**Smart Socket**

**Final year project**

**By Amar Essa 05157706**

**2014/2015**

**Supervised by:**

**Dr Peter Plassmann and Dr Emlyn Everitt.**

Contents

[1. Introduction 4](#_Toc414812693)

[1.1 Background 4](#_Toc414812694)

[1.2 Aim 4](#_Toc414812695)

[1.3 Objectives 4](#_Toc414812696)

[1.3.1 Technical objectives 4](#_Toc414812697)

[1.3.2 Project management objectives 5](#_Toc414812698)

[1.4 Initial time plan 6](#_Toc414812699)

[1.4.1 Project Gantt chart 6](#_Toc414812700)

[1.5 Deliverables 7](#_Toc414812701)

[2. Research 7](#_Toc414812702)

[2.1 Technical research 7](#_Toc414812703)

[2.1.1 Hardware platform (Electronics) 7](#_Toc414812704)

[2.1.2. Software 20](#_Toc414812705)

[2.1.3 Mobile application IDE 22](#_Toc414812706)

[2.1.4 3D Printer design application 25](#_Toc414812707)

[2.2 Summary 27](#_Toc414812708)

[2.3 Comparable devices 28](#_Toc414812709)

[3. Design 29](#_Toc414812710)

[3.1 Preliminary design 29](#_Toc414812711)

[3.2 Boards schematics 30](#_Toc414812712)

[3.2.1 Power converter schematics 30](#_Toc414812713)

[3.3 Software Design 31](#_Toc414812714)

[3.4 Final design 35](#_Toc414812715)

[3.5 Graphical user interface 37](#_Toc414812716)

[3.6 Test plan 38](#_Toc414812717)

[3.6.1 Software testing 38](#_Toc414812718)

[3.6.2 Hardware testing 38](#_Toc414812719)

[3.7 3D Printed components 38](#_Toc414812720)

[4. Implementation 43](#_Toc414812721)

[4.1 Hardware implementation 43](#_Toc414812722)

[4.1.1. Electronics 43](#_Toc414812723)

[4.1.2 Case 46](#_Toc414812724)

[4.2 Software implementation 47](#_Toc414812725)

[4.3 User instructions: 48](#_Toc414812726)

[4.4 Financial cost of producing the device. 49](#_Toc414812727)

[5. Validation 49](#_Toc414812728)

[5.1 Testing 49](#_Toc414812729)

[5.1.1 Hardware 49](#_Toc414812730)

[5.1.2 Software 49](#_Toc414812731)

[5.2 Evaluation 51](#_Toc414812732)

[6.0 Discussion 52](#_Toc414812733)

[6.1 Security 52](#_Toc414812734)

[6.2 Compatibility 52](#_Toc414812735)

[6.3 Use 52](#_Toc414812736)

[6. 4 Communication 52](#_Toc414812737)

[6.5 Safety 53](#_Toc414812738)

[6.6 legal requirements 53](#_Toc414812739)

[6.7 functions 53](#_Toc414812740)

[7. Conclusion 53](#_Toc414812741)

[7.1 achievement of project aim 53](#_Toc414812742)

[7.2 future work 53](#_Toc414812743)

[References 54](#_Toc414812744)

[Bibliography 57](#_Toc414812745)

[Appendix 58](#_Toc414812746)

1. Introduction

1.1 Background

Electrical wall sockets in houses have been used almost ever since electricity was invented. These sockets are mainly controlled by any household member by switching them ON/OFF manually from within the house.

**Problem**

Manually controlled sockets pose a problem to those who would like to switch an electric device ON/OFF from an external location from the house. Enabling the control of the electrical socket in the house from a remote location remains a challenge.

**Proposed solution**

The most obvious solution is to devise a circuit that is controlled remotely by a device

**Anticipated benefits**

The means to control electrical sockets from a remote location could be used both in industrial and residential sectors.

Examples of usage:

1. Shop owners can use the device to control advertising lights remotely from their shop.
2. A household member can turn the heater on in his/her house remotely before he/she arrives home from work to a warm house.
3. A household owner or a business owner can switch an alarm OFF/ON remotely.

1.2 Aim

The aim of this project is to research, design, implement and assess a device that control an electric socket remotely.

1.3 Objectives

1.3.1 Technical objectives

* To propose potential solutions for the following requirements:
* **Security**

The device must be secured and measures should be taken to ensure that no access to third party is allowed other than the intended user operating the device remotely.

* **Functions**
* Turn each of the sockets ON or OFF.
* Turn all the sockets ON or OFF.
* Retrieve the devices status.
* Enter the phone number that controls the device.
* **Compatibility**

The device must be compatible to electrical current used both in Europe, Asia and the USA so that it can be used almost in any country of the world.

* **Usage**

The device must be controlled by a mobile phone. Text messaging is the preferred communication mechanism.

A mobile application operating on Android operating system with the following functions:

* Turn each of the sockets ON or OFF.
* Turn all sockets ON or OFF.
* Retrieve the devices status.
* **Communication**

The device should communicate with the user, it must relay information of the status of the switch whenever the user request it.

* **Safety**

The device must also be safe to be used by the public.

* **Legal requirement**

The device must meet all legal requirement and have the following tests:

* PAT (Portable appliance testing) test.
* Ce Marking.

1.3.2 Project management objectives

* Create a project plan and review the plan against progress at 2 weeks intervals.
* Write a dissertation as per project guidelines.
* To design, implement, test and evaluate a device.

1.4 Initial time plan

1.4.1 Project Gantt chart

The project Gantt chart in figure 1.1 is a preliminary planned Gantt chart that shows the weekly plan of the project and the time it takes to undertake each task, further notes are included below the chart to explain each task.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | | |  |  |  |  |  |
|  |  |  |  |  | | |  |  |  |  |  |
| **Week** | **0** | **2** | **4** | **10** | **12** | **14** | **16** | **18** | **20** | **24** |  |
| **Research** |  |  |  |  |  |  |  |  |  |  |  |
| **plan** |  |  |  |  |  |  |  |  |  |  |  |
| **Design** |  |  |  |  |  |  |  |  |  |  |  |
| **implementation** |  |  |  |  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |  |  |  |  |
| **Adjustments** |  |  |  |  |  |  |  |  |  |  |  |
| **Evaluation** |  |  |  |  |  |  |  |  |  |  |  |
| **Dissertation** |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

Gantt .1 Project Gantt chart

**Gantt chart notes:**

**Research:** all the research that have to be undertaken to build the device and have a clear idea of its viability, usability, need and cost.

**Plan:** to prepare and devise a plan of action.

**Design:** to design all elements of the project and device.

**Implementation:** to implement the plan.

**Testing:** to test all aspects of the device and related aspects (code).

**Adjustments:** to adjust and solve any issues that arise from the testing process.

**Evaluation:** to evaluate the project by third parties.

**Dissertation:** to write a dissertation for the project.

1.5 Deliverables

**User manual**

* Using the device without the mobile application.
* Using the device with the mobile application.
* Step by step instructions on how to use the smart socket.

**Device**

* Deliver a working device.
* Smart phone application.
* Dissertation (incl. validation documents)

2. Research

This chapter provides a critical examination of the literature about the components and software needed to build the smart socket.

2.1 Technical research

The smart socket device consists of hardware and software components that perform the following tasks:

1. Control the circuit.
2. Programme that processes the network signal and issues commands.
3. Electronics to switch the sockets ON or OFF.
4. Circuit to convert and regulate the electronics power.
5. Circuit that connects to the mobile GSM network.
6. Programme that builds the user interface.

2.1.1 Hardware platform (Electronics)

1. **Micro controllers:**

A microcontroller or CPU (central processing unit) is needed to control the circuits and achieve the desired functions of the device. The following section suggests the available options on the market.

1. **Arduino Uno**

Arduino Uno is a microcontroller board that includes memory and a series of digital and analogue pins that can be controlled using c based language interface; it has been used in a multitude of projects and is extensively used as a hobby. [Arduino.cc, 2014]

**Advantages**

* + - Low cost.

The Arduino ranks as the cheapest microcontroller board in comparison with the raspberry pi and Beagle bone. According to [Allan, 2013], it is also the cheapest board that can be obtained from one of the largest product comparable web site online. [Google.co.uk, 2014]

* + - Ease of use to developers.

Many assessments of the boards mentioned in this research have been undertaken by industry professionals, as the tech watch report stated when a user would like to utilise one of these boards

“*If you want to play it safe, you should probably go with an Arduino or a Raspberry Pi.”* [Techwatch.keeward.com, 2014]

And as mentioned in the design spark report on Arduino vs raspberry pi

“*Thanks to Linux the Raspberry Pi benefits from a far more flexible and powerful development environment. However, the Linux kernel alone comprises millions of lines of code and for applications where simplicity is key the operating system-less Arduino holds certain appeal.*“ [Rs-online.com, 2012]

* + - Array of add on shields

Many shields have been developed to work in conjunction with Arduino boards, these shields come with software libraries that can be used to control and utilise their functions. [Arduino.cc, 2014]

* + - Has a programming interface.

The Arduino Uno is supported by an IDE that simplify the programming process of the built in components. [Arduino.cc, 2014]

* + - Built in memory.

The Arduino Uno has a Flash Memory 32 KB (ATmega328) of which 0.5 KB used by boot loader and an SRAM 2 KB (ATmega328). [Arduino.cc, 2014]

**Disadvantages**

* + - Limited memory.

The mere Flash Memory of 32 KB forces the programmer to build smaller programmes that can be loaded onto the Arduino board. [Arduino.cc, 2014]

* + - Limited pins on board

The Arduino Uno has Digital I/O Pins 14 (of which 6 provide PWM output) and 6 analogue Input Pins, these low pin count makes incorporating the Arduino Uno with other circuits difficult. [Arduino.cc, 2014]

1. **Raspberry Pi**

The Raspberry Pi is a credit card sized computer board that is used for electronics projects and a variety of computer related tasks. It is also widely used in the hobby industry and primarily chosen for its computational power and small size in experimental projects. [Raspberrypi.org, 2014]

**Advantages**

* + - Can be programmed by many languages.

As the writer stated

“*Being a Linux computer, it supports basically tens if not hundreds of programming languages, you just have to install the support you need if it's not installed. besides Python, Java, C/C++ you may use PHP, bash scripting, AWK, assembly language, LUA, JS scripting, HTML/CSS (is it a programming language?), Clojure, Haskell, Erlang, Forth, LUA, Pascal, TeX, SQL, Ruby, Smalltalk and so on... it's by no means a complete list.*” [Stack, 2014]

* + - Large memory on-board.

As stated by [Allan, 2013] the raspberry pi is amongst few boards that has a flash memory slot , it has a scalable SD card slot that is used as flash memory when installed with an SD card meaning that any amount of memory can be installed on it according to the sizes of SD cards available on the market.

**Disadvantages**

* + - No specific programming interface.

According to [MakeUseOf, 2013] raspberry pi can be used with 7 operating systems (Linux, openELC & XBMC, retro pi, RISC OS, Fire fox, Plan 9, android and bonus pip boy) all the mentioned operating systems except LINUX need a further IDE’S installed on them so the raspberry pi can perform electronic tasks thus complicating the process further.

* + - Limited compatible GSM shields or circuits.

The raspberry pi community have not developed and produced a GSM shield compatible with the raspberry pi according to a list of raspberry pi compatible devices [Elinux.org, 2014].

* + - Limited I/O pins.

The raspberry pi has 26 pins, 17 of the 26 are GPIO pins and the remaining are power or ground pins, these pins can be used to connect devices and circuits to the raspberry pi. Although 27 might seem excessive, several other devices might need to be connected to the raspberry pi to achieve the tasks required by this project that is requiring almost double the amount of pins the raspberry pi offer. [Raspberrypi.org, 2014]

1. **ODROID XU3**

ODROID is a powerful and small Linux operated computer that can be used to control electronic boards and preform most tasks a normal computer could perform. It’s mainly used in the hobby, research and developments industries. [Hardkernel.com, 2014]

**Advantages**

* + - Advanced processing power.

The ODROID has the most advanced processing power amongst the microprocessors boards. According to [Hardkernel.com, 2014] it supports The Heterogeneous Multi-Processing (HMP) Octa Core Linux Computer Samsung Exynos5422 Cortex-A15 2.0Ghz quad core and Cortex-A7 quad core.

* + - Large memory on board.

The board supports 2Gbyte LPDDR3 RAM at 933MHz (14.9GB/s memory bandwidth) PoP stacked according to [Hardkernel.com, 2014] which is exceptional amongst microprocessors boards.

**Disadvantages**

* + - Few compatible shields and circuits.

A handful of compatible shields or devices is available for the ODROID [Hardkernel.com, 2014]

* + - Expensive.

At $179 the ODROID is an expensive board in the market [Slashgear.com, 2014].

* + - Limited I/O pins.

The ODROID XU3 support 30 IO expansion board which is not enough to support multiple external devices concurrently [Hardkernel.com, 2014].

1. **Decision**

Due to the amount of I/O pins needed, cost and compatible circuits or shields needed, the ARDUINO is the most beneficial choice of micro controller or microprocessors circuits to be used in this project.

**2. GSM circuit**

GSM circuits are used to communicate with a GSM mobile network. These circuits are used to obtain commands from the user and send back reports to a GUI.

1. **Arduino GSM Shield**

The Arduino GSM shield is a circuit developed by Telefonica of Spain to connect a GSM network to the Arduino boards. An extensive software library was developed alongside it that gives developers the ability to send or receive text messages and voice calls. [Arduino.cc, 2014]

**Advantages**

* + - Compatible with Arduino Uno.

The Arduino GSM shield is compatible with the Arduino Uno board and is available with compatible software. The GSM shield fits on the Arduino board and provides an extension to the IO pins. [Arduino.cc, 2014]

* + - Programming library included.

The Arduino provides a compatible software library to the Arduino gsm shield and the library include many functions that sends commands to the integrated circuit on the GSM board and instigates different GSM functions. [Arduino.cc, 2014]

* + - Facilitates sending and receiving of text messages and calls.

The Arduino GSM shield can send and receive text messages and calls. [Arduino.cc, 2014]

**Disadvantages**

* + - Expensive.

The Arduino GSM shield retails at 69 euros and can be bought from [Arduino.cc, 2014]. It is expensive in comparison to other GSM shields on the market.

1. **SIM900 Quad-band GSM/GPRS Shield**

The SIM900 shield allows the user to communicate with a GSM network and utilise its services. The shield can send and receive text messages and GPS data to the user. A limited software library was used to support this shield by its manufacturer [Seeedstudio.com, 2014].

**Advantages**

* + - Compatible with Arduino Uno.

The SIM900 is also compatible with the Arduino boards. It is privately manufactured by independent companies to the Arduino organisation and the device fits on the Arduino board and provides an extension to the IO pins. [Seeedstudio.com, 2014]

* + - Facilitates receiving of text messages and calls.

The SIM900 can receive text messages and calls when connected to the GSM network with a SIM card. [Seeedstudio.com, 2014]

* + - Low cost.

The SIM900 is low in cost when compared to the Arduino GSM shield and retails at £20.26. [Seeedstudio.com, 2014]

**Disadvantages**

* + - No programming libraries included.

The SIM900 does not come with any software libraries but with ATI commands that is delivered straight to the integrated circuit on board and in turn executes functions on the GSM network. [Cooking-hacks.com, 2014]

* + - Limited buffering.

The SIM900 has a limited buffering on the board that makes it difficult to receive text messages. [Cooking-hacks.com, 2014]

**II. Decision**

The requirement for a supporting programming language coupled with the requirement to send and receive messages makes the Arduino GSM shield a better choice to support the functions of this project.

1. **Relay circuit**

A relay circuit is used to control an electric flow to another circuit or device. It acts as a switch that turns something on or off.

1. **Arduino relay circuit**

The pre-manufactured Arduino relay 4 channel circuits are manufactured by many companies. These circuits connect to the Arduino board utilising a common ground and 5v input from the Arduino. They are controlled by the Arduino pins and the Arduino relay supports a voltage of 110/240. Therefore, theoretically it can be used in countries that support either of the voltages mentioned. [Arduino-info.wikispaces.com, 2014]

**Advantages**

* + - Ready to use.

The Arduino relay is made to fit and can be connected to the Arduino board in few simple connections. One connection to each relay channel and two connections to the relay power supply. [Arduino-info.wikispaces.com, 2014]

* + - Low cost.

In comparison to all the available relay shields on the market the Arduino relay shield is the most affordable at around £6.[Google.co.uk, 2014]

* + - Low power consumption.

The Arduino shield requires a dc 5v supply from an external source making it a low consumption device. [Arduino-info.wikispaces.com, 2014]

* + - Supports 240v and 110v.

The relays on the Arduino relay shield can support a voltage from 100v to 240v respectively making it usable in different countries around the world. [Google.co.uk, 2014]

* + - Arduino compatible.

The Arduino relay shield was made specifically to be compatible with the Arduino Uno [Arduino-info.wikispaces.com, 2014].

**Disadvantage**

* + - Large size.

The Arduino relay shield is relatively large in size in comparison to the Arduino Uno board both can be compared at [Arduino-info.wikispaces.com, 2014] and [Arduino.cc, 2014]

1. **Manually built circuit**

A relay circuit could be built manually using available components and designs from the internet. These available designs could be modified according to specifications and requirements. [Qsl.net, 2014]

**Advantages**

* + - Low power consumption.

The relay board can be built to consume 3v to 5v dc input according to the schematics. . [Qsl.net, 2014]

* + - Supports 240v and 110v.

The relay board can also utilise relays that can support both 110v and 240v making it compatible with different electricity home power supplies around the world. . [Qsl.net, 2014]

**Disadvantage**

* + - Time consuming.

The circuit would take a few days to build from sourced components thus making it expensive in terms of time.

* + - Expensive to build.

The components needed to build the circuit would cost more than buying the readymade Arduino compatible relay circuit sold on the market. [Google.co.uk, 2014]

1. **Decision**

The Arduino relay board has an advantage in cost, use and compatibility with the other circuits making it the best choice for this project.

1. **Converter and regulating circuit.**

The need for extra power to support the Arduino shields and used circuits are crucial due to the limited amount of power the Arduino can output to supported circuits.

1. **Readymade volt converter/regulator**

Low voltage dc to dc converters and regulators that produce different voltages are available on the market. These circuits can support the Arduino board and its associated shields as required. [Linear.com, 2014]

**Advantages**

* + - Ready to use.

The readymade volt converter/regulator comes assembled and ready to use, it outputs a voltage range from 2v to 8v respectively. [Linear.com, 2014]

* + - Accurate.

The converter/regulator is accurate in terms of the output it produces, the circuits have been tested before manufacturing. [Linear.com, 2014]

* + - Low cost.

The cost of the converter/regulator is low in comparison to the others similar circuits available on the market. [EBay, 2014]

**Disadvantage**

* + - No pin rail.

The converter/regulator has no pin rail and one has to be made to fit to the device. [Linear.com, 2014]

1. **In house built converter/regulator.**

A converter/ regulator can be build using available components supported with readymade designs and tutorials found on the internet. [X-relsemi.com, 2014]

**Advantages**

* + - Low cost.

The converter/regulator components are fairly inexpensive and widely available on the market. The overall cost of building the circuit from existing designs are affordable. [X-relsemi.com, 2014]

* + - Built in pin rail.

A power rail could be built into the design making the converter/regulator power output easier to tap into as needed.

**Disadvantage**

* + - Time consuming to build.

The circuit would take a day to build from sourced components making it expensive in terms of time

1. **Decision**

This project requires a power converter and regulator that must have a pin rail to provide the other circuits with power and must not have a variable power regulation and conversion mechanism. Thus, the in house built power converter/ regulator is the best choice for this project.

1. **Liquid Crystal Screen**

A LCD is used on the smart socket to provide the user with the information of the state of the machine and prompts the user with different requests as needed.

1. **HD44780 LCD**

**Advantages**

* Low cost

The HD44780lcd is the cheapest LCD Arduino compatible device on the market [AVR, 2014].

**Disadvantage**

* No serial chip

A serial chip is necessary to reduce the amount of pins needed to control the device and also reduce the amount of programming needed to control the LCD screen [AVR, 2014].

1. **Serial IIC/I2C/TWI 2004 204 20X4 Character LCD**

**Advantages**

* Includes serial chip

To reduce the amount of pins needed to control the device and also reduce the amount of programming needed to control the LCD screen [UK, 2014]

**Disadvantage**

1. It is Larger in comparison HD44780 LCD

The serial LCD is larger due to the fact that the electronic board need space for extra components and chip. [UK, 2014]

1. **Decision**

Due to the fact that an LCD needs a considerable amount of programming if it does not include a serial chip the Serial IIC/I2C/TWI 2004 204 20X4 Character LCD is the best option for this project.

2.1.2. Software

Software is needed to control the micro controller and also needed for designing the user interface of the device.

1. **Java**

Java is a programming language created by Oracle Corporation and used widely in many industries including microcomputers. [Oracle.com, 2014]

**Advantages**

* + - Used in designing mobile applications (android studio).

*“Android apps are written in the Java programming language.”* [Fundamentals, 2014]

* + - Extensive libraries provided.

The java programming language has been around for decades thus extensive libraries have been developed for it. [Oracle.com, 2014]

**Disadvantages**

* + - Cannot be used with the Arduino micro controller.

The Arduino GUI use C language to dump code to the microcontroller and therefore, java cannot be directly used with the Arduino boards. [Arduino.cc, 2014]

* + - Not primarily used for embedded systems.

Java is used by many platforms, however, the embedded systems languages usually used are VHDL, VERILOG, C and assembly. [Bamafolks.com, 2014]

1. **C**

The C programming language was developed by Dennis Ritchie at AT&T bell labs between 1969 and 1973, it is a widely used language in embedded systems. [Www2.its.strath.ac.uk, 2014]

**Advantages**

* + - Used extensively in embedded systems.

The C programming language has been used extensively by embedded systems since its creation and to date. [Bamafolks.com, 2014]

* + - Used by the Arduino programming interface.

The Arduino IDE uses C programming language to dump code into the microprocessor on the board. [Arduino.cc, 2014]

**Disadvantages**

* + - Cannot be used to design mobile applications.

Two platforms are used to develop android compatible applications (android studio and Eclips). These two IDE’s utilise the java programming language as its core programming language. [Fundamentals, 2014]

* + - Memory oriented programming language (low level).

The C language is a language that is more compatible to machine code in its design and operations and according to [Taral, 2012] other languages, it uses greater abstraction when utilising machine code than C and assembly language.

1. **Python**

The python programming language was designed by Guido van Rossum to enable a the user to use fewer lines of code to perform an operation, it is regarded as a high level programming language, it is widely used alongside Linux on microcomputers and in general purpose software’s.[Python.org, 2014]

**Advantages**

* + - High level programming language that provides extensive libraries for embedded systems.

Python can be used for embedded systems as well as other systems and technologies. [Python.org, 2014]

**Disadvantages**

* + - Not compatible with the Arduino programming interface.

The Arduino GUI use C language to dump code to the microcontroller. Therefore, java cannot be directly used with the Arduino boards. [Arduino.cc, 2014]

* + - Primitive in the field of mobile applications programming paradigm.

Two platforms are used to develop android compatible applications, (android studio and Eclips) these two IDE’s utilise the java programming language as its core programming language. [Fundamentals, 2014]

1. **Decision**

The requirement for a user interface makes java the best language to use for building a mobile application and another requirement for a language compatible with Arduino programming interface makes C the prime language to use.

2.1.3 Mobile application IDE

The android mobile application development process requires an integrated development environment to produce a satisfactory product. Below are a few options that could be used to produce an android mobile application.

1. **Eclipse**

Although the eclipse development environment can be used to develop software in many languages, it is primarily used for developing in the Java programming language. It is also used to develop android mobile applications using a separate add on plugin to the actual development environment. [Guindon, 2014]

**Advantages**

* Drag and drop GUI(graphical user interface) Designer

The drag and drop interface gives the user the ability to draw buttons and many other tools simply by dragging and dropping them into the window. [Cogswell, 2014]

* Google cloud platform

Google Cloud Platform allows one to run server-side code using Google App Engine and it can also be utilised to store user information on a google server. [Cogswell, 2014]

**Disadvantages**

* Large application

Eclipse is a large application built with many libraries and functionality that makes it complex to figure out. [Cogswell, 2014]

* Legacy software

Eclipse is old and have been around for many years and the interface feels and looks old. [Cogswell, 2014]

* Hard to understand

Due to the amount of tasks and languages supported by eclipse, the application is hard to understand and a user would quickly lose focus. [Cogswell, 2014]

1. **Android Studio**

Android Studio is a software or IDE (integrated development environment) developed primarily to build android mobile applications. It is fairly new to the IT industry and is not capable of producing any other applications other than applications used with the android operating system. [Editor, 2014]

**Advantages**

* Drag and drop GUI Designer

The drag and drop interface gives the user the ability to draw buttons and many other tools simply by dragging and dropping them into the window. [Cogswell, 2014]

* Complete IDE Designed for Android applications

The Android studio has been specifically made to develop android mobile applications thus giving it an advantage if one is wishing to build Android mobile applications. [Cogswell, 2014]

* Code completion.

The Android studio code completion utility completes any code you are typing and gives you auto suggestions you can chose from. [Cogswell, 2014]

* Gradle integration

Gradle is a build system that can take different libraries from an array of programming languages and add them to your android project. This feature is an advanced build system feature. [Cogswell, 2014]

* Google cloud platform

Google Cloud Platform allows one to run server-side code using Google App Engine and it can also be utilised to store user information on a google server. [Cogswell, 2014]

**Disadvantages**

* Limited learning materials

Android studio is fairly new and not much learning material is available online or the market place, the main source of learning material is the android studio website. [Editor, 2014]

1. **Decision**

Although eclipse has a large amount of functionality and ability, it is cumbersome and complex. Hence android studio would be the best choice to build an android application for the planned device.

2.1.4 3D Printer design application

The 3D printer design application gives the designer the ability to design a product in 3 dimensional axe’s and produce a binary file to be executed and produced by a 3D printer. The final product quality relies heavily on the 3D design application used in the development process. Below is the review of two of the commonly free 3d design software used on the market. [3ders.org, 2014)]

1. **Sketchup**

Sketchup was developed by google to give a 3d designer a quick and simple designing application to use, it’s a ready to use application. [Sketchup.com, 2014]

**Advantages**

* Large amount of drawing tools for simple designs

SketchUp provides large amount of tools for simple designs in its application with simple features and abilities. [Sketchup.com, 2014]

* Supports STL format.

SketchUp gives the user the ability to convert a design to many formats. One of these formats is STL which is used by most 3D printers to produce 3d designs. [Sketchup.com, 2014]

**Disadvantages**

* No repair tool.

To print a 3d object from an STL file, user must first clean the design from any unprintable parts. The repairing tool normally cleans the design for the user before printing. [Sketchup.com, 2014]

1. **Netfab**

Netfab is a 3d design and repair application that is widely used in 3d printing industry. [GmbH, 2014]

**Advantages**

* Supports STL format.

SketchUp gives the user the ability to convert the design to many formats. One of these formats is STL which is used by most 3D printers to produce 3d designs.

[GmbH, 2014]

* Repair tool

To print a 3d object from an STL file, user must first clean the design from any unprintable parts. The repairing tool normally cleans the design for the user before printing. [GmbH, 2014]

**Disadvantages**

* Complex tools

Netfabb is mostly used by professional and seasoned designers in 3D printing industry. The tools used in the application are extensive and would take time and experimenting to understand. [GmbH, 2014]

**V.I Decision**

Both the above mentioned applications can produce the designs needed. The Netfabb is crucially needed for the repairing process and Sketchup is needed for the preliminary design. Therefore, both should be used in this project.

2.2 Summary

Summary of the decisions to use the hardware and software needed for this project is provided in the table given below. The table shows a list of the choices which have been decided on the basis of the decisions mentioned in the above in review.

|  |  |
| --- | --- |
| Choice | Use |
| Hardware | Hardware to be used |
| Arduino Uno | Micro controller to control circuits |
| Arduino GSM Shield | GSM chip circuit to connect to network |
| Arduino Relay | Relays circuit to switch sockets ON/OFF |
| In house built power converter/regulator | To convert/regulate and give power to circuits |
| Software |  |
| Java | To produce a user interface (mobile application) |
| C | To programme the Arduino micro controller |
| Android Studio | To develop the mobile application |
| Sketch up | To develop the used parts |
| Netfabb | To repair the design |
| Serial IIC/I2C/TWI 2004 204 20X4 Character LCD | Will be used as a display |
| Keypad | Will be used as a Keypad |

*Table 2.1 summery of Hardware and Software to be used.*

2.3 Comparable devices

The concept of smart GSM sockets is fairly new in the industry and only few companies have attempted to build and sell them. The Australian companies that have succeeded in bringing the device to the market but for a much higher retail price than their Chinese counterparts. Therefore, Chinese companies managed to dominate the market with their product due to the price difference. Below are the comparable devices to the smart socket in this report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Country** | **Price** | **Reference** | **Picture** |
| Cell phone GSM Remote Control 3 Sockets Smart Socket Power Switch | Sold from the USA , made in china | £24.60 | [Banggood.com, 2014)] |  |
| GSM Cell Phone Remote Control Switch Electric Wireless Smart Socket | Sold from the USA , made in china | £16.97 | [Banggood.com, 2014] |  |
| iSocket GSM 706 | Sold online from the link in the reference and made in china but designed in Australia | 89 EURO | [Isocket.eu, 2014] |  |

*Table 1**2.2 Comparable Device*

3. Design

3.1 Preliminary design

The preliminary design in schematic No.3.1 show how the circuits are connected to each other. Problems might arise of implementing this preliminary design. It shows the connections established between the Arduino Uno, GSM shield, key pad LCD and relays shield with the external power supply provided.



*Schematics 3.1 electronics preliminary Design*

3.2 Boards schematics

3.2.1 Power converter schematics

The power converter in this schematics No 3.2 show the use of rails to convey the power to the other circuits that need more than what the Arduino Uno can provide, it also feeds the power to the LED lights used in the device, an LM7805 [Fairchildsemi.com, 2014] was used to step down the voltage and create a regulated 5v dc source from a 9v dc source.

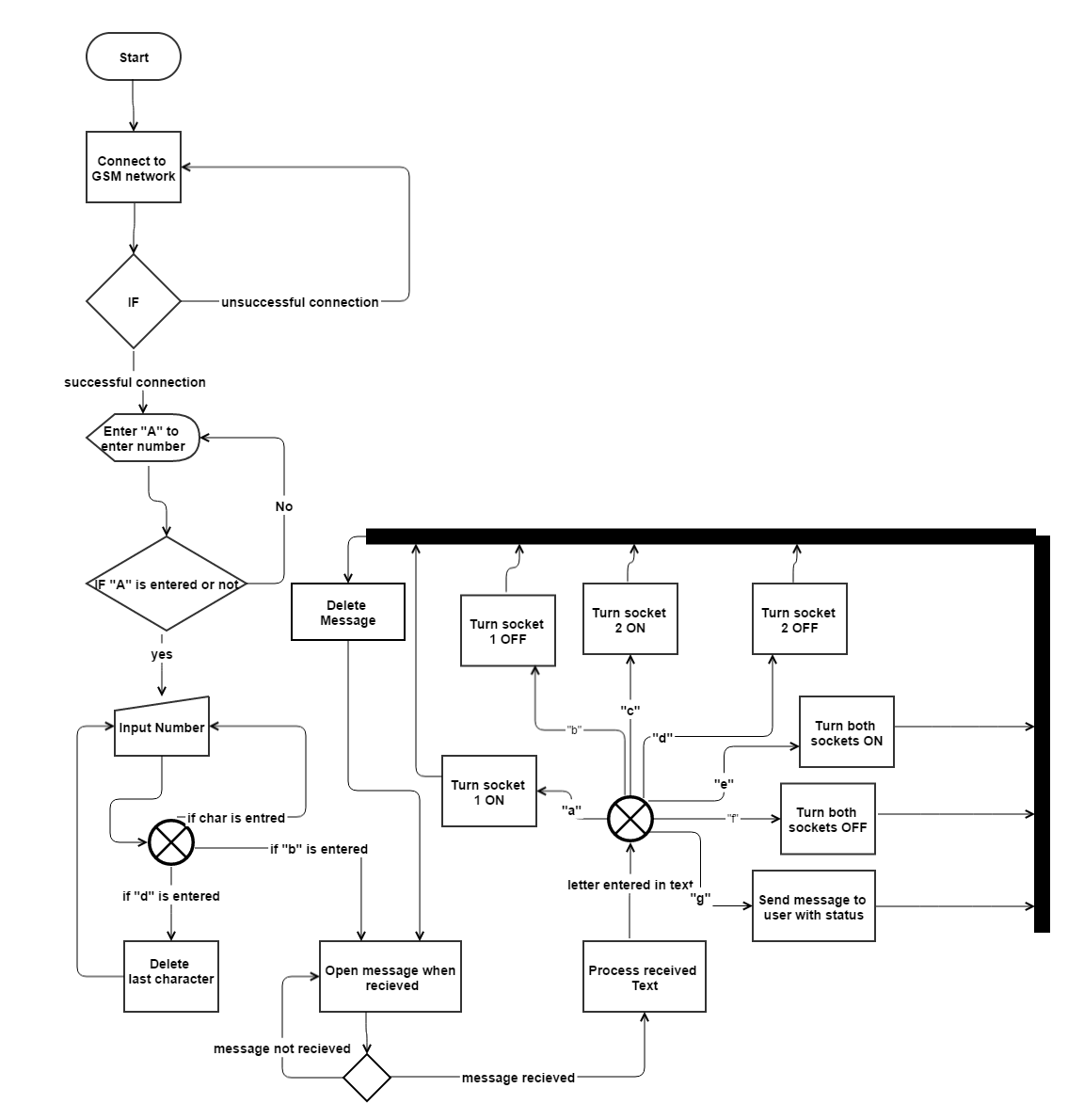


*Schematics 3.2 Power Converter*

3.3 Software Design

**Embedded software**

The software from figure No 3.1 was used in the Arduino controller to manage the feeds coming from the GSM circuit and control the relays. Commands are given to the relay circuit to switch the electricity ON/OFF as commanded by the user from the user interface.

****

*Figure 3.1 Embedded Software*

The C code used to programme the software design shown in list 3.1 includes a fragment of code that receives a text message from a mobile phone and issues commands to the Arduino pins or sends back a text message to the mobile holder with the states of the sockets. The code is provided in the below section [Arduino.cc, 2014]

List 3.1 Arduino Code snippet

while(c=sms.read())

{ if(senderNumber[3]==numpass[3] &&senderNumber[4]==numpass[4] && senderNumber[5]==numpass[5] && senderNumber[6]==numpass[6] && senderNumber[7]==numpass[7] && senderNumber[8]==numpass[8] ){match =1;}else{match=0;}; Serial.print(c);if (c=='a' && match == 1) {digitalWrite(A0, LOW); digitalWrite(A2, LOW); //Serial.println("socket a0 on");

};if (c=='b'&& match == 1) {digitalWrite(A0, HIGH); digitalWrite(A2, HIGH); // Serial.println("socket a0 on");

}; if (c=='c' && match == 1) {digitalWrite(A1, LOW); digitalWrite(A3, LOW); // Serial.println("socket a1 on");

}; if (c=='d'&& match == 1) {digitalWrite(A1, HIGH); digitalWrite(A3, HIGH); // Serial.println("socket a1 on");

}; if (c=='e' && match == 1) {digitalWrite(A0, LOW);digitalWrite(A1, LOW); digitalWrite(A2, LOW);digitalWrite(A3, LOW); // Serial.println("socket a0 a1 on");

}; if (c=='f' && match == 1) {digitalWrite(A0, HIGH);digitalWrite(A1, HIGH); digitalWrite(A2, HIGH);digitalWrite(A3, HIGH); // Serial.println("socket a0 a1 off");

}; if (digitalRead(A0)==HIGH){strcpy(sock1,"OFF .");}; if (digitalRead(A0)==LOW){strcpy(sock1,"ON .");};

if (digitalRead(A1)==HIGH){strcpy(sock2,"OFF .");};

if (digitalRead(A1)==LOW){strcpy(sock2,"ON .");};

if (c=='g' && match == 1) {

strcpy (str,"SOCKET ONE IS ")

strcat (str,sock1);

strcat (str," SOCKET TWO IS ");

strcat (str,sock2);

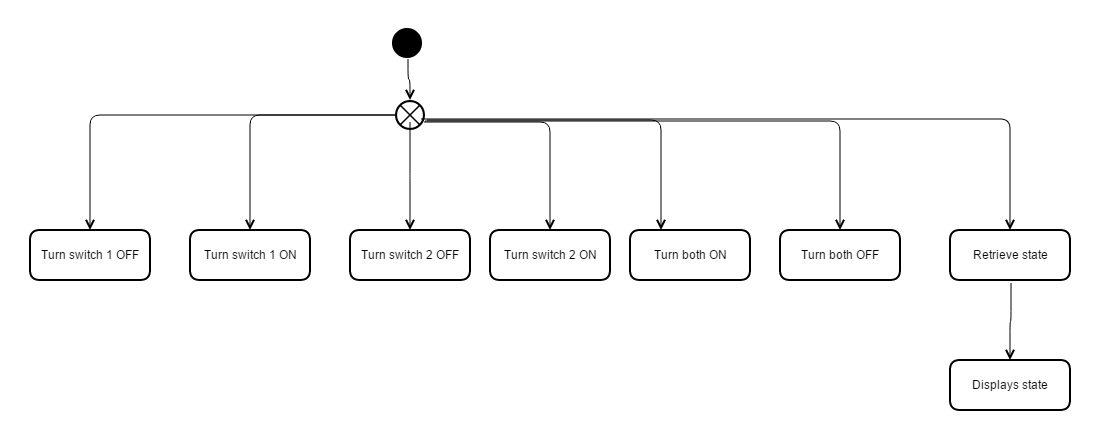
// lcd.println(remoteNum);

sms.beginSMS(numpass); sms.print(str);sms.endSMS();

};

**Graphical user interface software**

The mobile graphical user interface in figure No 3.2 is designed to control the smart socket device by sending commands and receiving data back for the smart socket. The application uses the mobile phone text messaging system to send and receive commands. It is to be noted that the smart socket can be controlled using only text messages from any phone.



*Figure 3.2 GUI Software Design*

The design in Figure 3.2 is coded using java and the android studio to produce a graphical user interface. Most of the code is generated by the android studio and libraries included in the studio were also used to send commands using the text messaging service the android phones provides.

To send and receive text messages in android studio, the user must allow permissions in the xml fie as seen below. [Mobiforge.com, 2014]

List 3.2 Code to Enable Permissions on an Android application

**<?xml** version="1.0" encoding="utf-8"**?>**

**<manifest** xmlns:android="http://schemas.android.com/apk/res/android"

package="net.learn2develop.SMSMessaging" android:versionCode="1" android:versionName="1.0.0"**>**

**<application** android:icon="@drawable/icon" android:label="@string/app\_name"**>**

**<activity** android:name=".SMS" android:label="@string/app\_name"**><intent-filter>**

**<action** android:name="android.intent.action.MAIN" **/>**

**<category** android:name="android.intent.category.LAUNCHER" **/>** **</intent-filter></activity>** **<receiver** android:name=".SmsReceiver"**>** **<intent-filter>** **<action** android:name= "android.provider.Telephony.SMS\_RECEIVED" **/>** **</intent-filter>** **</receiver>** **</application>** **<uses-permission** android:name="android.permission.SEND\_SMS"**></uses-permission><uses-permission** android:name="android.permission.RECEIVE\_SMS"**></uses-permission></manifest>**

The tags in list No 3.2 <uses-permission android:name="android.permission.SEND\_SMS"> </uses-permission>

<uses-permission android:name="android.permission.RECEIVE\_SMS"> </uses-permission>

Allows the user to send and receive text messages using the android operating system on a mobile phone.

**The functions below in list No 3.3 sends text to the smart socket. [Mobiforge.com, 2014]**

The function sendSMS include a parameter msg with the message to be sent then the string phoneNumber is assigned a phone number before smsManager.sendTEXTManager send the message.

List 3.3 Text Messaging Send Function

public void sendSMS(String msg){

String phoneNumber="07854170580";

String messege = msg;

SmsManager smsManager = SmsManager.getDefault();

smsManager.sendTextMessage(phoneNumber,null,messege,null,null);;}

**The function in list 3.4 below receives text messages from the device. [Mobiforge.com, 2014]**

The onRecieve function accepts a text message before looping through the buffer extracting the commands needed to control the device.

List .4 Text Messaging Receiving Function

public void onReceive(Context context, Intent intent)

{ //---get the SMS message passed in---

Bundle bundle = intent.getExtras();

SmsMessage[] msgs = null;

String str = "";

String num="";

if (bundle != null)

{

//---retrieve the SMS message received---

Object[] pdus = (Object[]) bundle.get("pdus");

msgs = new SmsMessage[pdus.length];

for (int i=0; i<msgs.length; i++){

msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);

// str += "SMS from " + msgs[i].getOriginatingAddress();

num = msgs[i].getOriginatingAddress();

// str += " :";

str += msgs[i].getMessageBody().toString();

str += "\n";

}

//---display the new SMS message--

Toast.makeText(context, str, Toast.LENGTH\_SHORT).show();

Toast.makeText(context, num, Toast.LENGTH\_SHORT).show();

//PUT IF STATMENT TO FILTER NUMBER HERE

if ( num.equals("+447854170580") ) {

Intent broadcastIntent = new Intent();

broadcastIntent.setAction("SMS\_RECEIVED\_ACTION");

broadcastIntent.putExtra("sms",str);

context.sendBroadcast(broadcastIntent);};

3.4 Final design

The final schematics in schematics No.3.3 shows the electronics in the device, the problem of power was solved from the preliminary design by providing a power converter and regulator circuit (figure No 3.2).

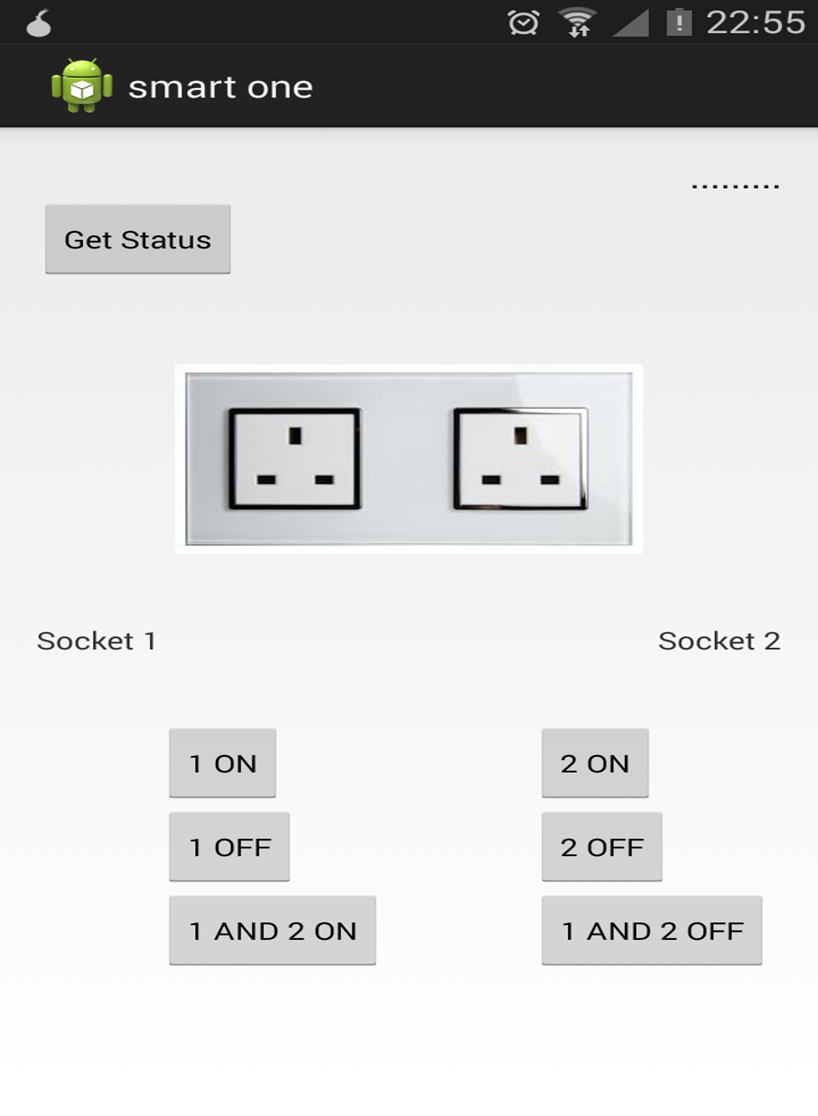


*Schematics 3.3 Final Design*

As noted in Schematics No 3.3 the power to the relays and LCD screen is provided from the power converter rather than the Arduino Uno board.

3.5 Graphical user interface

Figure No 3.3 below shows the graphical user interface as it reflected to the user in the android environment and a series of buttons are provided to control the smart socket device remotely.



*Figure 3.3 Graphical User Interface*

## 3.6 Test plan

### 3.6.1 Software testing

All aspects of the software must be tested to ensure robustness and continuation of processes without any failures arising from software malfunction. Three tests are planned for the software aspect of the device as explained below:

**Functional tests**

All the functions must be tested to meet the requirements of the device and input/outputs respectively. (Istqbexamcertification.com 2015)

**Speed test**

The speed of the device communication must be tested and documented to note any failures.

**Memory leak tests**

Due to the nature of the device’s use memory leaks must be prevented and the heap memory must be controlled. (Msdn.microsoft.com 2015)

### 3.6.2 Hardware testing

The hardware must be tested for the following:

* PAT
* Durability test
* analogue signature analysis
* analogue verification
* CE marking

3.7 3D Printed components

3d printed components are used to case some of the devices electronics and wiring. These components have been designed using a 3d design software called Sketchup and printed using a 3d printer, some of these components are used to hold the boards used inside the device and others are used to manage the cabling or batteries. A brief explanation is provided below for each component.

Image 3.1 is a general impression of the finished device.

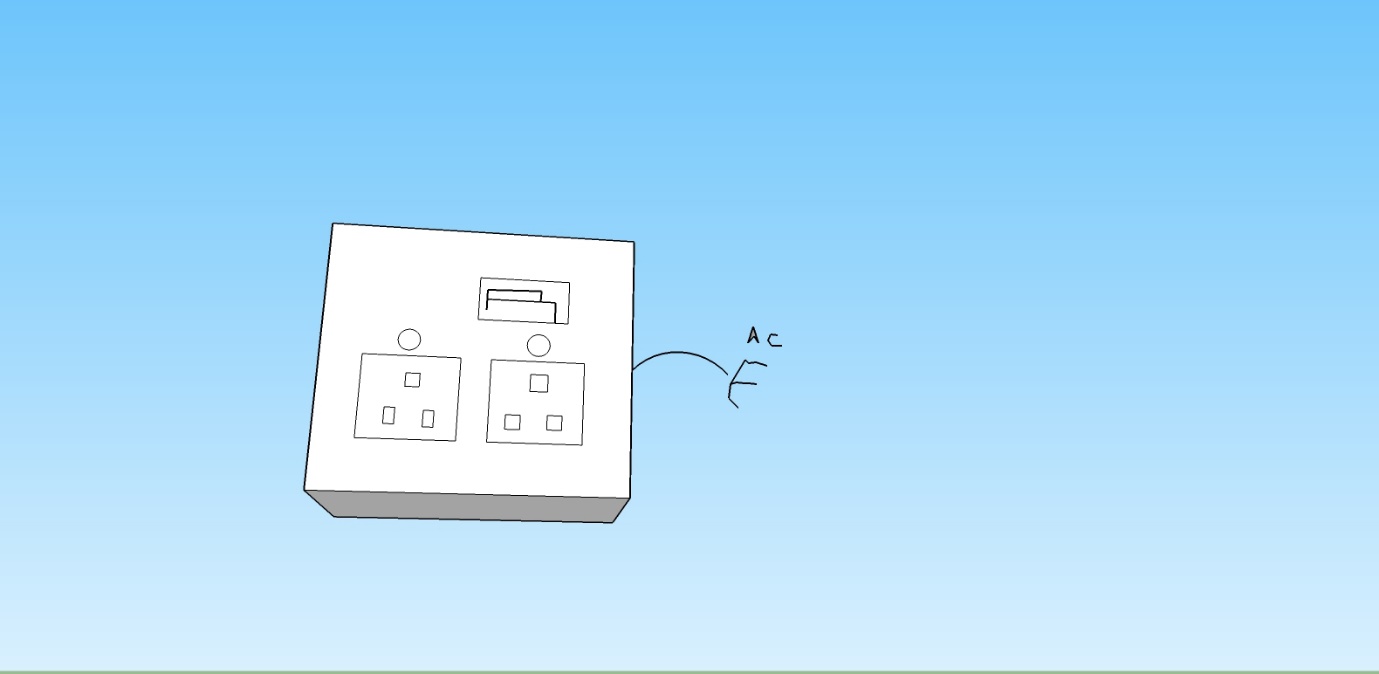
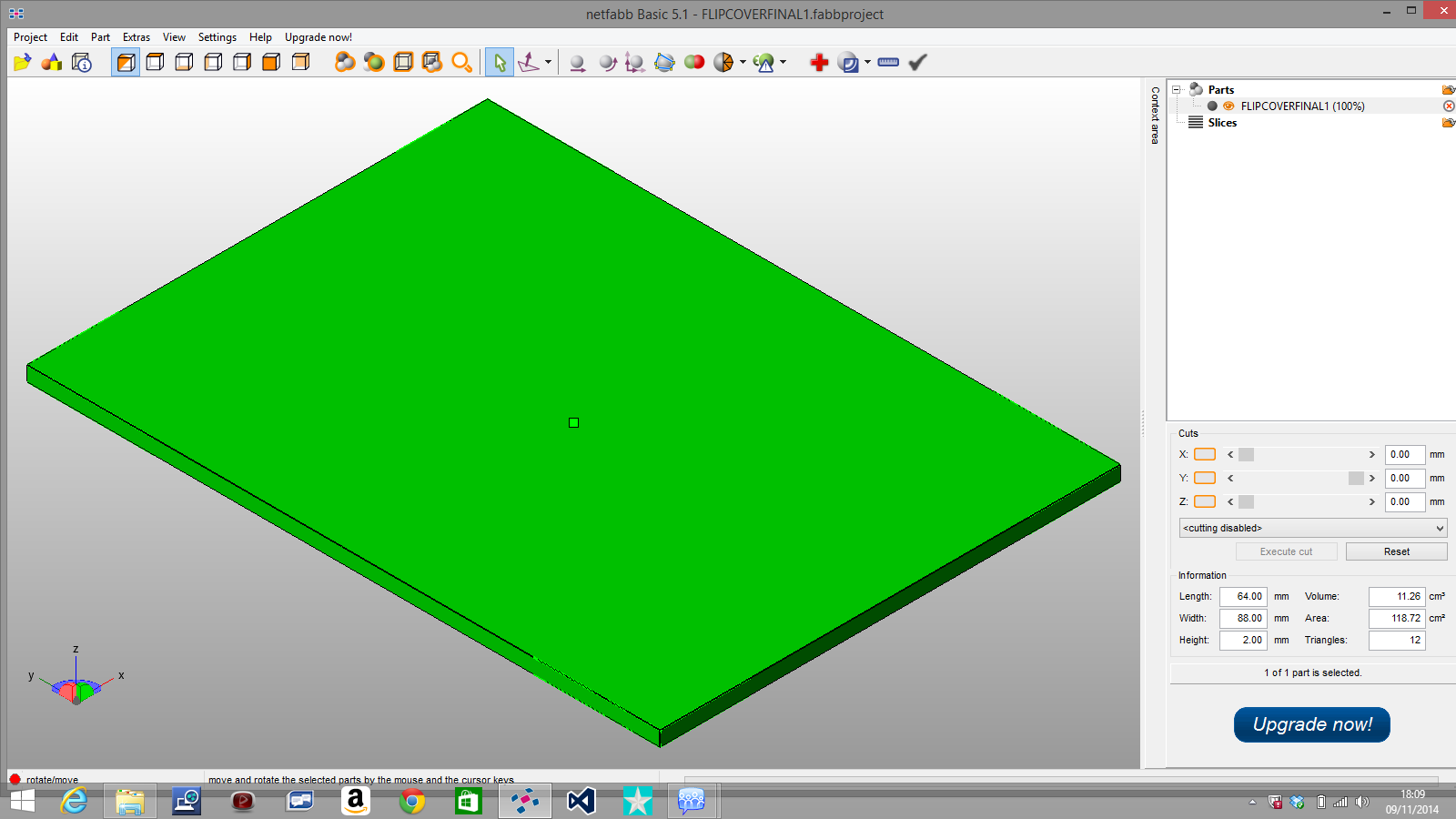


Image 3.1 Device Impression

* **Boards holders 3d components**

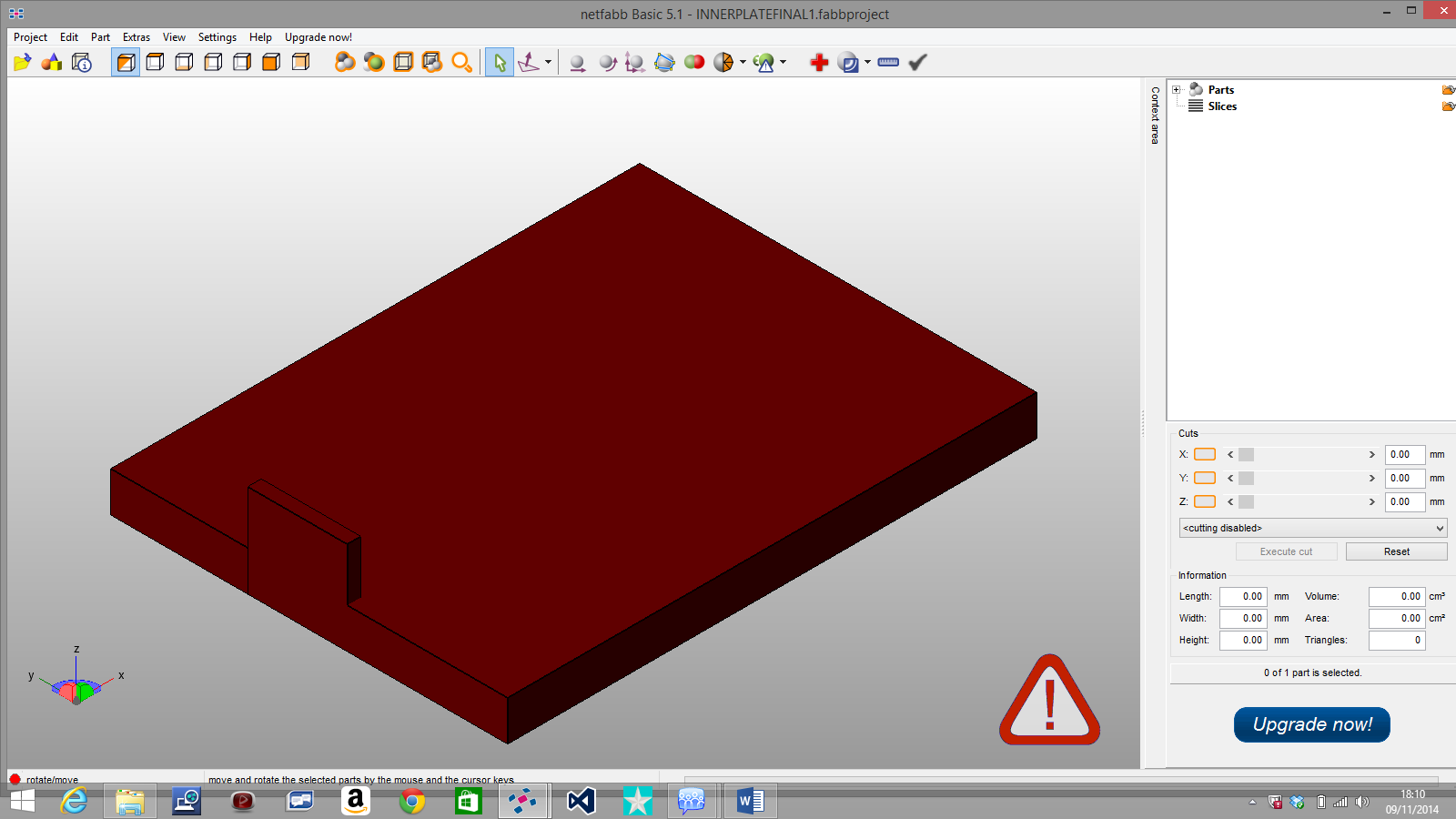
The figure No 3.4 shows the 3D printed plastic plate that is used to fit some of the Arduino shields on. It holds the electronic shield tight to the casing of the device and ensures that the integrity of the board is not compromised if the device is moved.



*Figure 3.4 Arduino Holder*

* **Board holder**

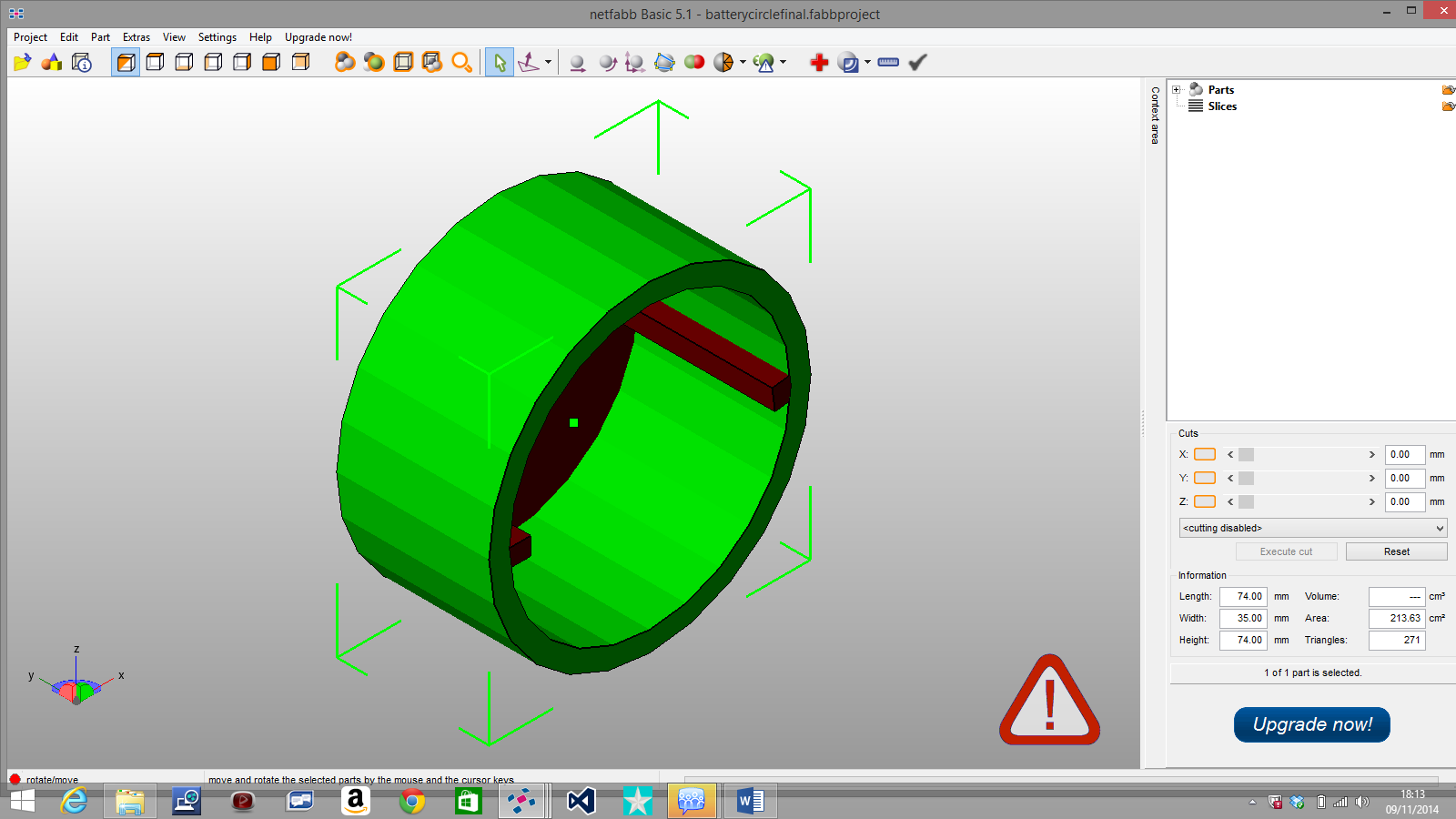
The board holder in figure 3.5 is a 3D plastic printed component that holds the GSM shield and the Arduino Uno to the side of the case allowing the GSM aerial to be exposed to the outer surface of the case.



*Figure 3.5 Relay and GSM Holder*

* **Battery holder 3d components**

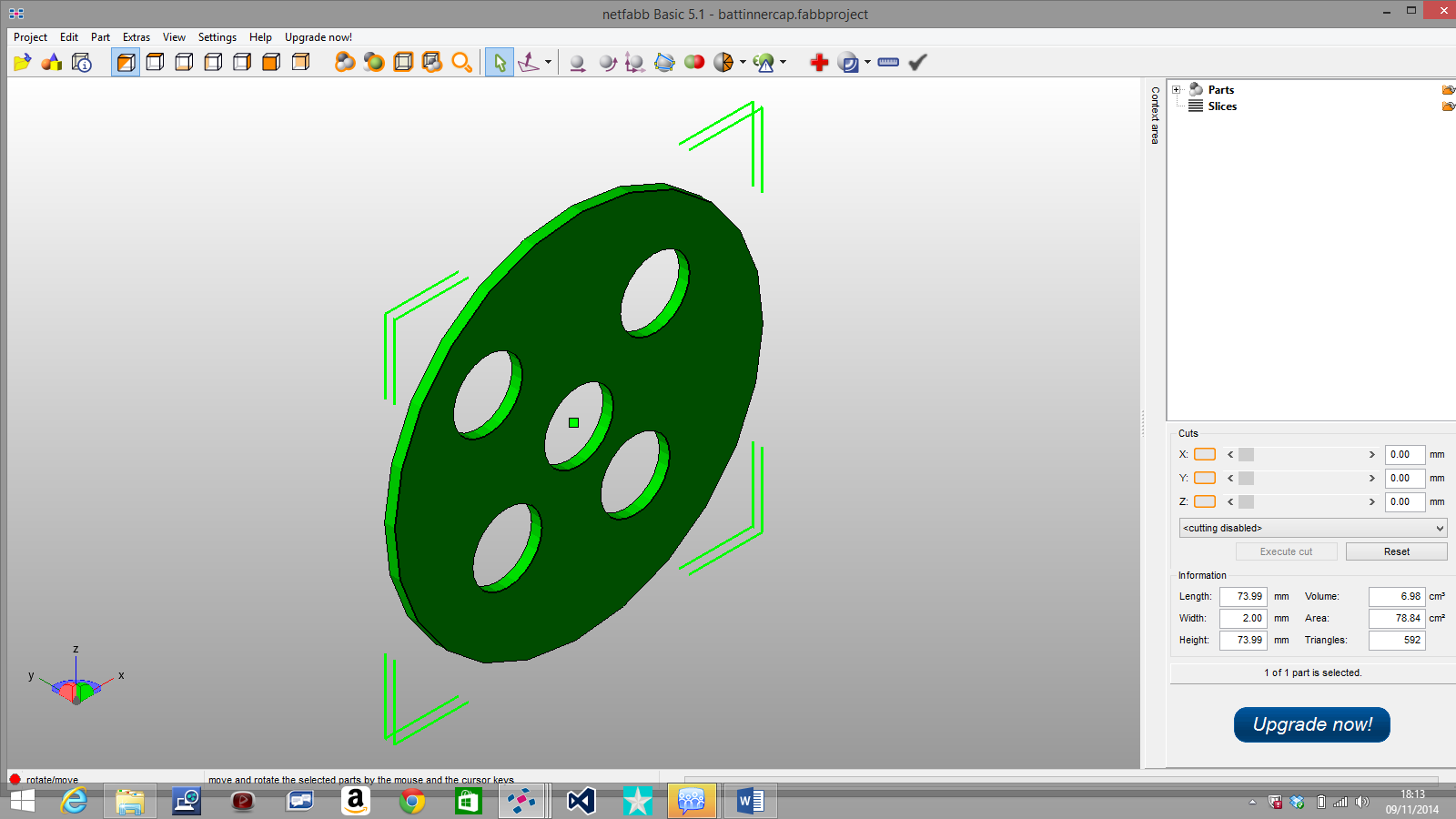
The figure 3.6 shows the 3D printed plastic component used in housing two 9v DC batteries. It fits inside the case and is used with the components shown in figure 3.7 and 3.8 to cover either side it.



*Figure 3.6 Battery Holder*

* **Battery holder cover**

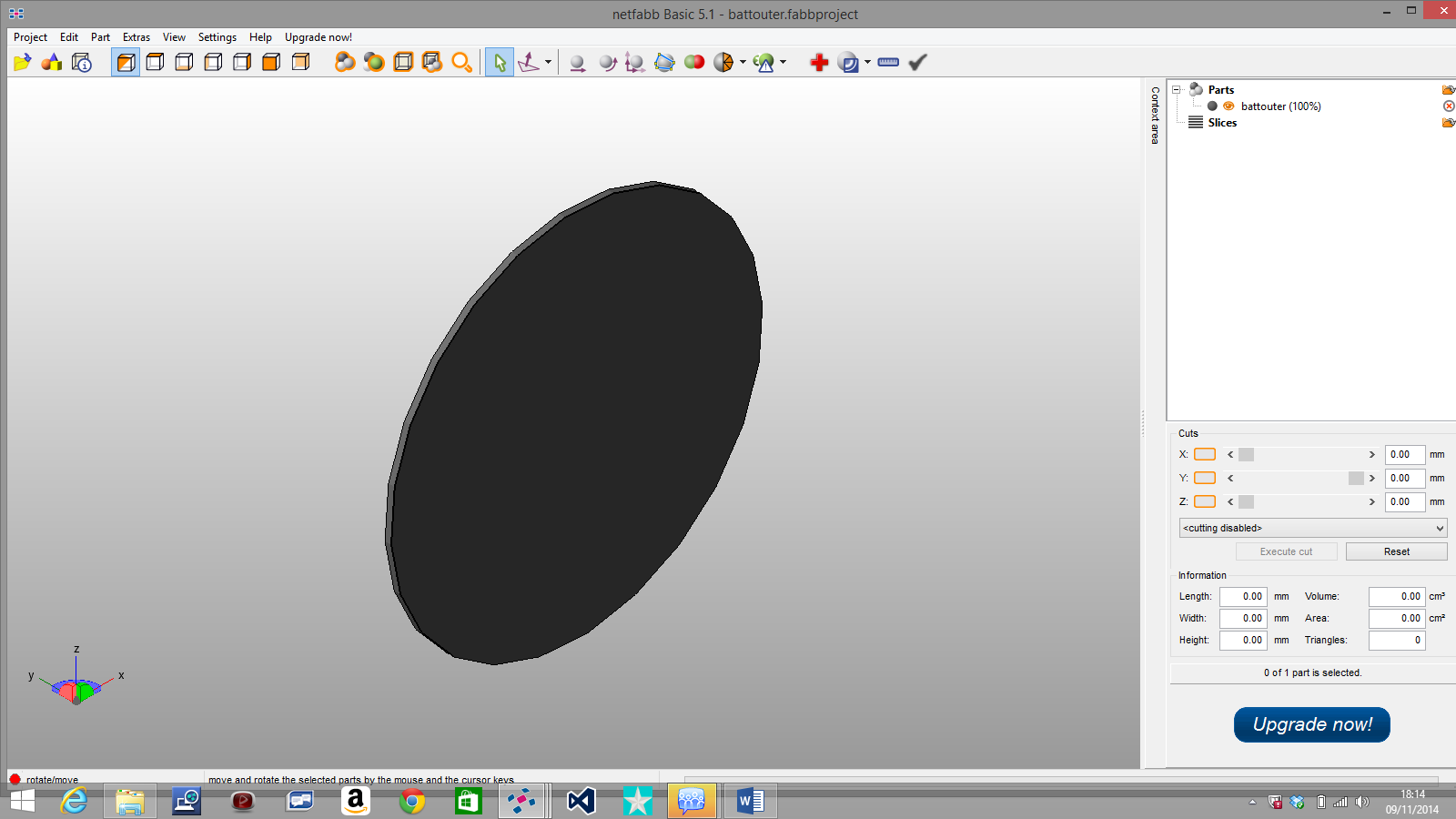
The battery holder cover below in figure 3.7 include small circular holes that allow easy access for any wires to be inserted into the device.



*Figure 3.7 Batteries inner cover*

* **Outer cover**

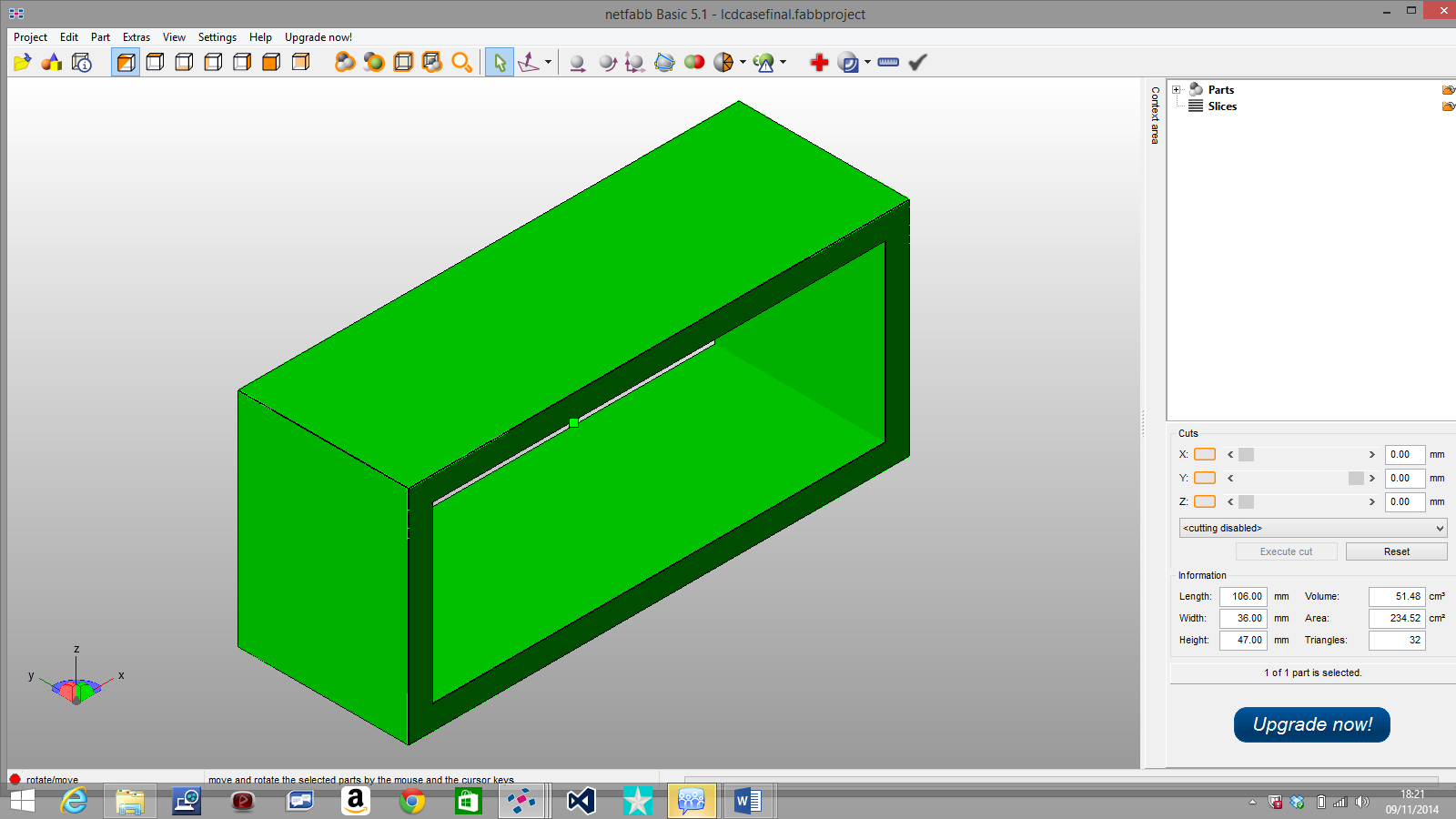
The outer cover in figure 3.8 is a 3D printed plastic component that covers the housing of the batteries.



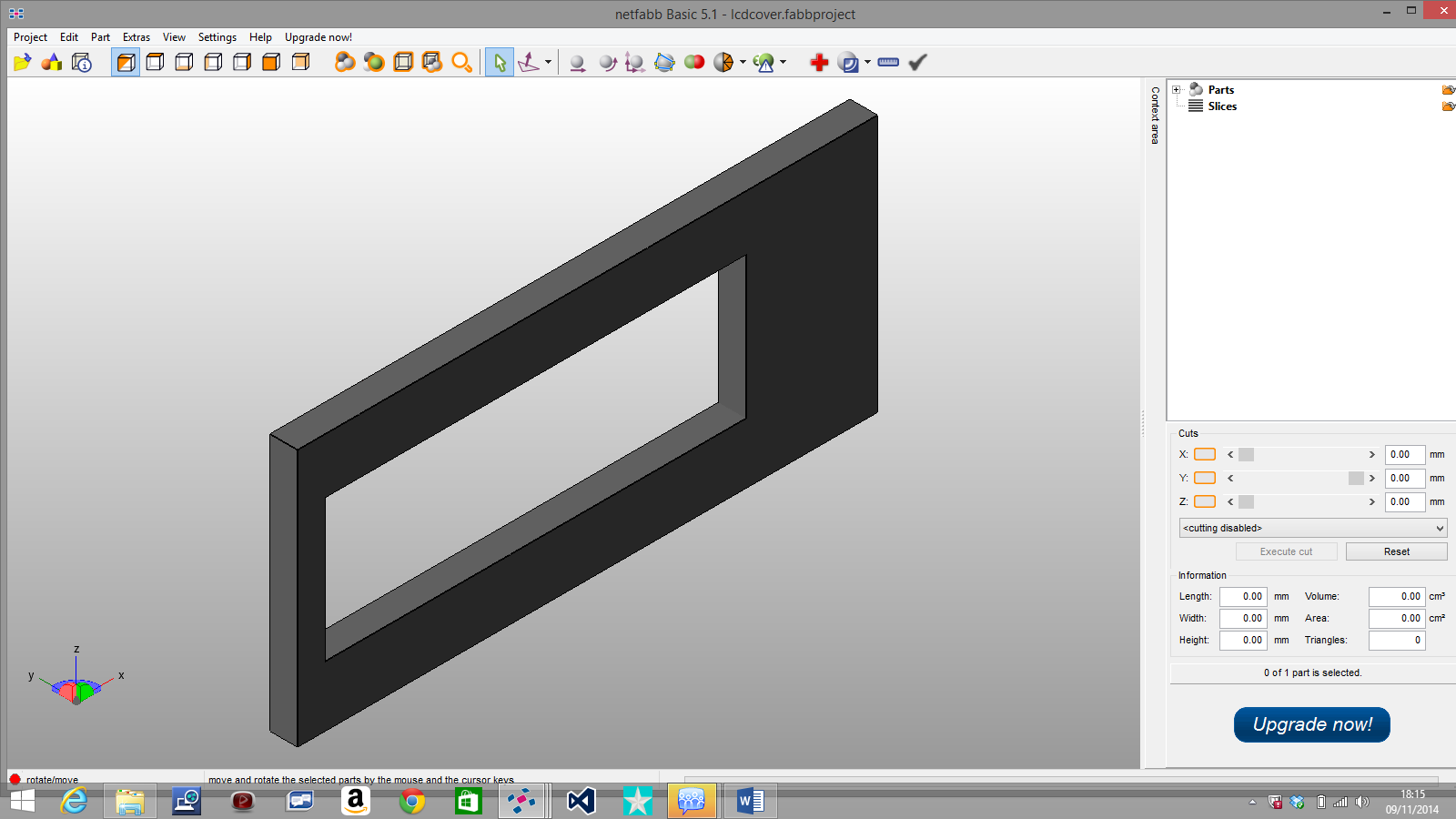
*Figure 3.8 batteries outer cover*

* **LCD 3d holder components**

The figure 3.9 below shows the plastic printed 3d component that houses the LCD screen for the device, the figure 3.10 below shows the cover for the LCD holder.



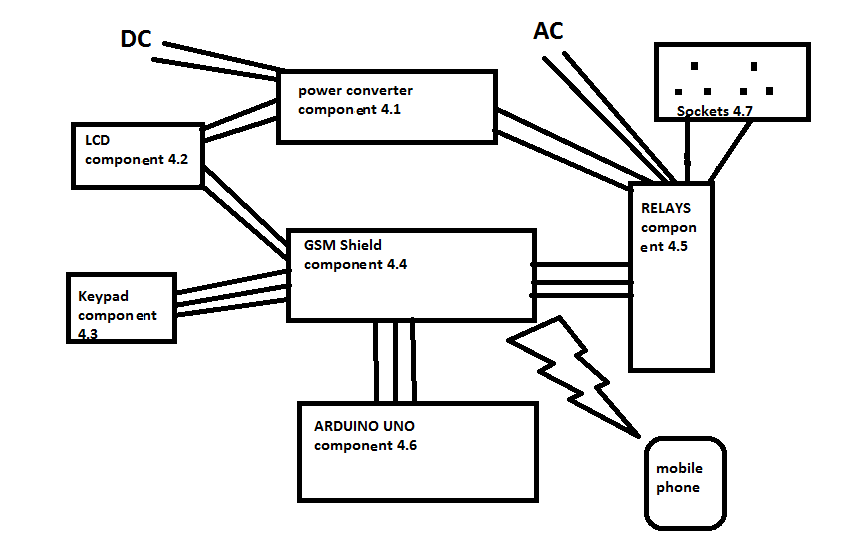
*Figure 3.9 LCD Holder*



*Figure 3.10 LCD Cover*

# 4. Implementation

The final prototype device was implemented using the schematics and components chosen in the design chapter. Below in the visualization 4.1 each component was given a number so it can be tracked in the explanations. The final schematics 3.3 in the design chapter show a more detailed view of the connections between the electronic components.



*Visualisation 4.1 showing components and general function*

## 4.1 Hardware implementation

### 4.1.1. Electronics

* **Power converter**

The power convertor [component 4.1] was built in accordance to the schematics 3.2, a PCB (printed circuit board) was used in conjunction with the following electronics to produce the power converter circuit

|  |  |  |
| --- | --- | --- |
| **Part** | **Usage** | **Notes** |
| LM7805 step down Regulator | To regulate the power |  |
| Pin rails | To provide positive and negative pin rails and an out voltage to led lights. |  |
| 2x 100 ohms resistors | To support led lights |  |
| 1 uf and 10 uf capacitors | To give continuous voltage |  |
| Heat sink | To keep the LM7805 cool. |  |

*Table 4.1 list of components in power converter*

All the components were soldered together and the final circuit tested using a multi meter to obtain output values.

* **Boards mounted**

All four boards are mounted on the 3D printed pieces shown in figures 3.4 and 3.5. Each of the 3D printed component was used to attach two boards, one at each side of it.

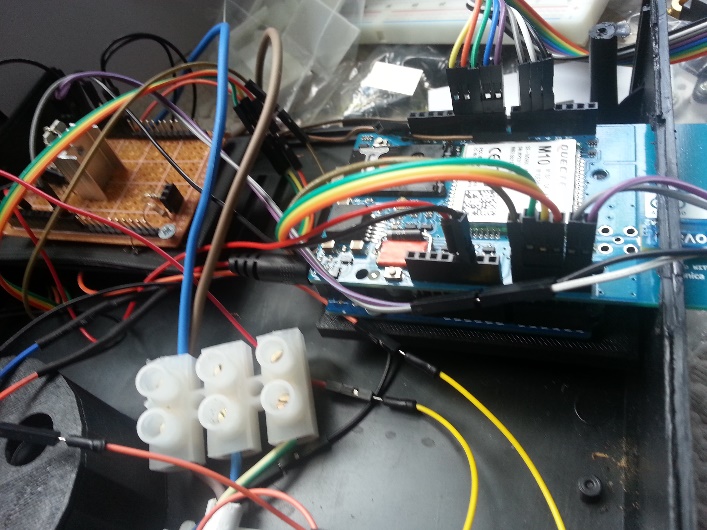
* **Connecting the circuits**

Each electronic board or circuit was connected in accordance to schematics 3.3 and visualization 4.1 using jumper cables of various sizes and connecters to other boards. The cables were rooted carefully around the case and tied together whenever possible. Further cables were connected to the power convertor board 4.1 to feed it with 9v dc power from a batteries compartment.

The GSM shield shown in [component 4.4] was mounted on the Arduino Uno [component 4.6], the relay’s board [component 4.5] was connected to the Arduino Uno’s pins through the GSM shield [component 4.4], the relay’s board [component 4.5] was then connected to the power converter directly to supply it with power, the relay’s board was also connected to the sockets [component 4.7] and AC power cable.

The LCD board [component 4.2] was connected to the power converter [component 4.1] and the Arduino Uno [component 4.6] through the GSM shield [component 4.4] pins, finally the keypad was connected to the Arduino Uno [component 4.6] through the GSM shield [component 4.4].

the image 4.1 below shows some of the circuit~~s~~ connections with the various cable jumpers running from one point to the other including the 240v 2.5mm cable connecting to the relays. The boards are clearly shown to be mounted on the 3D printed component as mentioned above.



*Image 4.1 circuit’s connections*

* **AC cable and sockets**

A 2.5 mm AC cable was used with a pair of AC plug sockets [component 4.7]. The sockets circuit was connected to the relays [component 4.5] to obtain the switching capability, and all connections were secured making sure no live connections are exposed by using shrinkable sleeves.

The electrical sockets were secured to the case as seen below in image 4.2.



*Image 4.2 electrical sockets*

### **4.1.2 Case**

After considering to make a complete case for the device using a 3D printed components, an alternative readymade case was found on the market (Ebay.co.uk, 2015-2) that cost £10 and that is significantly less than printing all the components necessary to make the case which would be around £200.

* **Modification**

The case needed modification to accompany all the electronic boards and circuitry. It also needed the battery compartment to hold the double 9v batteries needed by the circuitry.

Various openings were drilled using a drill saw and all the 3D printed components were fitted to the case using small screws as seen in image 4.1 and 4.2.

* **Using 3D printed components**

The batteries compartment was made by drilling a large diameter hole at the back of the case and fitting the 3D components in figures 3.6, 3.7 and 3.8.

The LCD screen compartment as shown in image 4.2 was fitted to the case using the 3D components in figures 3.9 and 3.10 by screwing them on to the case and running the LCD cables into the case from a pre-existing hole.

## **4.2 Software implementation**

The software was implemented in two parts, first the embedded part then the graphical user using an android application. Both software’s are coordinated and communicate with each other using the gsm mobile network.

* **Embedded software**

The embedded software was written using the C programming language and the Arduino development environment. The design in figure 3.1 was used as a blueprint to produce the programme and the code snippet used in list 3.1 was also utilised to read commands coming in from any mobile devices to the smart socket device, the commands would make the smart socket do various tasks in accordance with preprogramed instructions.

The final programme was uploaded to the Arduino Uno using a USB cable that extended from a laptop to the Arduino board.

* **Graphical user interface software**

An android graphical user interface was developed using android studio and java programming language. The application development process was as follows:

* Download and install android studio IDE.
* Connect the smartphone to the android IDE using a drop down list.
* Drag and drop text boxes and buttons to the android visual view.
* Enable all permissions to receive and send text messages in the android application programme as shown in list 3.2.
* Input the code to receive and send text messages in the application as shown in lists 3.3 and 3.4.
* Programme the application to send commands to the smart socket device using the send and receive functions as shown above.

**Summary of the smart socket API**

The smart socket embedded programme was programmed with a loop that recognises all incoming text messages and initially uses the incoming number to check if it corresponds with the phone number stored in memory and then processes the text message for any characters. Each character will instigate a command to the hardware as shown in the user manual table, the device could be controlled using normal text messages as well as the android application supplied with it.

## 4.3 User instructions:

This is a step by step instruction on how to use the smart socket:

1. Make sure you install two 9v batteries in the smart sockets battery compartment at the bottom of the device by unscrewing the small round battery compartment cap.
2. Plug the smart socket device into your home 240v electricity wall socket.
3. Switch both of the smart socket switches on and wait for the LCD display to load the programme (you will see words “Enter Phone number” appearing when it’s loaded correctly).
4. Press the button (A) on the keypad until it prompt you to enter a new phone number.
5. If you are in the UK start to enter your mobile phone number starting with +44 i.e.( +447791320992) , if you make any mistakes press (D) to delete the character you entered, when you entered the right number just save it by clicking (B).
6. You’re ready to send messages to the smart socket now, the smart sockets number is 077854170580.

|  |  |  |
| --- | --- | --- |
| **Letter** | **Action** | **notes** |
| a | Turn socket 1 on | If the batteries are not fully charged the device won’t work properly. |
| b | Turn socket 1 off |  |
| c | Turn socket 2 on |  |
| d | Turn socket 2 off |  |
| e | Turn socket 1 & 2 on |  |
| f | Turn socket 1 & 2 off |  |
| g | Sends a text message to you with the sockets states |  |

1. To change the Sim card of the smart socket you must unplug the smart socket and switch it off then open the box by unscrewing all four screws and find the Sim on the GSM shield and replacing it. Please note that some Sim cards might not work with the device.

## 4.4 Financial cost of producing the device.

Costing below is for hardware only.

|  |  |  |
| --- | --- | --- |
| **Part** | **Source** | **Cost** |
| Arduino Uno | EBAY | £6 |
| Arduino GSM shield | EBAY | £120 |
| Arduino Relay shield | EBAY | £9 |
| PCB, regulator, resisters, jumpers, switches | EBAY | £7 |
| Case | EBAY | £10 |
| Sockets, cable, connector | Screwfix | £3 |
| LCD, Keypad | EBAY | £8 |
| **Total** |  | **£163** |

# 5. Validation

## 5.1 Testing

### 5.1.1 Hardware

The hardware could be tested in many ways i.e. (PAT, Durability test, analogue signature analysis, analogue verification). All the hardware tests mentioned above is beyond the scope of this academic endeavour and would require funds that is not available to the project.

### 5.1.2 Software

**Black box testing**

* **Speed test**

The speed test was undertaken using the method of sending multiple commands seconds apart to see how the device would cope and react, below is a table 5.1 showing the results.

|  |  |  |  |
| --- | --- | --- | --- |
| **Action** | **Expected outcome** | **Actual outcome** | **Notes** |
| 1 ON then after 3 seconds 1 OFF | Socket 1 comes ON then OFF | Socket 1 comes ON then OFF | PASS |
| 1 ON then after 7 seconds 1 OFF | Socket 1 comes ON then OFF | Socket 1 comes ON then OFF | PASS |
| 1 ON then after 10 seconds 1 OFF | Socket 1 comes ON then OFF | Socket 1 comes ON then OFF | PASS |
| 1 ON then after 15 seconds 1 OFF | Socket 1 comes ON then OFF | Socket 1 comes ON then OFF | PASS |

*Table 25.1 Speed test*

NOTE: the instruction sent to the smart socket device is executed according to FIRST COME FIRST EXECUTED WAY. So, whichever instruction of text message is received first is the first executed, thus the GSM network is responsible for the order the messages are sent to the device.

* **Functional tests**

The functional test was undertaken using the mobile application shown in figure 3.3, all the buttons in the table 5.2 below correspond to the buttons in the android application mentioned above. (Istqbexamcertification.com 2015)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Button** | **Action** | **Expected outcome** | **Actual outcome** | **Notes** |
| 1 ON | Click to turn socket 1 ON | Socket 1 turns ON | Socket 1 turns ON | PASS |
| 1 OFF | Click to turn socket 1 OFF | Socket 1 turns OFF | Socket 1 turns OFF | PASS |
| 1 AND 2 ON | Click to turn socket 1 and 2 ON | Socket 1 and 2 turn ON | Socket 1 and 2 turn ON | PASS |
| 2 ON | Click to turn socket 2 ON | Socket 1 and 2 turn ON | Socket 1 and 2 turn ON | PASS |
| 2 OFF | Click to turn socket 2 OFF | Socket 2 turns OFF | Socket 2 turns OFF | PASS |
| 1 AND 2 OFF | Click to turn socket 1 and 2 OFF | Socket 1 and 2 turn OFF | Socket 1 and 2 turn OFF | PASS |
| Get Status | Click to get current status of sockets | Socket 1 and 2 are OFF | Socket 1 and 2 are OFF | PASS  The status might return different values depending on the state of the sockets. |

*Table 5.2 functional test*

**White box testing**

* **Memory leak test**

Both the embedded and the mobile software created to control this device were coded without the use of heap allocation commands thus a memory leak is not possible in this instance unless further modifications are made to the code that would require further memory tests. (Msdn.microsoft.com 2015)

## 5.2 Evaluation

The device was evaluated by a members of the public and questions were presented after the evaluation and the following answers were obtained below in table 5.3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **What did you think of the device?** | **Was it easy to use?** | **What changes would you recommend for the device?** | **Would you buy it?** | **How much would you pay for it?** |
| Tester 1 | I thought it was interesting | Relatively easy | Make it simpler to use | YES | £50 |
| Tester 2 | Very useful | Easy enough to understand if shown | Make it smaller and simpler to use | Yes if I needed the functions. | £30 |
| Tester 3 | Never seen anything like before | I need to read the instructions | Make it simpler to use, maybe only a web app without the rest. | YES | £50 |
| Tester 4 | I liked it | Not really | Make it simpler to use | probably | £35 |

*Table 5.3 Device evaluation*

# 6.0 Discussion

Below section discuses about the device focusing on the following objectives and also comments on it achievability.

* **Security**
* **Compatibility**
* **Use**
* **Communication**
* **Safety**
* **Legal requirements**
* **Functions**

## Security

As stated in the objective section

*“The device must be secured and measures should be taken to ensure that no access to a third party is allowed other than the intended user operating the device remotely.”*

The security objective has been met due to the use of phone number validation in the Arduino code (List 3.1).

## Compatibility

As stated in the objective section

*“The device must be compatible to electrical current used both in Europe, Asia and the USA so that it can be used almost in any country of the world.”*

This objective has been reached and the device can operate with 240v and 110v respectively however a plug convertor must be used.

## 6.3 Use

As stated in the objective section

“The device must be controlled by a mobile phone. Text messaging is the preferred communication mechanism.”

This objective has been accomplished and the device is controlled from an android application that sends a text message to command the device as shown in figure 3.3 with all the functions implemented.

## 6. 4 Communication

As stated in the objective section

“The device should communicate with the user. It must relay information of the status of the switch whenever the user request it.”

This objective has been achieved using a button on and android application that retrieves the device’s status as shown in figure 3.3.

## 6.5 Safety

As stated in the objective section

“The device must also be safe to use by the public.”

This objective has not been achieved: the hardware could be tested in many ways i.e. (PAT, Durability test, analogue signature analysis, analogue verification) , all the hardware tests mentioned above is beyond the scope of this academic endeavour and would require funds that are not available to the project. In-text: (Hse.gov.uk 2015)

## 6.6 legal requirements

Both the PAT Test and the CE marking could not be undertaken due to the cost. Thus, the objective has not been achieved. (Ce-test 2015)

## 6.7 functions

All the functions have been implemented and are working as required.

# 7. Conclusion

The conclusion is broken down to two part as shown below.

## 7.1 achievement of project aim

The aim of this project was to research, design, implement and assess a device that control an electric socket remotely. This aim has been achieved to a good standard and the fully functional prototype has been successfully produced.

## 7.2 future work

After developing, testing and evaluating the device it was apparent that it could have been planned better with less functionalities that were not needed by the end user. The use of the LCD display together with the keypad is not needed and could be replaced with a device that can be operated entirely from a mobile application.

The mobile user interface can be designed better including lighting the socket on the GUI when a socket is ON or OFF. Furthermore, in terms of connectivity, the device could do with the addition of internet connection so it can be used without text messages.

# References

Arduino.cc, (2014). *Arduino - ArduinoBoardUno*. [Online] Available at: http://arduino.cc/en/Main/arduinoBoardUno [Accessed 5 Nov. 2014].

Arduino.cc, (2014). Arduino - ArduinoGSMShield. [Online] Available at: http://arduino.cc/en/Main/ArduinoGSMShield [Accessed 5 Nov. 2014].

Arduino-info.wikispaces.com, (2014). arduino-info - RelayIsolation. [Online] Available at: http://arduino-info.wikispaces.com/RelayIsolation [Accessed 5 Nov. 2014].

Allan, A. (2013). Arduino Uno vs BeagleBone vs Raspberry Pi. [Online] MAKE. Available at: http://makezine.com/2013/04/15/arduino-uno-vs-beaglebone-vs-raspberry-pi/ [Accessed 5 Nov. 2014].

AVR, H. (2014). HD44780 1602 LCD Display Kit Blue Yellow/Green 16x2 Arduino Raspberry Pi AVR | eBay. [online] Ebay.co.uk. Available at: http://www.ebay.co.uk/itm/like/271383948315?limghlpsr=true&hlpv=2&ops=true&viphx=1&hlpht=true&lpid=108&chn=ps&device=c&adtype=pla&crdt=0&ff3=1&ff11=ICEP3.0.0-L&ff12=67&ff13=80&ff14=108&ff19=0 [Accessed 13 Nov. 2014].

Bamafolks.com, (2014). Programming Languages for Embedded Systems. [online] Available at: http://www.bamafolks.com/randy/students/embedded/programming\_languages.html [Accessed 8 Nov. 2014].

Banggood.com, (2014). Cellphone Phone GSM Remote Control 3 Sockets Smart Socket Power Switch - US$45.99. [online] Available at: http://www.banggood.com/Cellphone-Phone-GSM-Remote-Control-3-Sockets-Smart-Socket-Power-Switch-p-926484.html?currency=GBP&utm\_source=google&utm\_medium=shopping&utm\_content=miko\_ruby&utm\_campaign=Industrial-uk&gclid=CjwKEAjwre6dBRC94d-Gma7g3wcSJACNatZeI7arYWHbs2WBbPL3TJl6FDHwN3f93XRhzMrprMgl3xoCf1Tw\_wcB [Accessed 8 Nov. 2014].

Banggood.com, (2014). GSM Cell Phone Remote Control Switch Electric Wireless Smart Socket - US$35.08. [online] Available at: http://www.banggood.com/GSM-Cell-Phone-Remote-Control-Switch-Electric-Wireless-Smart-Socket-p-924989.html?currency=GBP&utm\_source=google&utm\_medium=shopping&utm\_content=miko\_ruby&utm\_campaign=Industrial-uk&gclid=CjwKEAjwre6dBRC94d-Gma7g3wcSJACNatZeC9KThokttZklz-Kr-Qt1o3lruo8IG6tEviR8dQWBDxoCldvw\_wcB [Accessed 8 Nov. 2014].

ce-test, b. 2015. CE marking for Electronic equipment [Online]. Available at: http://cetest.nl/cemark0.htm [Accessed: 14 March 2015].

Cogswell, J. (2014). Google's Android Studio vs. Eclipse: Which Fits Your Needs? - Dice News. [Online] Dice News. Available at: http://news.dice.com/2014/03/19/googles-android-studio-vs-eclipse-fits-needs/ [Accessed 13 Nov. 2014].

Cooking-hacks.com, (2014). Cooking Hacks - Documentation - GPRS/GSM Quadband Module for Arduino Tutorial (SIM900). [Online] Available at: http://www.cooking-hacks.com/documentation/tutorials/arduino-gprs-gsm-quadband-sim900 [Accessed 7 Nov. 2014].

eBay, (2014). DC ADJUSTABLE VOLTAGE REGULATOR CONVERTER IN: 40V TO 3V OUT: 36V TO 1.2V 1.5A. [Online] Available at: http://www.ebay.co.uk/itm/DC-ADJUSTABLE-VOLTAGE-REGULATOR-CONVERTER-IN-40V-TO-3V-OUT-36V-TO-1-2V-1-5A-/271296897498?pt=UK\_BOI\_Electrical\_Test\_Measurement\_Equipment\_ET&hash=item3f2a8e21da [Accessed 7 Nov. 2014].

Ebay.co.uk, (2015-2). Enclosure box | eBay. [Online] Available at: http://www.ebay.co.uk/sch/i.html?\_from=R40&\_trksid=p2050601.m570.l1311.R3.TR11.TRC1.A0.H1.Xenclos&\_nkw=enclosure+box&\_sacat=0 [Accessed 22 Jan. 2015].

Elinux.org, (2014). RPi Expansion Boards - eLinux.org. [Online] Available at: http://elinux.org/RPi\_Expansion\_Boards [Accessed 6 Nov. 2014].

Editor, U. (2014). Using the Layout Editor | Android Developers. [online] Developer.android.com. Available at: https://developer.android.com/sdk/installing/studio-layout.html [Accessed 13 Nov. 2014].

Fairchildsemi.com, (2014). LM7805 PDF Datasheet 3-Terminal 1A Positive Voltage Regulator. [Online] Available at: https://www.fairchildsemi.com/products/power-management/voltage-regulators/positive-voltage-linear-regulators/LM7805.html [Accessed 9 Nov. 2014].

Fundamentals, A. (2014). Application Fundamentals | Android Developers. [Online] Developer.android.com. Available at: http://developer.android.com/guide/components/fundamentals.html [Accessed 8 Nov. 2014].

Google.co.uk, (2014). *arduino - Google Search*. [Online] Available at: https://www.google.co.uk/search?output=search&tbm=shop&q=arduino&oq=arduino&gs\_l=products-cc.3..0l10.2142.3499.0.3709.7.4.0.3.3.0.75.274.4.4.0....0...1ac.1.58.products-cc..0.7.336.46YsMguXwoU&gws\_rd=ssl#tbm=shop&q=arduino+uno [Accessed 5 Nov. 2014].

Google.co.uk, (2014). Google. [Online] Available at: https://www.google.co.uk/#q=arduino+relay+shield&tbm=shop&spd=5844175747183059684 [Accessed 7 Nov. 2014].

GmbH, n. (2014). netfabb Software - Software for 3D Printing - 3D Software for STL files - fixing, repair, editing, merge STL data for Rapid Manufacturing - STL Viewers and STL repair. [online] Netfabb.com. Available at: http://www.netfabb.com/ [Accessed 13 Nov. 2014].

Guindon, C. (2014). [Online] Eclipse.org. Available at: https://www.eclipse.org/ [Accessed 13 Nov. 2014].

Hardkernel.com, (2014). ODROID | Hardkernel. [Online] Available at: http://www.hardkernel.com/main/main.php [Accessed 5 Nov. 2014].

Hse.gov.uk, 2015. Electrical inspection and testing - Electrical safety at work [Online]. Available at: http://www.hse.gov.uk/electricity/information/testing.htm [Accessed: 14 March 2015].

Isocket.eu, (2014). iSocket: GSM Power Socket only 99e, Made in Finland, isocet, isoket Best Device for Remote GSM Control. [online] Available at: http://www.isocket.eu/devices [Accessed 8 Nov. 2014].

Istqbexamcertification.com, 2015. What is Functional testing (Testing of functions) in software? [Online]. Available at: http://istqbexamcertification.com/what-is-functional-testing-testing-of-functions-in-software/ [Accessed: 14 March 2015].

Linear.com, (2014). LTC3108 - Ultralow Voltage Step-Up Converter and Power Manager - Linear Technology. [Online] Available at: http://www.linear.com/product/LTC3108 [Accessed 5 Nov. 2014].

MakeUseOf, (2013). 7 Operating Systems You Can Run With Raspberry Pi. [Online] Available at: http://www.makeuseof.com/tag/7-operating-systems-you-can-run-with-raspberry-pi/ [Accessed 6 Nov. 2014].

Msdn.microsoft.com, 2015. How to Find A Memory Leak [Online]. Available at: https://msdn.microsoft.com/en-us/library/ms859415.aspx [Accessed: 15 March 2015].

Mobiforge.com, (2014). SMS Messaging in Android | mobiForge. [Online] Available at: http://mobiforge.com/design-development/sms-messaging-android [Accessed 9 Nov. 2014].

Oracle.com, (2014). Java Software | Oracle. [Online] Available at: https://www.oracle.com/java/index.html [Accessed 5 Nov. 2014].

Python.org, (2014). Welcome to Python.org. [Online] Available at: https://www.python.org/ [Accessed 5 Nov. 2014].

Qsl.net, (2014). Circuits Controlling Relays page. [Online] Available at: http://www.qsl.net/yo5ofh/hobby%20circuits/circuits\_controlling\_relays.htm [Accessed 5 Nov. 2014].

Raspberrypi.org, (2014). Raspberry Pi. [Online] Available at: http://www.raspberrypi.org/ [Accessed 5 Nov. 2014].

Rs-online.com, (2012). Arduino or Raspberry Pi? » DesignSpark. [online] Available at: http://www.rs-online.com/designspark/electronics/blog/arduino-or-raspberry-pi [Accessed 5 Nov. 2014].

Sketchup.com, (2014). Home | SketchUp. [Online] Available at: http://www.sketchup.com/ [Accessed 13 Nov. 2014].

Stack, W. (2014). What programming languages does the Raspberry Pi support? [Online] Raspberrypi.stackexchange.com. Available at: http://raspberrypi.stackexchange.com/questions/15480/what-programming-languages-does-the-raspberry-pi-support [Accessed 6 Nov. 2014].

Seeedstudio.com, (2014). GPRS Shield V1.0 - Wiki. [Online] Available at: http://www.seeedstudio.com/wiki/GPRS\_Shield\_V1.0 [Accessed 5 Nov. 2014].

Slashgear.com, (2014). ODROID-X trumps Raspberry Pi with quad-core CPU - SlashGear. [Online] Available at: http://www.slashgear.com/odroid-x-trumps-raspberry-pi-with-quad-core-cpu-13238487/ [Accessed 7 Nov. 2014].

Taral, P. (2012). What is Low Level Language? | C Programming: Fundamental. [Online] Learn Programming Language Step By Step. Available at: http://www.c4learn.com/c-programming/low-level-language/ [Accessed 8 Nov. 2014].

Techwatch.keeward.com, (2014). Arduino vs. Raspberry Pi vs. CubieBoard vs. Gooseberry vs. APC Rock vs. OLinuXino vs. Hackberry A10 | Keeward’s TechWatch blog. [Online] Available at: http://techwatch.keeward.com/geeks-and-nerds/arduino-vs-raspberry-pi-vs-cubieboard-vs-gooseberry-vs-apc-rock-vs-olinuxino-vs-hackberry-a10/ [Accessed 5 Nov. 2014].

UK, D. (2014). Serial IIC/I2C/TWI 2004 204 20X4 Character LCD Module Display For Arduino UK. [Online] eBay. Available at: http://www.ebay.co.uk/itm/Serial-IIC-I2C-TWI-2004-204-20X4-Character-LCD-Module-Display-For-Arduino-UK/141415310957?\_trksid=p2054897.c100204.m3164&\_trkparms=aid%3D222007%26algo%3DSIC.MBE%26ao%3D1%26asc%3D20140407115239%26meid%3D84f747b5347d40f98ed89762147e391f%26pid%3D100204%26prg%3D20140407115239%26rk%3D7%26rkt%3D30%26sd%3D111126810190 [Accessed 13 Nov. 2014].

Www2.its.strath.ac.uk, (2014). C Programming. [Online] Available at: http://www2.its.strath.ac.uk/courses/c/ [Accessed 9 Nov. 2014].

X-relsemi.com, (2014). Schematic Collections - XREL Semiconductor. [Online] Available at: http://www.x-relsemi.com/EN/Design-support/Schematic-Collections/ [Accessed 5 Nov. 2014].

3ders.org, (2014). 3ders.org - List of 3D Software. [Online] Available at: http://www.3ders.org/3d-software/3d-software-list.html [Accessed 13 Nov. 2014].

# Bibliography

Bridgman, R. (1993). Electronics. London: Dorling Kindersley.

Craft, B. (2013). Arduino projects for dummies. Chichester: Wiley.

Dewhurst, S. and Stark, K. (1989). Programming in C++. Englewood Cliffs, N.J.: Prentice Hall.

Hellman, E. and Hellman, E. (2014). Android programming. Chichester: Wiley.

McManus, S. and Cook, M. (2013). Raspberry Pi for dummies. Hoboken, NJ: John Wiley & Sons.

Redl, S., Weber, M. and Oliphant, M. (1998). GSM and personal communications handbook. Boston: Artech House.

Schildt, H. (2012). Java. New York: McGraw-Hill Osborne Media.

Wilcher, D. (2012). Learn Electronics with Arduino. Berkeley, CA: Apress.

# Appendix

**Updated Gantt chart of the smart socket project**

The Gantt charts 1.2, 1.3 and 1.4 below show the change in the timing of the task beginnings and endings according to real life events and work undertaken on the project, as seen in 1.2 the Design task began two weeks earlier due to the research accumulated at that stage, this has affected the whole duration of the project and shortening the expected completion date.



*Gantt 1.2 Project Gantt chart*

In the Gantt chart 1.3 the implementation period is extended due to unforeseen technical issues.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **Gantt chart smart socket** | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Week** | 0 | 2 | 4 | 10 | 12 | 14 | 16 | 18 | 20 | 24 |
|  |  |  |  |  |  |  |  |  |  |  |
| **Research** |  |  |  |  |  |  |  |  |  |  |
| **Plan** |  |  |  |  |  |  |  |  |  |  |
| **Design** |  |  |  |  |  |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |  |  |  |
| **Adjustments** |  |  |  |  |  |  |  |  |  |  |
| **Evaluation** |  |  |  |  |  |  |  |  |  |  |
| **Dissertation** |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

Gantt 1.3 Project Gantt chart

Lastly Gantt chart 1.4 shows the extension of the testing and adjustment period due to the unforeseen testing issues.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **Gantt chart smart socket** | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Week** | 0 | 2 | 4 | 10 | 12 | 14 | 16 | 18 | 20 | 24 |
|  |  |  |  |  |  |  |  |  |  |  |
| **Research** |  |  |  |  |  |  |  |  |  |  |
| **Plan** |  |  |  |  |  |  |  |  |  |  |
| **Design** |  |  |  |  |  |  |  |  |  |  |
| **Implementation** |  |  |  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |  |  |  |
| **Adjustments** |  |  |  |  |  |  |  |  |  |  |
| **Evaluation** |  |  |  |  |  |  |  |  |  |  |
| **Dissertation** |  |  |  |  |  |  |  |  |  |  |

Gantt 1.4 Project Gantt chart

**Gantt chart notes:**

**Research:** all the research that have to be undertaken to build the device and have a clear idea of its viability, usability, need and cost.

**Plan:** to prepare and devise a plan of action.

**Design:** to design all elements of the project and device.

**Implementation:** to implement the plan.

**Testing:** to test all aspects of the device and related aspects (code).

**Adjustments:** to adjust and solve any issues that arise from the testing process.

**Evaluation:** to evaluate the project by third parties.

**Dissertation:** to write a dissertation for the project.

**Objective Settings Proforma**

(to be completed and submitted to your supervisor by Friday 17th October, 2014)

Student’s Name: Amar Essa

First Assessor: Dr Peter Plassmann

Second Assessor: DrEmlyn Everitt

Project Title:

**Smart Socket**

Project Objectives & Deliverables

***Included below***

***Please tick this box to indicate your awareness of the university’s policy on ethical issues***

AE

***The deliverables and objectives can often change due to unforeseen circumstances, or through the student’s research causing the project to follow a different path. If this is the case, and the project objectives change significantly, then the first assessor should make a note of the date and fill in a new objectives proforma, which should also be included as an appendix to the project report. The project organiser is to be consulted at this stage.***

Appendix V

**Degree Scheme in Computing**

**FINAL Project Assessment and Comment Form 2014-15**

**Student Amar Essa**

**Project Title: Smart Socket**

**Supervisor: Peter Plassmann**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mark**  **Category** | **Proposed Weighting Ranges** | **Agreed**  **Weighting** | **Mark**  **Allocated** |
| **Project Management**  ***(Only set by Supervisor 1)*** | 50 – 80 | **50** |  |
| **Originality & Self-Direction** | 40 – 80 | **80** |  |
| **Technical Complexity** | 20 – 80 | **80** |  |
| **Solutions, Evaluation & Conclusions** | 80 – 120 | **90** |  |
| **Final & Sub-Report Quality** | 50 | **50** |  |
| **Prototype / System Demo**  **Or Project Deliverable** | 50 – 100 | **100** |  |
| **Sponsor Mark** | 00 – 60 |  |  |
| **Sub-Total Marks** | ------- |  |  |
| **Sub-Total Percentage** | ------- | **60%** | % |
| **Milestone 1 (initial)**  **Research**  **Research Applied to Design**  **Prototype Development** | ------- | **20%** | % |
| **Final Presentation (including poster production)** | ------- | **20%** | % |
| **TOTAL PERCENTAGE** | ------- | **100%** | % |

*The weightings for the various aspects of the project are to be set at the initial objectives setting stage. The ranges shown are guidelines and the actual weighting set may be outside of these ranges if deemed appropriate.*

Appendix VII

**STATEMENT OF ORIGINALITY**

**SCHOOL OF COMPUTING**

**DEGREE SCHEME IN COMPUTING**

**LEVEL SIX PROJECT**

This is to certify that, except where specific reference is made, the work described within this project is the result of the investigation carried out by myself, and that neither this project, nor any part of it, has been submitted in candidature for any other award other than this being presently studied.

Any material taken from published texts or computerised sources have been fully referenced, and I fully realise the consequences of plagiarising any of these sources.

Student Name (Printed) Amar Essa

Student Signature ………………………………..

Registered Scheme of Study Bsc Software Engineering

Date of Signing 15/3/2015