

**EEE502 - Embedded Systems Design**

**Arduino Password-Based Door Lock Security System**

**Coursework Documentation**

**Student Names: Radu Prajescu**

**Leontin Groza**

Table of Contents

[Table of Figures 3](#_Toc6532178)

[Introduction 4](#_Toc6532179)

[Embedded Systems Trends 4](#_Toc6532180)

[Background 4](#_Toc6532181)

[Development 5](#_Toc6532182)

[Project objectives 5](#_Toc6532183)

[Project Design Diagram 6](#_Toc6532184)

[Process Flow Diagram 7](#_Toc6532185)

[Hardware Required 8](#_Toc6532186)

[Solution 9](#_Toc6532187)

[Significant Code Snippets 9](#_Toc6532188)

[Project Design Diagram with Bluetooth Implementation 13](#_Toc6532189)

[Solution with Bluetooth Implementation 16](#_Toc6532190)

[Supplementary Hardware 16](#_Toc6532191)

[Code Snippet 16](#_Toc6532192)

[Issues 17](#_Toc6532193)

[Work Division 19](#_Toc6532194)

[Summary 19](#_Toc6532195)

[Course Work Complete Code Snippets 20](#_Toc6532196)

[Android Password-Based Door Lock Security System 20](#_Toc6532197)

[Android Password-Based Door Lock Security System with Bluetooth Implementation 33](#_Toc6532198)

[References 47](#_Toc6532199)

# Table of Figures

[Figure 1- Fritzing Diagram 6](#_Toc6530315)

[Figure 2 - Fritzing Schematics 7](#_Toc6530316)

[Figure 3 - Process Flow Diagram 8](#_Toc6530317)

[Figure 4 - "Back Gate" Process Flow Diagram 8](#_Toc6530318)

[Figure 5 - Project Diagram Implementation 9](#_Toc6530319)

[Figure 6 - Serial Bluetooth Terminal Software 13](#_Toc6530320)

[Figure 7 - Bluetooth Fritzing Diagram 14](#_Toc6530321)

[Figure 8 - Fritzing Bluetooth Schematics 15](#_Toc6530322)

[Figure 9 - Bluetooth Process Flow Diagram 16](#_Toc6530323)

[Figure 10 - Project Bluetooth Implementation 17](#_Toc6530324)

[Figure 11 - System cannot upload sketch 19](#_Toc6530325)

[Figure 12 - Program not responding after Bluetooth implementation 20](#_Toc6530326)

# 

# Introduction

Embedded System Devices have transformed the information technology industry. ESD had an impact over the industry by constantly thriving for increasing devices capabilities while extensively lowering the costs of communication devices resulting in an end product with functional components nearly unknown to the user. Embedded System Devices have the ability to alter interactions between the users and environment with the aid of minimised devices developed with the purpose of collecting, processing and sharing information.

The Embedded System Devices had a widespread in the industry due to the constant trend of device miniaturisation and the integration of development tools and universal standard communication systems. ESD has a different structure and functionality compared to standard distributed systems in multiple ways. Traditional systems are designed to link a fixed number of nodes in the schematics, according to the requirements and design, while ESD can interact with millions of nodes leading to a new method of node-to-node interaction. (Sehmi, 2013)

A key component of Embedded System Devices is the controller, a device automated and manipulated with a real-time operating system that implements an electrical or mechanical system developing a complete embedded system with hardware software and mechanical parts.

## Embedded Systems Trends

Embedded System Designs have a wide application in different industries such as commercial, medical and automotive industry. Many ESD's have an application in telecommunication beginning from switches, networks communications, network bridges, dedicated routers and the most common, cell phone. In the consumer industry, ESD's are applicated in video game consoles, digital cameras, printers and many other devices. Household appliances are extending the use of ESD in household appliances developed in devices such as washing machines and dishwashers and even microwaves. (Al-Ali, Al-Rousan and Al-Shaikh, 2003)

The transportation industry increases yearly the use in embedded system devices, new global positioning systems and guided systems are equipped on newly manufactured aeroplanes. Automobiles incorporate various embedded systems in the attempt to reduce pollution, traction control, vehicle braking systems and traction control, these are systems that implement embedded system devices. (Rouse, 2018)

# Background

Embedded System Devices have a wide application in the security field. From video surveillance, alarm systems and password-based locks, embedded system devices are the key component in their development. With the growing popularity of the Internet of Things, security has become an important aspect in many embedded systems. In the present day, everything is connected and while embedded systems stand at the base of the connection also stand exposed to the misuse risks from external factors.

The project’s target is to develop a minimal password-based lock system to demonstrate the ease and simplicity of creating a simple security gadget with the aid of embedded system devices. The device is developed to open and close a lock, a door or any other locking system, via a keypad input. The system is using authentication via a pin code, for simplicity the pin code is hardcoded in the software, and future programming can be added to allow multiple users and change of password features even for such a minimalistic device.

The gadget is constructed to provide feedback to the user, to help them understand the full process beginning with the input they provide. The hardcoded default password for the system will be “4567”, a subject of change by the developer at will. While a correct input is provided the system will provide feedback, through a liquid crystal display, confirming the correct pin number input. “Access Granted, Welcome” will be displayed and the servo motor will be rotated by 180 degrees. The system will also provide feedback about the locking process and the incorrect pin number input with the help of a buzzer added in the system which will sound when an incorrect combination is entered and will sound continuously while the input has been wrong three times in a row. If the pin number entered is incorrect, a “Code Incorrect, Go Away” will be displayed. Led lights have been incorporated into the system to complete the feedback process and to produce a lightning-based warning procedure throughout the whole system process.

A mandatory “back door” has been implemented into the design to allow user system access in case the user has forgotten the password. It will allow the user to have instant access. As described above, the system is minimal, and the option is developed to be known only to the user and to the developer. One simple button click will override the system and allow access. For market implementation, the feature must be developed to allow access either via user details input (ex. Date of birth), which can be hardcoded into the code or by using a simple Bluetooth module to send an access request to the user's mobile phone. Through mobile technology, the user can gain access to the system. Further system coding, upgrades and implementations can be added to the system, such as password change, mobile connectivity, add a new user or delete user, upgrades that are a must for the product to be released on the market.

# Development

## Project objectives

The beginning of the project mandates the development of a list of requirements based on the objectives of the system.

Requirements:

* the system must be intuitive and provide feedback on a liquid crystal display;
* pin password must be entered via keyboard;
* password must be displayed in a ‘\*’ format for security reasons;
* the system must show the processing functionality (led lights must be present at password input and server motor functions);
* for correct pin input the system must act upon a server motor mechanism (unlock);
* once the unlocking system process has been completed it must be followed by a locking system process;
* the system must provide an alarm warning for correct, incorrect pin input, server motor functions and warnings;

## Project Design Diagram

The diagram of the project has been created in specialised software, Fritzing, an open source development designed to create electronics design environments giving the users the possibility of learning, design and share their electronic models.

Project Diagram:

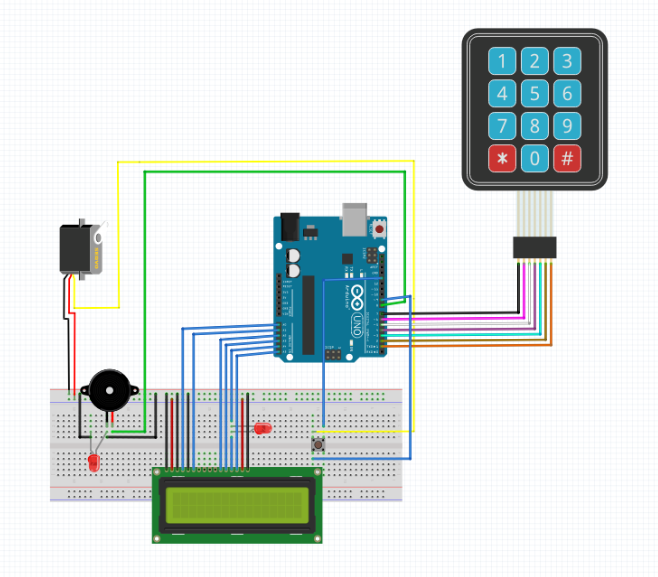


Figure 1- Fritzing Diagram

Along with the project diagram a schematic has been developed to view the total of nodes connected and the defined route of each of the nodes. The diagram offers a deeper insight into the components involved in the design along with every existing connection established between parts.

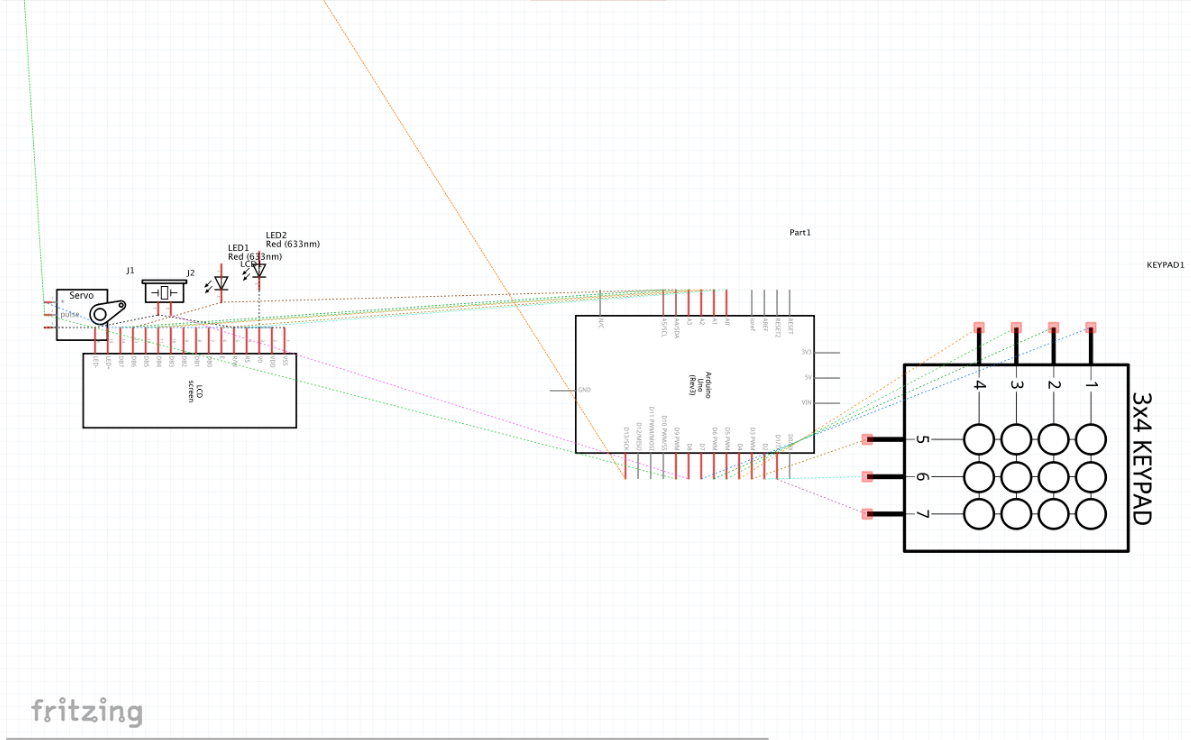


Figure 2 - Fritzing Schematics

## Process Flow Diagram

Below is the process flow diagram as seen by the developers.

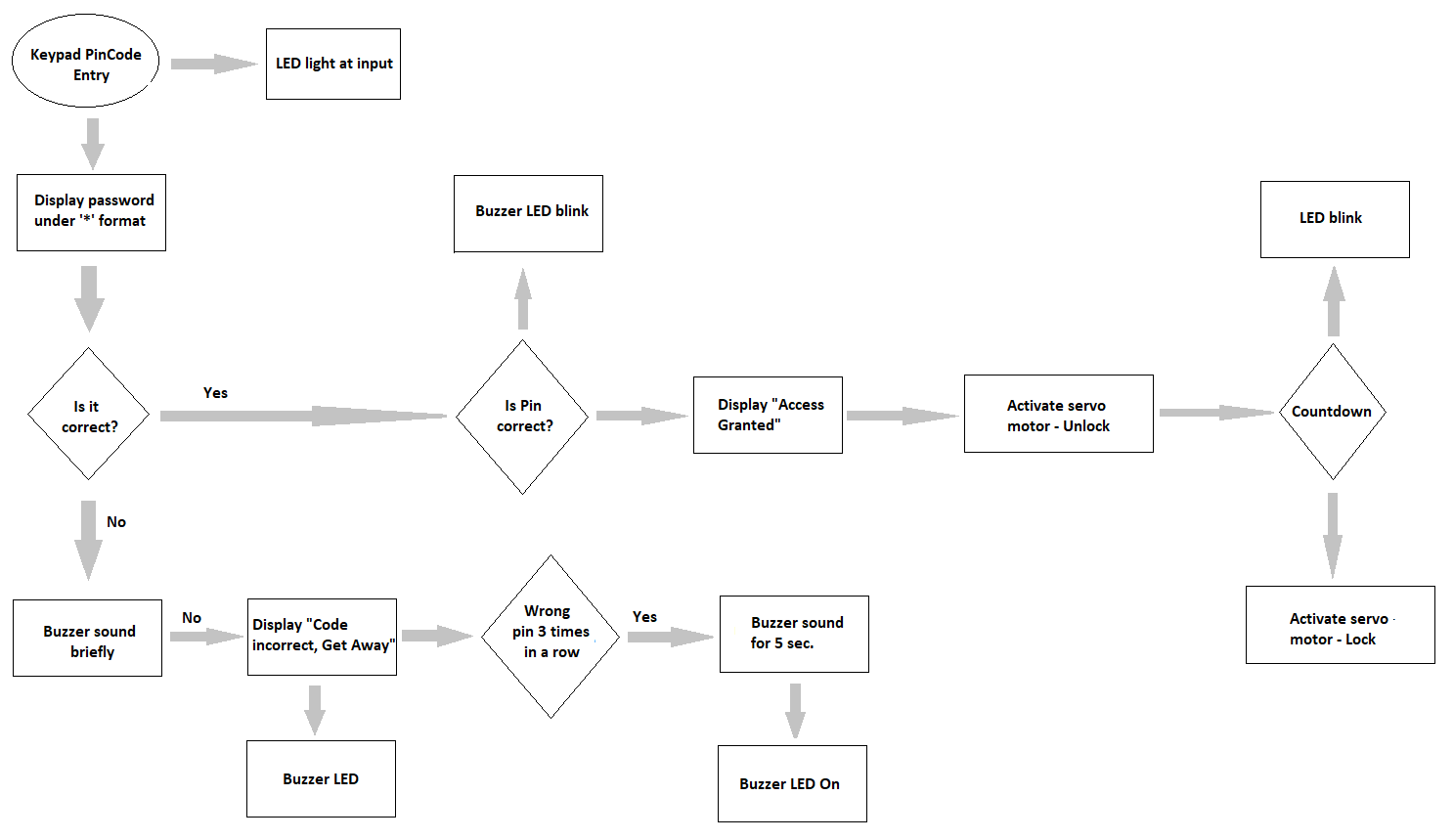


Figure 3 - Process Flow Diagram

The project flow diagram with a “back gate” is presented below.

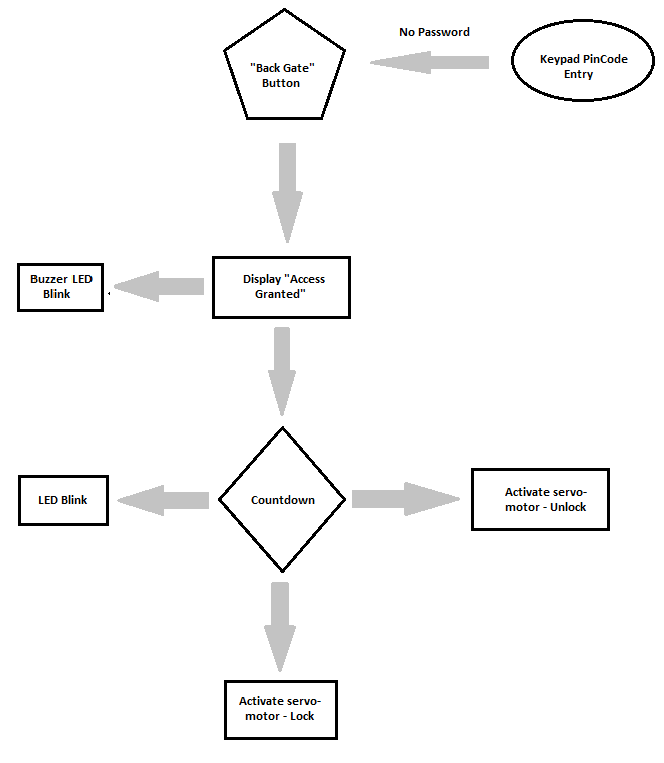


Figure - "Back Gate" Process Flow Diagram

## Hardware Required

List of the required project hardware:

* Arduino UNO;
* 4X4 Keypad;
* Liquid Crystal Display;
* Buzzer;
* Servo Motor SG90;
* Breadboard;
* Wire connections;
* 2 X LED;
* 9V battery for final implementation.

The project is simple. Arduino controls the process from taking the pin code input, comparing it with the hardcoded pin number, rotating the servo motor, displaying system status on the liquid crystal display and driving the buzzer. A keypad is used for input, buzzer and LED for warnings and indicators and servo motor to act on the mechanism.

## Solution

The final system solution designed with the project diagram is presented below. At the moment the power supply is provided through a cable connection to the computer, but the system is designed to stand alone, powered by a 9V battery as a power supply.

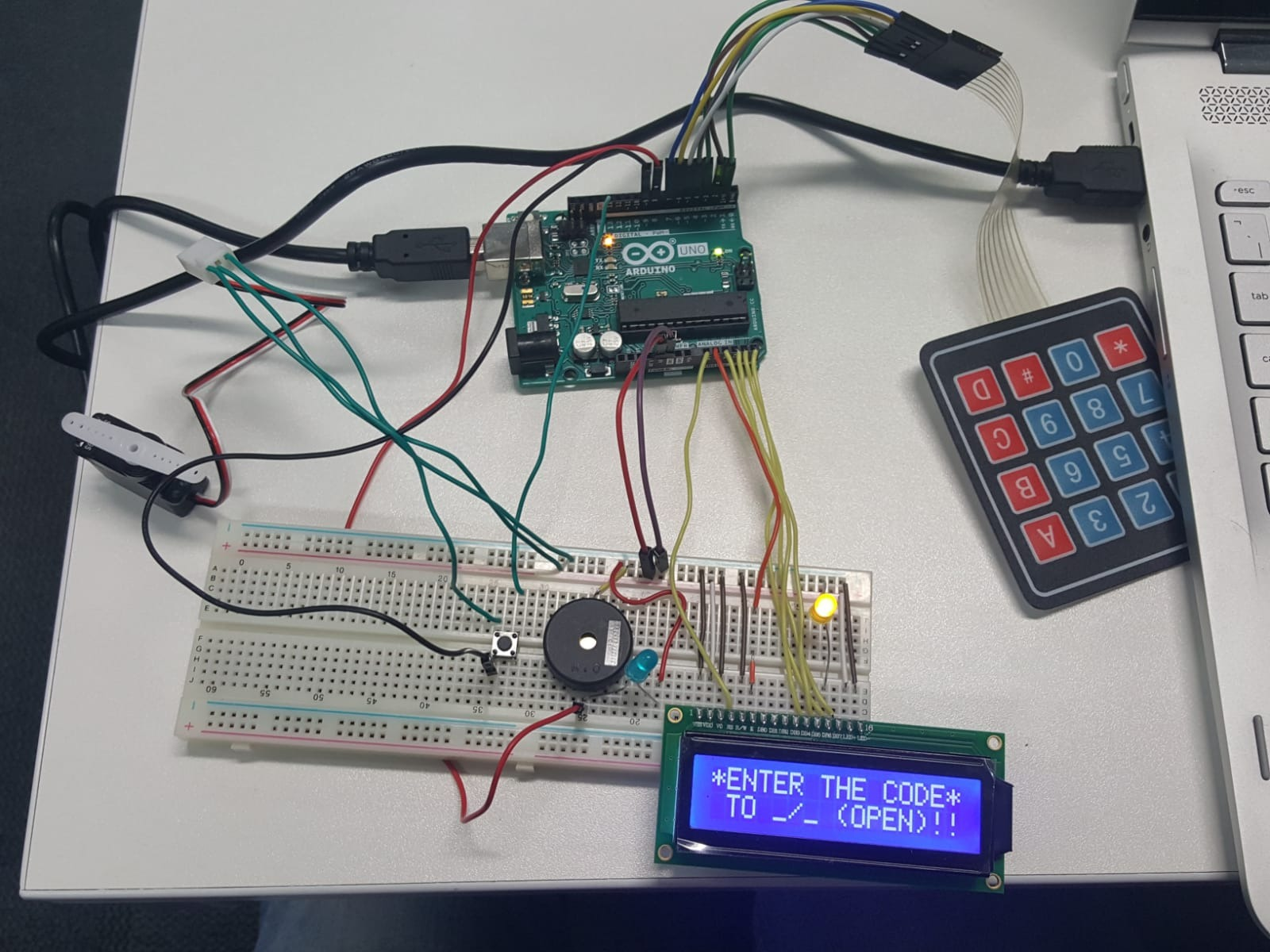


Figure 5 - Project Diagram Implementation

## Significant Code Snippets

The complete coursework project will be submitted via blackboard together with the developed code from which some examples have been extracted and presented below.

This method, in case of correct pin input, will determine the position of the servo motor and will determine a rotation from 0 to 180 degrees to act as an unlock mechanism. Once the unlock process is completed it will access a counter method that keeps the servo motor in position for a time period of 5 seconds before returning to the initial locking position. This is the end result of the complete process.

*“void unlockdoor()*

*{*

*delay(900);*

*lcd.setCursor(0,0);*

*lcd.println(" ");*

*lcd.setCursor(1,0);*

*lcd.print("Access Granted");*

*lcd.setCursor(4,1);*

*lcd.println("WELCOME!!");*

*//////*

*for(pos = 180; pos>=0; pos-=5) // goes from 180 degrees to 0 degrees*

*{*

*myservo.write(pos); // tell servo to go to position in variable 'pos'*

*delay(5); // waits 15ms for the servo to reach the position*

*}*

*//////*

*for(pos = 0; pos <= 180; pos +=5) // goes from 0 degrees to 180 degrees*

*{ // in steps of 1 degree*

*myservo.write(pos); // tell servo to go to position in variable 'pos'*

*//////*

*}*

*}”*

The following method is designed to check the pin number input by the user and check it with the existing pin number hardcoded in the software. If the input is not equal to the hardcoded pin the system will display a warning message returning the user to the start of the process. If the pin number has been entered incorrect three times in a row the buzzer will sound for a period of 3 second along with LED light warnings after which the process will reset to the initial state.

void incorrect()

{

//////

lcd.print("CODE");

lcd.setCursor(6,0);

lcd.print("INCORRECT");

lcd.setCursor(15,1);

lcd.println(" ");

lcd.setCursor(4,1);

lcd.println("GET AWAY!!!");

lcd.setCursor(13,1);

lcd.println(" ");

Serial.println("CODE INCORRECT YOU ARE UNAUTHORIZED");

//////

}

The last code snipper presented is designed to start once the pin number has been entered more than three times. It is the initial method for wrong input that will keep the system locked from the user for a total period of 20 seconds. The method will display several messages to the user.

An important part of the code refers to the instant system access granted to the user by a simple button push. A simple implementation that will override the system in case the user forgot the password and seeks immediate access.

// initialize the Servo Motor as an output:

pinMode(9,OUTPUT);

// initialize the pushbutton pin as an input:

pinMode(buttonPin, INPUT);

//process to be carried out on button push:

attachInterrupt(digitalPinToInterrupt(9),unlockdoor,OUTPUT);

The system will check if the button is in its initial state if true the system will follow its normal flow control, otherwise, the access procedure will be enabled.

void loop()

{

////////////////////////

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

if (buttonState == HIGH) {

//follow program flow

}

else {

// unlock

unlockdoor();

}

// LOOP ENDS!!!//

}

## Project Design Diagram with Bluetooth Implementation

An implementation of a Bluetooth module, for the system, has been developed. The Bluetooth module stands as a safety measure in case the user has forgotten the password. The procedure is simple, the “back gate” button has been removed from the design and a Bluetooth module has been added to allow the user a remote connection via a mobile device with the Arduino software. The Bluetooth connection is stablished via a mobile app specifically designed for Arduino, Serial Bluetooth Terminal that will allow the user to send directives to Arduino to unlock the system. The directives are coded in Arduino, if the user inputs “1” the system will unlock, if the user sends “0” the system will lock. The directives can be replaced with user details such as date of birth, but for the development process the input expected from the user will be simple.

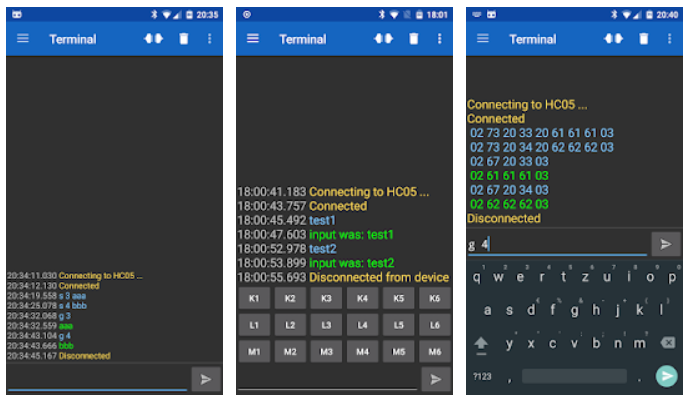


Figure - Serial Bluetooth Terminal Software

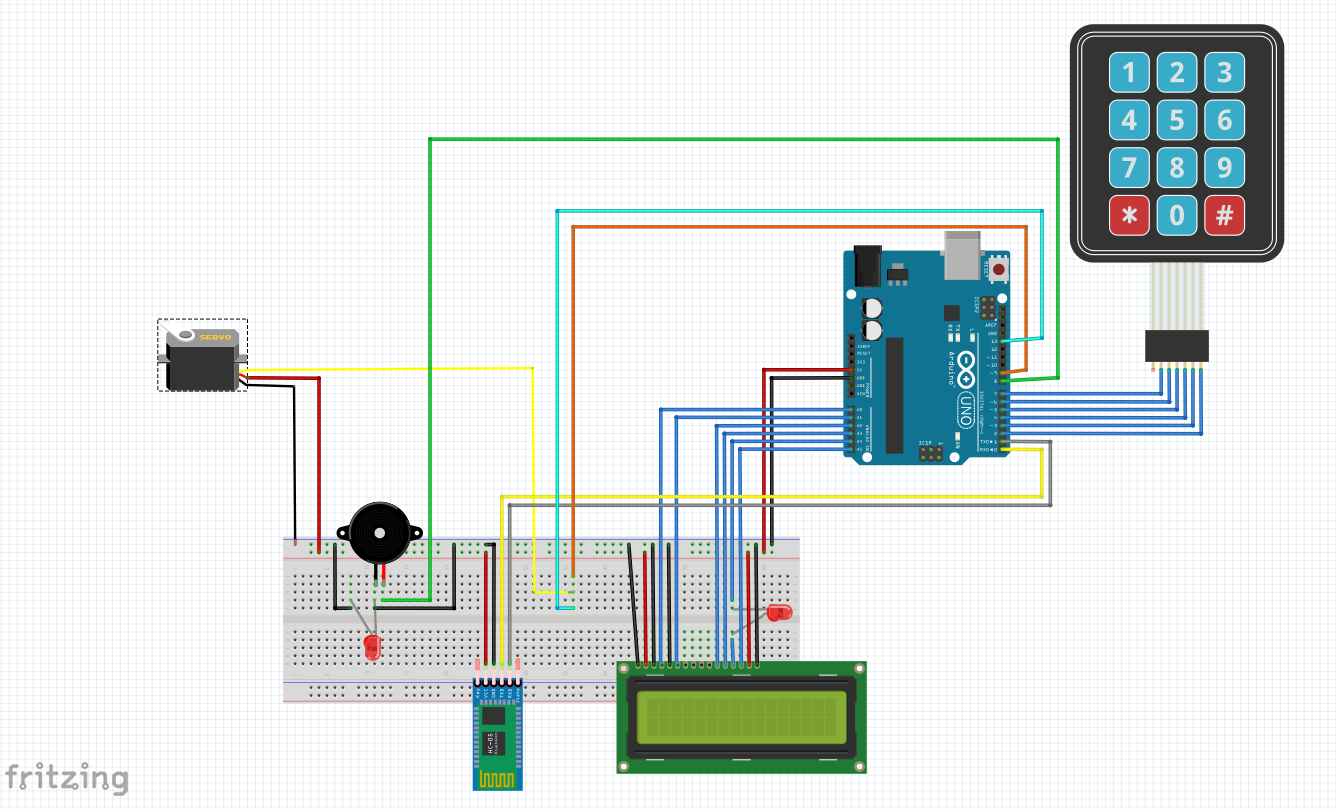


Figure - Bluetooth Fritzing Diagram

Along with the Fritzing Bluetooth diagram a schematic has been generated to reveal the total number of nodes and the connection established between them.

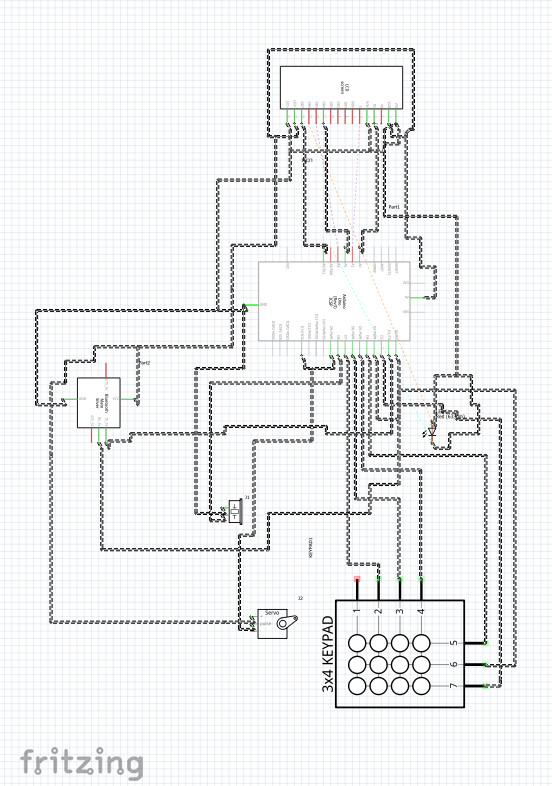


Figure - Fritzing Bluetooth Schematics

The Bluetooth flow diagram in continuation of the main project flow diagram is presented below.

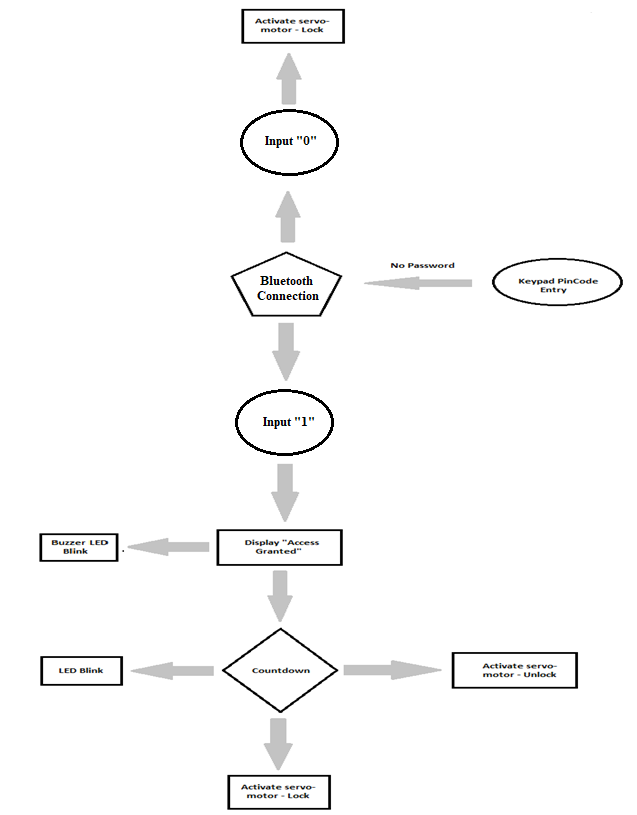


Figure - Bluetooth Process Flow Diagram

## Solution with Bluetooth Implementation

The following image present the solution for Arduino password-based security door lock system with Bluetooth incorporated.

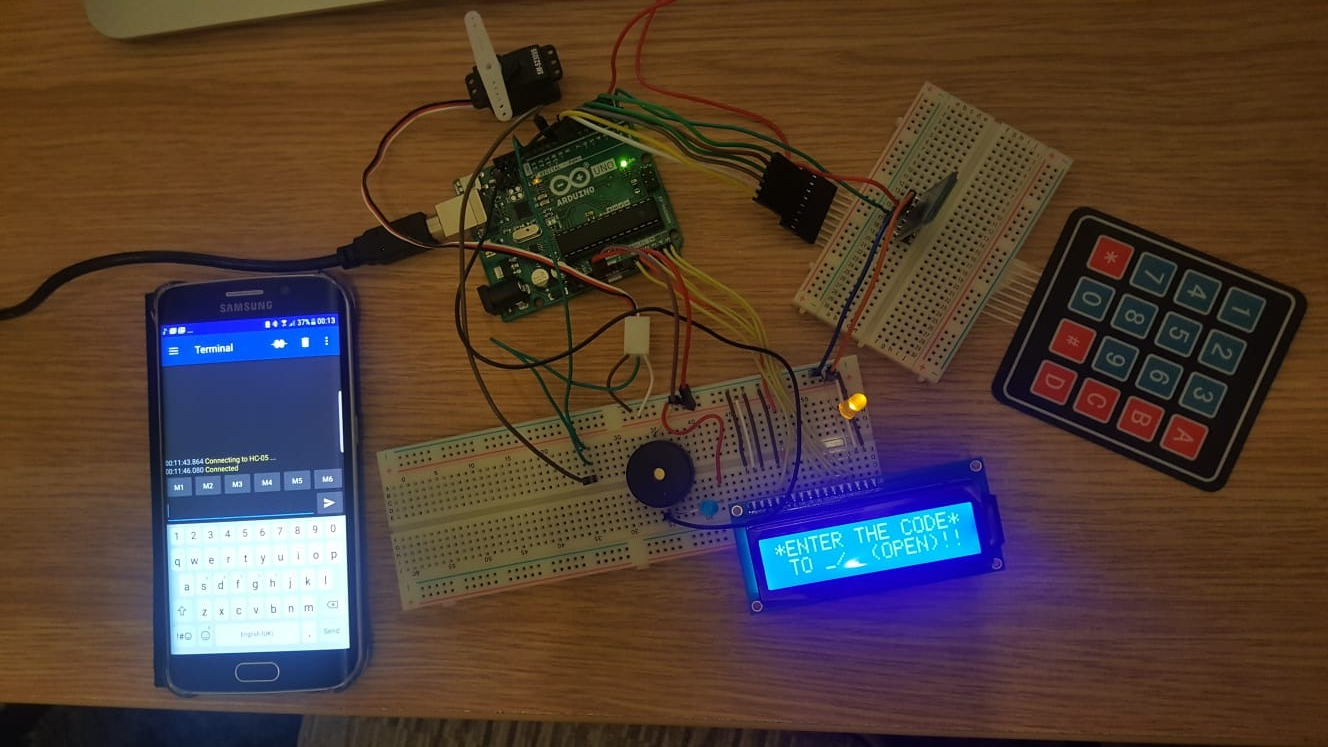


Figure - Project Bluetooth Implementation

As stated above, it is a simple process, if the user lost the password, a Bluetooth connection is available to connect Arduino to a mobile device via the Serial Bluetooth Terminal from where the user can unlock or lock the system. During development, simple number input is available for locking and unlocking the system. To enhance the security of the system, specific details known only by the user must be hardcode in the software, to be requested by the system from the user in an emergency, otherwise the system will remain locked.

## Supplementary Hardware

The Bluetooth implementation is developed on the original Arduino password-based door lock security system and the extra hardware required is an Arduino HC-05 Bluetooth module.

## Code Snippet

Following the implementation of the Bluetooth module in the system supplementary codding was required to complete the functionality of the system.

Bluetooth.begin(9600);

Serial.begin(9600);

Serial.println("Waiting for command...");

Bluetooth.println("Send 1 to turn on the LED. Send 0 to turn Off");

pinMode(LED,OUTPUT);

//////////////////////////////////////////////////////

if (Bluetooth.available()){ //wait for data received

Data=Bluetooth.read();

if(Data=='1'){

digitalWrite(LED,1);

Serial.println("LED On!");

Bluetooth.println("LED On!");

}

else if(Data=='0'){

digitalWrite(LED,0);

Serial.println("LED Off!");

Bluetooth.println("LED On D13 Off ! ");

}

else{;}

}

delay(100);

The code snippet initialises the Bluetooth module after connection to a mobile device. After the connection has been established the device awaits input from the user and based on the input it will control the device.

## Issues

The functionality issue of the system implemented with a Bluetooth module is presented in a video via the following GoogleDrive link: https://drive.google.com/open?id=1XlAXLBnCBpkCEHWh2mCJVREPaVfXdWt8

The code is fully functional, the Bluetooth device is pairing with the mobile device, but it will not communicate with the device. String messages have been added in the code to observe the communication between the Bluetooth module and the device but appears to be inexistent.

Serial.println("Waiting for command...");

Bluetooth.println("Send 1 to turn on the LED. Send 0 to turn Off");

The Bluetooth module can also be controlled by Arduino Serial Port, but once the code has been compiled and send to the device, with the Bluetooth module implemented, it will not open presenting “COM3 Busy” message. With the Bluetooth module implemented in the system, several messages are displayed in Arduino console logs.

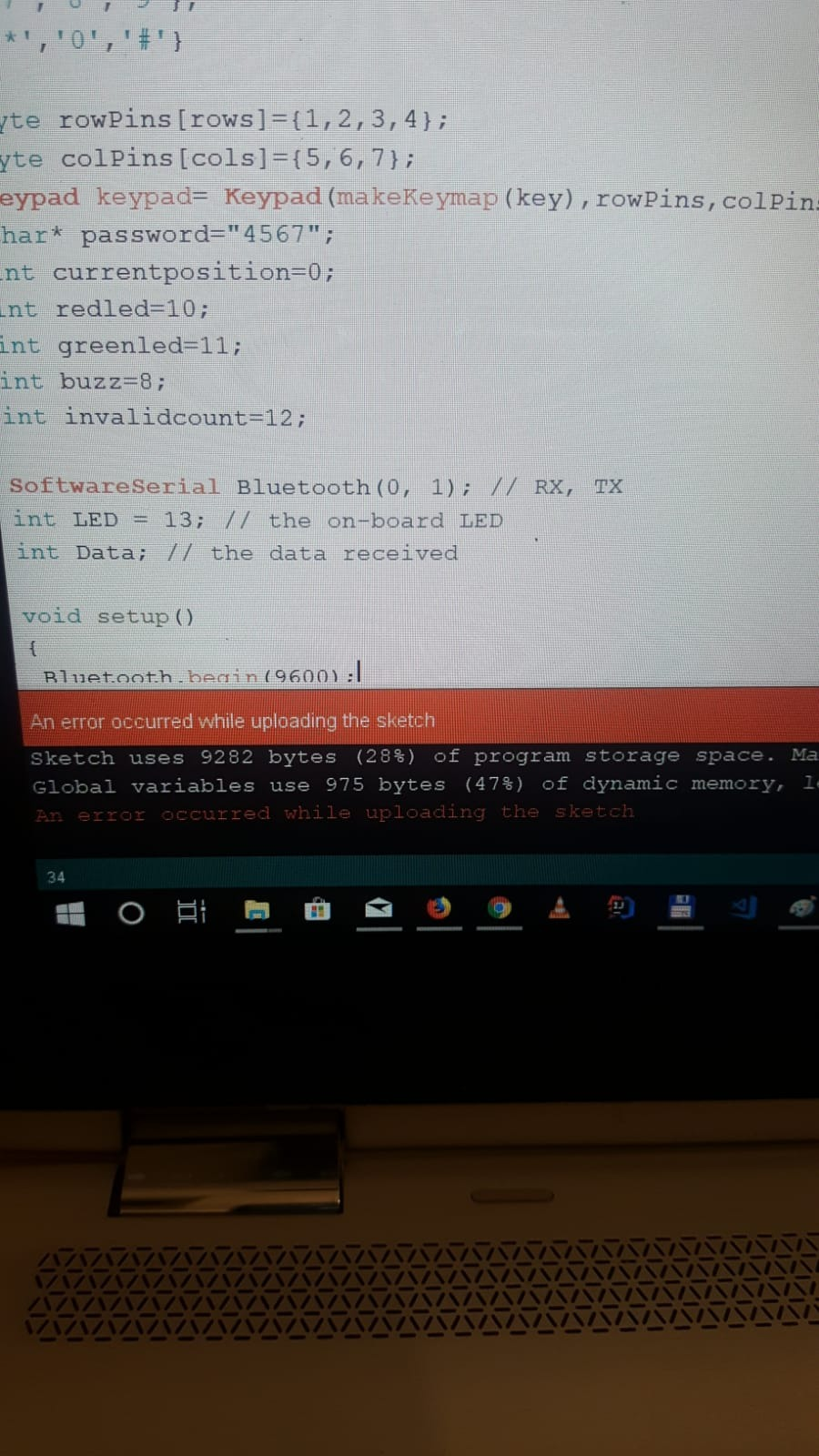


Figure - System cannot upload sketch

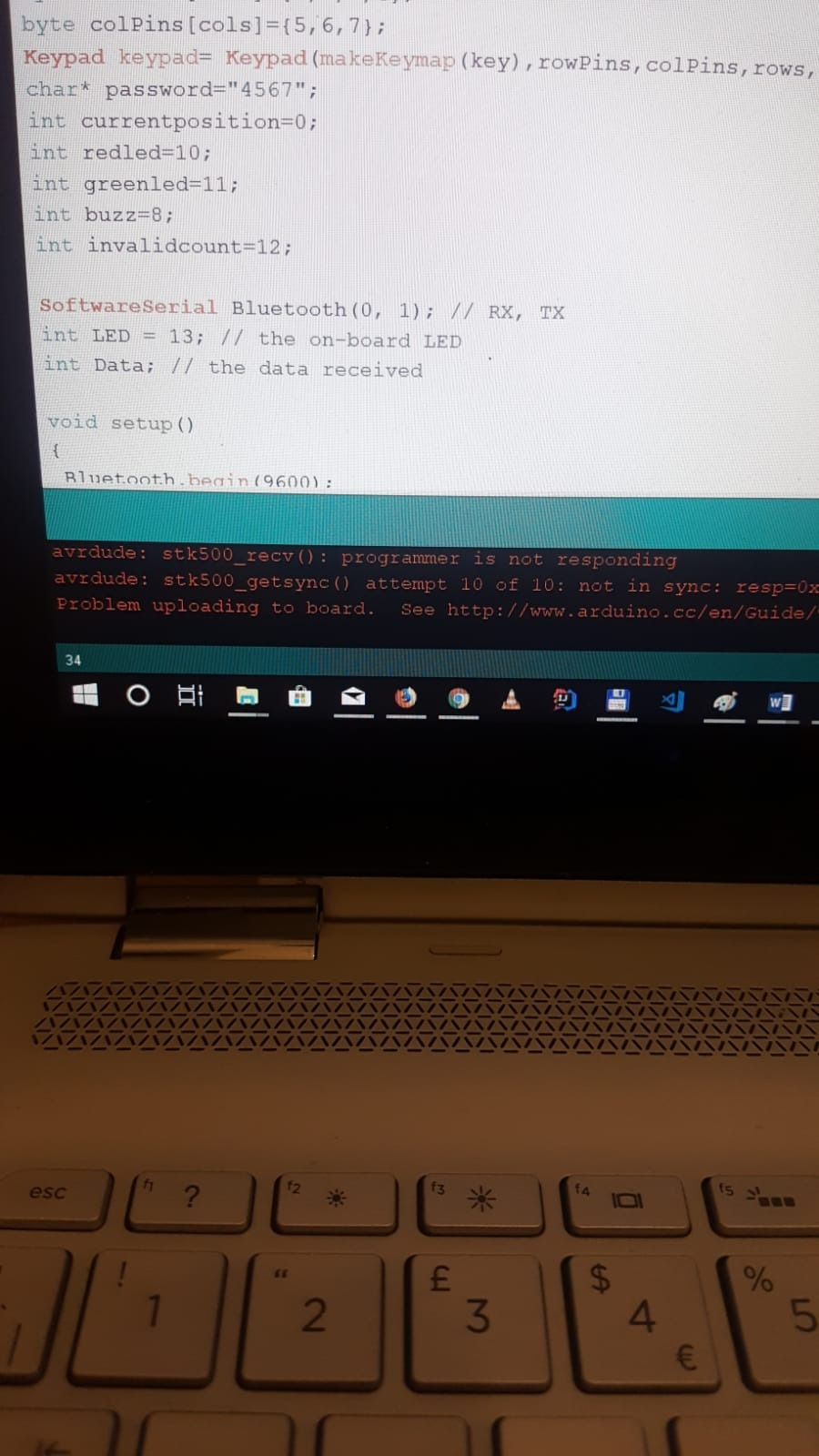


Figure - Program not responding after Bluetooth implementation

As soon as the Bluetooth module has been powered off or removed from the system, the functionality of the software is normal. Tests have been carried out to check the functionality of the modules. Individually, on simple implementations, the modules function normally, implemented in a more complex design, a synchronisation issue interferes with the functionality of the system.

## Work Division

The work on the project has been divided evenly and the development of the project has been carried out as a group at the university. Radu has been working on documentation, project diagram and Bluetooth implementation in the system while Leo was in charge of the physical implementation of the project, the system flow diagram and contributing at documentation. Every issue found during development has been discussed and decisions have been made as a group.

# Summary

During system development, we encountered a few hardware problems. The main issue was related to the breadboard, a large MB-102 model with 830 solderless points. The system is designed for standalone use, powered by a 9V battery, but during implementation, the power source was provided by the computer. This had an impact on functionality at first, as the components being spread around over the full breadboard the servomotor was not provided with enough voltage to function. Component testing was required to get an insight into the problem but eventually, we managed to realise what was happening and reconnected the components to develop a functional system.

At the completion of the project, our knowledge towards embedded system device greatly increased. During development, we have experienced problems with coding and wiring but with time and effort, we managed to overcome the problems and completing the coursework. We managed to gain an understanding of the common functionality of ESD and understanding their applications in the real world.

## Course Work Complete Code Snippets

The following GoogleDrive link is provided to access the code for the original Arduino Password-Based Door Lock Security System and together with the Bluetooth Implementation:

https://drive.google.com/open?id=1M6xkEkDpEi0ZtKh7UrhEpbDMnPZzfhYJ

# References

Sehmi, A. (2013). On Distributed Embedded Systems. *Advances in Computer Science: an International Journal, 2*(1), 30-39. Retrieved 3 22, 2019, from http://acsij.org/documents/v2i1/5. on distributed embedded systems.pdf

Al-Ali, A. R., Al-Rousan, M. and Al-Shaikh, M. ‘Embedded system-based mobile patient monitoring device’, in *16th IEEE Symposium Computer-Based Medical Systems, 2003. Proceedings.* IEEE, pp. 355–360. two

Rouse, M. (2018) *What is microcontroller? - Definition from WhatIs.com*. Available at: https://internetofthingsagenda.techtarget.com/definition/microcontroller (Accessed: 22 March 2019).