStore	SA0045	SOCKET IC SOLDER TAG 14PIN	£0.11	2	£0.22
RS	827-3559	3M 28 Way IC Dip Socket, 1A	£0.42	2	£0.84
RS	631-3009	ICB-288 Matrix Board, 72x47x1.6mm	£0.74	3	£2.22
RS	922-2142	Atmel ATMEGA328P-PU Microcontroller	£2.50	2	£5.00
		Total			*£54.890*

9.3 Arduino Code

9.3.1 Microcontroller A

```
#define LED_1  4
#define LED_2  5
#define LED_3  6
#define LED_4  7
#define LED_R 2
#define Switch 3

#define SensorA A0
#define SensorB A1
#define SensorC A2
#define SensorD A3

int state;

#define size_array 3

#define LOCKED_IDLE 0
#define KNOCK_1 1
#define KNOCK_2 2
#define KNOCK_3 3
#define UNLOCKED_IDLE 99

#define directThreshold 100

#define LCD_1 9
#define LCD_2 10
#define LCD_3 11

void RecordPattern(int *Pattern);

int Sequence[size_array] = {2,2,3};

void setup() {
  // initialize digital pin 13 as an output.
  Serial.begin(9600);
  pinMode(LCD_1, OUTPUT);
  pinMode(LCD_2, OUTPUT);
  pinMode(LCD_3, OUTPUT);


  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);
  pinMode(LED_3, OUTPUT);
  pinMode(LED_4, OUTPUT);

  pinMode(SensorA, INPUT);
  pinMode(SensorB, INPUT);
  pinMode(SensorC, INPUT);
  pinMode(SensorD, INPUT);

  pinMode(Switch,OUTPUT);
  pinMode(LED_R, OUTPUT);

}

// the loop function runs over and over again forever
void loop() {

  if (digitalRead(Switch)==HIGH){
    Serial.println("Record");


    digitalWrite(LED_1, LOW);
    digitalWrite(LED_2, LOW);
    digitalWrite(LED_3, LOW);
    digitalWrite(LED_4, LOW);
```

```
  state = LOCKED_IDLE;

  digitalWrite(LED_R,HIGH);
  delay(500);
  RecordPattern(Sequence) ;
  digitalWrite(LED_R,LOW);
  for (int k=0;k<size_array;k++){
      Serial.println(Sequence[k]);
  }

  digitalWrite(LED_1, LOW);
  digitalWrite(LED_2, LOW);
  digitalWrite(LED_3, LOW);
  digitalWrite(LED_4, LOW);
  delay(500);

  digitalWrite(LED_1, HIGH);
  digitalWrite(LED_2, HIGH);
  digitalWrite(LED_3, HIGH);
  digitalWrite(LED_4, HIGH);
  delay(500);

  digitalWrite(LED_1, LOW);
  digitalWrite(LED_2, LOW);
  digitalWrite(LED_3, LOW);
  digitalWrite(LED_4, LOW);
  delay(500);

  digitalWrite(LED_1, HIGH);
  digitalWrite(LED_2, HIGH);
  digitalWrite(LED_3, HIGH);
  digitalWrite(LED_4, HIGH);
  delay(500);
}

//Read input
int knockA = analogRead(SensorA);
int knockB = analogRead(SensorB);
int knockC = analogRead(SensorC);
int knockD = analogRead(SensorD);

delay(100);


Serial.print("SensorA:  ");
Serial.print(knockA);
Serial.print("       SensorB:  ");
Serial.print(knockB);
Serial.print("       SensorC:  ");
Serial.println(knockC);

//headpattern 1-A 2-B 3-C 4-D

bool hit=false;
int heardPattern =0;
if(knockA >= directThreshold) {
  heardPattern = 1;
  hit=true;
}else if(knockB >= directThreshold) {
  heardPattern = 2;
  hit=true;
}else if(knockC >= directThreshold) {
  heardPattern = 3;
  hit=true;
}


//FSM - Transition of States
switch (state){
```

```
    case LOCKED_IDLE:
      state = KNOCK_1;
      break;

    case KNOCK_1:
      message1();
      if (hit ==true){
        if(heardPattern == Sequence[0]){
          hit=false;
          delay(150);
          message2();
          state = KNOCK_2;
        }else{
          message4();
          hit = false;
          state = LOCKED_IDLE;
        }
      }
      break;

    case KNOCK_2:
      if (hit ==true){
        if(heardPattern == Sequence[1]){
          hit=false;
          delay(150);
          message2();
          state = KNOCK_3;
        }else{
          message4();
          hit = false;
          state = LOCKED_IDLE;
        }
      }
      break;

    case KNOCK_3:
      if (hit ==true){
        if(heardPattern == Sequence[2]){
          message3();
          hit=false;
          delay(150);
          state = UNLOCKED_IDLE;
        }else{
          message4();
          hit = false;
          state = LOCKED_IDLE;
        }
      }
      break;

    case UNLOCKED_IDLE:
      //state = LOCKED_IDLE;
      break;

    default:
      state = LOCKED_IDLE;
      break;
  }



  //FSM - Output of States
  switch (state){

    case LOCKED_IDLE:
      digitalWrite(LED_1, LOW);
      digitalWrite(LED_2, LOW);
      digitalWrite(LED_3, LOW);
      digitalWrite(LED_4, LOW);
```

```
      break;

    case KNOCK_1:
      digitalWrite(LED_1, HIGH);
      digitalWrite(LED_2, LOW);
      digitalWrite(LED_3, LOW);
      digitalWrite(LED_4, LOW);


      break;

    case KNOCK_2:
      digitalWrite(LED_1, HIGH);
      digitalWrite(LED_2, HIGH);
      digitalWrite(LED_3, LOW);
      digitalWrite(LED_4, LOW);
      break;

    case KNOCK_3:
      digitalWrite(LED_1, HIGH);
      digitalWrite(LED_2, HIGH);
      digitalWrite(LED_3, HIGH);
      digitalWrite(LED_4, LOW);
      break;

    case UNLOCKED_IDLE:
      for (int i = 0;i<10;i++){
      digitalWrite(LED_1, HIGH);
      digitalWrite(LED_2, LOW);
      digitalWrite(LED_3, HIGH);
      digitalWrite(LED_4, LOW);
      delay(100);
      digitalWrite(LED_1, LOW);
      digitalWrite(LED_2, HIGH);
      digitalWrite(LED_3, LOW);
      digitalWrite(LED_4, HIGH);
      delay(100);
      }

      break;

    default:
      state = LOCKED_IDLE;
      break;
  }
}

void RecordPattern(int *Pattern){
  message5();
  int j=0;
    while(j<size_array){
    int knockA = analogRead(SensorA);
    int knockB = analogRead(SensorB);
    int knockC = analogRead(SensorC);
    delay(100);


    bool hit=false;
    int heardPattern =0;

    if(knockA >= directThreshold) {
      heardPattern = 1;
      hit=true;
      delay(150);
    }else if(knockB >= directThreshold) {
      heardPattern = 2;
      hit=true;
      delay(150);
    }else if(knockC >= directThreshold) {
      heardPattern = 3;
```

```
      hit=true;
      delay(150);
    }

    if ((hit==true)){
      *Pattern=heardPattern;
      Serial.print(*Pattern);
      j++;
      *Pattern++;
      digitalWrite(LED_R,LOW);
      delay(500);
      digitalWrite(LED_R,HIGH);
    }
  }
message6();
}

void message1(){
  digitalWrite(LCD_1, LOW);
  digitalWrite(LCD_2, LOW);
  digitalWrite(LCD_3, HIGH);
}

void message2(){
  digitalWrite(LCD_1, LOW);
  digitalWrite(LCD_2, HIGH);
  digitalWrite(LCD_3, LOW);
}

void message3(){
  digitalWrite(LCD_1, LOW);
  digitalWrite(LCD_2, HIGH);
  digitalWrite(LCD_3, HIGH);
}

void message4(){
  digitalWrite(LCD_1, HIGH);
  digitalWrite(LCD_2, LOW);
  digitalWrite(LCD_3, LOW);
}

void message5(){
  digitalWrite(LCD_1, HIGH);
  digitalWrite(LCD_2, LOW);
  digitalWrite(LCD_3, HIGH);
}

void message6(){
  digitalWrite(LCD_1, HIGH);
  digitalWrite(LCD_2, HIGH);
  digitalWrite(LCD_3, LOW);
}
```

9.3.2 Microcontroller B

```
// include the library code:
#include <LiquidCrystal.h>    //in order to operate the LCD screen we need to call a specific library
bool first=true; // defining a boolean variable to be used to determined whether the door should be unlocked
or not

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup(){
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.clear();    // setting initial message to be displayed on screen
  lcd.setCursor(0,0);
  lcd.print("WELCOME TO");
  lcd.setCursor(0,1);
  lcd.print("KNOCKLOCK");
  delay(1000);


  Serial.begin(9600);
}

void loop() {

  bool A=digitalRead(8);
  bool B=digitalRead(9);
  bool C=digitalRead(10);  //declaring variables A,B,C to read digital pins 8,9 and 10
                           // Two values are available either 0 or 1, LOW or HIGH

  Serial.print("SensorA:  ");
  Serial.print(A);
  Serial.print("       SensorB:  ");
  Serial.print(B);
  Serial.print("       SensorC:  ");
  Serial.println(C);

  delay(200);
  if((A==LOW)&&(B==LOW)&&(C==HIGH)){
    lcd.clear();
    lcd.print("PLEASE KNOCK");     //different combinations of HIGHs and LOWs represent
                                   //different messages
  }


  else if((A==LOW)&&(B==HIGH)&&(C==LOW)){
    lcd.clear();
    delay(1);
    lcd.setCursor(0,0);
    lcd.print("KNOCK IDENTIFIED");


  }else if((A==LOW)&&(B==HIGH)&&(C==HIGH)){
    if (first){ // in these cases when the FSM is at its final state ready to be unlocked, the boolean
variable
      lcd.clear(); //is set to false after it recognised the final knock and a message of DOOR UNLOCKED is
displayed
      lcd.print("KNOCK IDENTIFIED");
      delay(1000);
      first=false;
    }
    lcd.clear();
    lcd.print("DOOR UNLOCKED");

  }else if((A==HIGH)&&(B==LOW)&&(C==LOW)){
    lcd.clear();
    lcd.print("WRONG KNOCK");
    delay(1000);
```

```
  }else if((A==HIGH)&&(B==LOW)&&(C==HIGH)){
    lcd.clear();
    lcd.setCursor(0,0); // using the setCursor command we are able to write on both lines of the LCD
    lcd.print("START RECORDING");
    lcd.setCursor(0,1);
    lcd.print("NEW PATTERN");
    delay(1000);
  }
  else if((A==HIGH) && (B ==HIGH) && (C ==LOW)){
    lcd.clear();
    lcd.print("RECORDING ENDS");
  }
}
```