

DESIGN AND DEVELOPMENT OF A MOBILE CONTROLLED DOOR LOCK SYSTEM

A dissertation submitted to the
EAST WEST UNIVERSITY
for the degree of
B.Sc. in Electronics and Telecommunication Engineering



❖ SUBMITTED BY:

➤ Name: Mahinul Mannan

ID: 2011-3-55-004

➤ Name: Md. Nabinur Rahman

ID: 2011-2-55-005

❖ Supervised by:

Dr. Md. Habibur Rahman

Professor

Dept. of Electrical and Electronic Engineering

University of Dhaka

DECLARATION

With due respect, we would like to declare that this project on “DESIGN AND DEVELOPMENT OF A MOBILE CONTROLLED DOOR LOCK SYSTEM” has been done in the Department of Electronics and Communication Engineering, East West University, under the supervision of Dr. Md. Habibur Rahman. Any part of this report has not been submitted to elsewhere for any other degree and all sources of knowledge used here have been truly acknowledged.

Sincerely Yours:

.....

Md. Nabinur Rahman

ID:2011-2-55-005

Mahinul Mannan

ID.2011-3-55-004

APPROVAL

This is to certify that the project titled as “Design and Development of a Mobile Controlled Door Lock System” submitted to respected members of the Board of Examiners of the Faculty of Engineering for partial fulfillment of the requirements for the Degree of Bachelor of Science in Electronics and Telecommunication Engineering by the following students and has been accepted as satisfactory.

Submitted by:

Md. Nabinur Rahman

ID.2011-2-55-005

Mahinul Mannan

ID.2011-3-55-004

.....
Dr. MD. Habibur Rahman

Professor, Electrical and Electronic Engineering

University of Dhaka

.....
Dr. Mofazzal Hossain

Chairperson and Associate professor,

Dept. of Electronics and Communication

Engineering

East-West University, Dhaka

Acknowledgements

All eulogy and thanks to almighty Allah who is our creator, sustainer, nourisher, protector and curer.

I would like to reveal my sincere gratitude and appreciation to my respected supervisor Dr. Md. Habibur Rahman for his continuous support throughout the academic journey.

I would also like to thank all the honorable teachers of the Department of Electronics and Communication Engineering, of East-West University for their inspirations and advices.

I am also thankful to the laboratory staffs of our department for their fast response and cooperation.

At the end my big thanks and respects goes to my parents for their constant support and encouragement throughout the study period.

The author

ABSTRACT

Mobile controlled lock system refers to the automatic and electronic control of doors and some other appliances. With the development of modern technology, smart phones have become a necessity for everyone in this world. Smart phone affordability is also increasing every year. Android operating system (OS) of Google is the mostly used operating system in smart phones and tablets around the world. Controlling home door appliances by using an Android device gives the users convenience, security and safety. In this project, a Bluetooth-based home automatic door lock system has been designed and developed using Android device. This system includes an Android device (Smart phone or Tablet), a Bluetooth module, Arduino Uno R3 board, servo motor and various home appliances. A mechanical lock has been developed which is suitable for this project. The mechanical lock is attached to the door and interfaced with a driving servo motor. The servo motor is rotated clockwise and anticlockwise by the Arduino Uno controller. The Bluetooth module is also interfaced with the Arduino Uno board. The Bluetooth technology is used to establish communication between the Android device and Arduino board. Blueterm, an Android application, is used in android phone to lock and unlock the desired door. When the buttons of the App are pressed on, corresponding Bluetooth signals are sent from the Android device to the Bluetooth module which is connected to the Arduino. The Arduino finds out which signal has been sent and compares it to the predefined signals assigned for each appliance. The complete system has been developed and found completely functioning. The most interesting feature of this developed system is, only the owner can lock and unlock his door. Because there is a password for more security. If the password is correct only then, the door will be locked or unlocked.

CONTENTS

Chapter 01

1.1 Introduction	09
1.2 Objective	09
1.3 Literature Review	09
1.4 Android Control Automobile	10

Chapter 2(Theory Revised)

2.1 Microcontroller Arduino	12
2.1.1 What is Arduino?	12
2.1.2 History of Arduino-UNO	13
2.1.3 Why arduino?	13
2.1.4 Different types of aurdino boards	14
2.1.5 Different parts of Arduino UNO	15
2.1.6 Description of parts Arduino UNO	16
2.1.8 Arduino IDE	18
2.1.9 Hardware features and Software Features	18
2.1.10 Plugging the Arduino into the computer	19
2.1.11 Advantages of Microcontroller	19
2.2 Servomotor	20
2.2.1 What is Servo motor?	20
2.2.2 How does servo motor works?	21
2.2.3 Important notes	22

2.3 Hc-05 Bluetooth Module	23
2.3.1 Pins	24
2.3.2 Hardware Features	24
2.3.3 Software Features	25
2.3.4 Schematic Diagram of HC-05 Module	25
2.4 How does bluetooth works?	26
2.5 Overview of existing system	26
2.6 Existing door lock system	27
2.7 Description of blocks	27
2.8 Feature of the proposed system	29

Chapter 03

3.1 Design and Development of the System	30
3.1.1 Full Circuit Design	30
3.2 Description of Each Block	30
3.2.1 Equipment Used in this System	30
3.2.2 Connection of Arduino with Bluetooth Device	34
3.2.3 Connection of Arduino with Servo Motor	35
3.3 Development of Software	35
3.3.1 Installing Arduin	36
3.3.2 Installing Blueterm	36
3.3.3 Verifying the Hardware	36
3.3.4 Arduino Language	36
3.3.5 Logical Diagram	36
3.3.6 Flow Chart	37

3.3.7 Programming Codes	38
-------------------------	----

Chapter 04

4.1 Implementation	40
--------------------	----

4.2 Results	40
-------------	----

Chapter 05

5.1 Conclusion	42
----------------	----

5.2 Future Work Scope	42
-----------------------	----

REFERENCES	43
-------------------	-----------

Chapter 01

1.1 INTRODUCTION

Door security is top priority of all concerned. Today there are plenty of home or office security products to ensure our family's or business security completely. Security is a prime concern in our day-to-day life. Everyone wants to be as much secure as possible. So they want to make their home or office more secure. Home door security is the most significant one for every home/office owner either in an individual house or an apartment. For that different security systems are available. An access control for doors forms a vital link in a security chain. The mobile controlled Door lock is an access control system that allows only authorized persons to access a restricted area.

1.2 OBJECTIVE

The aim of this project is to design and develop a low cost prototype of Bluetooth based door system using an Android device. The design is based on an Arduino Uno R3 board. The home appliances are connected to the input/output ports of this board via Bluetooth module. The communication between the Android device and the Arduino BT board is wireless. An Android application is used to provide user interface through which the user can send signals easily for controlling door app.

1.3 LITERATURE REVIEW

Electronic door locking is the primary requirement in home automation system. Over the years many technologies have emerged like password based, fingerprint based and Internet(ZigBee) based systems for door locking. These technologies have certain limitations where in the user cannot get the status of the system. In this project titled "DESIGN AND DEVELOPMENT OF A

MOBILE CONTROLLED DOOR LOCK SYSTEM” aims at controlling the door from a remote location by sending a SMS and also provides the intimation to the user.

Smart Home is the term commonly used to define a residence that uses a home controller to integrate the residence's various home automation systems. The most popular home controllers are those that are connected to a Windows based PC. In our research we presented a part of smart home technology which using Bluetooth in a mobile device, so it will more easy and efficient to use. It also based on Android and Arduino platform both of which are free open source software. In this paper, a system called door locks automation system using Bluetooth-based Android Smartphone is proposed and prototyped. First the hardware design and software development are described, then the design of a Bluetooth-based Smartphone application for lock/unlock the door are presented. The hardware design for door-lock system is the combination of android smart phone as the task master, Bluetooth module as command agent, Arduino microcontroller as controller center / data processing center, and solenoid as door lock output. All of the tests indicate that all goes according to the initial design of this research.

Keywords: android, Arduino Uno, door automation, Bluetooth, smart phone, servomotor based and motor driven IC based systems for door locking. These technologies have certain limitations where in the user cannot get the status of the system. In this project aims at controlling the door from a remote location by sending a SMS and also provides the intimation to the user regarding the status.

1.4 Android Controlled Automobile

An Android controlled automobile project that allows user to control a battery powered automobile wirelessly through an android application. The system uses a Bluetooth modem as a medium to transmit signals between the android based phone and automobile. The android application provides a GUI to the user. The commands sent through GUI are sent in the form of string variables by the android application through an active Bluetooth connection. These string variables are then received by Bluetooth modem and then forwarded to the Microcontroller. The microcontroller now processes the data sent by Bluetooth modem and checks for user commands. On recognizing direction commands sent by user (Forward |Backward |Left |Right), the microcontroller sends

signals to appropriate driver IC's. The driver IC's now operate motors to achieve the desired movement of the automobile as sent by microcontroller. This process is continuously repeated by the system to achieve appropriate vehicle movement as per user commands through the android application.

Chapter 02 || Theory Revised

2.1 Microcontroller ARDUINO

2.1.1 What is Arduino?

An Arduino is an open-source microcontroller development board. Arduino consists of both a physical programmable circuit board and a piece of software or IDE (Integrated Development Environment) that runs on the computer, used to write and upload computer code to the physical board.

Arduino can be used to read sensors and control things like motors and lights. This allows us to upload programs to this board which can then interact with things in the real world. With this, we can make devices can be made which respond and react to the world at large.

For instance, we can read a humidity sensor connected to a potted plant and turn on an automatic watering system if it gets too dry. Or, we can make a stand-alone chat server which is plugged into our internet router. Or, we can have it tweet every time our cat passes through a pet door. Or, we can have it start a pot of coffee when the alarm goes off in the morning.

Basically, if there is something that is in any way controlled by electricity, the Arduino can interface with it in some manner. And even if it is not controlled by electricity, we can probably still use things which are (like motors and electromagnets) to interface with it.

The possibilities of the Arduino are almost limitless. I've done my best to give a basic overview of the fundamental skills and knowledge that one need to get an Arduino up and running. If nothing more, this should function as a springboard into further experimentation and learning.

2.1.2 History of Arduino-UNO

The Arduino is developed in 2005. The Arduino microcontroller was initially created as an educational platform for a class project at the Interaction Design Institute Ivrea in Milan (Italy) in 2005. It derived from a previous work of the Wiring microcontroller designed by Hernando

Barragan in 2004. From the beginning, the Arduino board was developed to attract artists and designers. The Wiring microcontroller was created by Hernando Barragan to be used for parsing data to electronic devices. His aim was that it could be used by non-technical people who only had basic experience with using computers. He first of all wanted it to be used as a prototyping tool. Since he needed help to create an easy software tool to programmed the board he engaged Casey Reas and Massimo Banzi as his assistants. Reas created the visual programming language for the prototyping tool.

2.1.3 Why Arduino?

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handy board and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive: Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than BDT 800. **Cross-platform:** The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows. **Simple, clear programming environment:** The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino. **Open source and extensible software:** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be

expanded through C++ libraries and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, AVR-C code can be added directly into the Arduino programs if one wants to. Open source and extensible hardware: The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

2.1.4 Different Types of Arduino Boards:

There are a number of different types of Arduino boards to choose from. This is a brief overview of some of the more common types of Arduino boards one may encounter. The name and description of some common Arduino boards are given below.

Arduino Uno: The most common version of Arduino is the Arduino Uno. This board is what most people are talking about when they refer to an Arduino. In the next step, there is a more complete rundown of its features.

Arduino NG, Diecimila, and the Duemilanove (Legacy Versions): Legacy versions of the Arduino Uno product line consist of the NG, Diecimila and the Duemilanove. The important thing to note about legacy boards is that they lack particular feature of the Arduino Uno. Some key differences: The Diecimila and NG use an ATMEGA168 chips (as opposed to the more powerful ATMEGA328), Both the Diecimila and NG have a jumper next to the USB port and require manual selection of either USB or battery power. The Arduino NG requires that we hold the reset button on the board for a few seconds prior to uploading a program.

Arduino Mega 2560: The Mega is the second most commonly encountered version of the Arduino family. It boasts 256 KB of memory (8 times more than the Uno). It also had 54 input and output pins, 16 of which are analog pins, and 14 of which can do PWM. However, all of the added functionality comes at the cost of a slightly larger circuit board. It may make the project more powerful, but it will also make the project larger.

Arduino Mega ADK: This specialized version of the Arduino is basically an Arduino Mega that has been specifically designed for interfacing with Android smartphones.

Arduino LilyPad: The LilyPad was designed for wearable and e-textile applications. It is intended to be sewn to fabric and connected to other sew able components using conductive thread. This board requires the use of a special FTDI-USB TTL serial programming cable.

2.1.5 Different parts of Arduino UNO Board

Power (USB Interface/ External Power Supply): Every Arduino board needs a way to be connected to a power source. The Arduino Uno can be powered from a USB cable coming from a computer or a wall power supply that is terminated in a barrel jack. The USB connection is also how the code is loaded onto the Arduino board. The recommended voltage for most Arduino models is between 6 and 12.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF): The pins on the Arduino are the places where wires are connected to construct a circuit. They usually have black plastic ‘headers’ that allow users to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

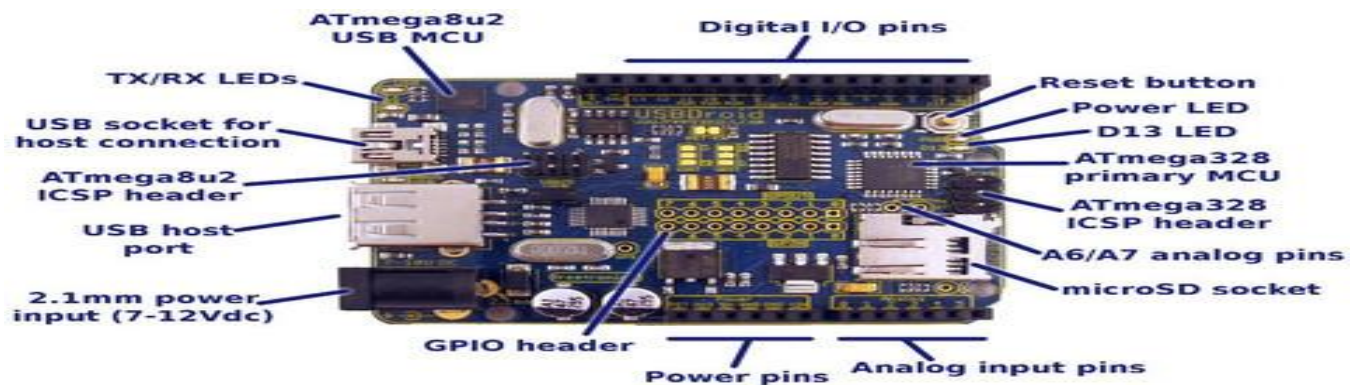


Figure 2.1: Different part of the Arduino UNO

2.1.6 Description of Different Parts of Arduino Module

GND: Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground the circuit. **5V & 3.3V:** the 5V pin supplies 5 volts of power and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts. **Analog Pins:** The pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog Input pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read by the Arduino. **Digital Pins:** Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input and digital output. **PWM:** Some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO) have tildes (~) next to them. These pins act as normal digital pins, but can also be used for Pulse-Width Modulation (PWM). These pins are able

Reset Button: Arduino has a reset button. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if the code doesn’t repeat, but we want to test it multiple times.

Test LED 13: Just beneath and to the right of the word “UNO” on the circuit board, there’s a tiny LED next to the word ‘ON’. This LED should light up whenever Arduino is plugged into a power source. If this light doesn’t turn on, something is wrong.

TX / RX LEDs: TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In this case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs. These LEDs will give us visual indications whenever the Arduino is receiving or transmitting data.

Main IC: The black thing with all the metal legs is an IC, or Integrated Circuit (13). It is the brains of the Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC’s from the ATMEL Company. This can be important to know the IC type (along with the board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC.

2.1.7 Arduino IDE

Before we can start doing anything with the Arduino, we have to download and install the Arduino IDE (integrated development environment). Arduino IDE is referred as the Arduino Programmer. The Arduino Programmer is based on the Processing IDE and uses a variation of the C and C++ programming languages.

An Arduino program is called a sketch. All code in an Arduino sketch is processed from top to bottom. Arduino sketches are typically broken into five parts.

The sketch usually starts with a header that explains what the sketch is doing, and who wrote it.

Next, it usually defines global variables. Often, this is where constant names are given to the different Arduino pins.

After the initial variables are set, the Arduino begins the setup routine. In the setup function, we set initial conditions of variables when necessary and run any preliminary code that we want to run once. This is where serial communication is initiated, which is required for running the serial monitor.

From the setup function, we go to the loop routine. This is the main routine of the sketch. This is not only where the main code goes, but it will be executed over and over, so long as the sketch continues to run.

Below the loop routine, there are often other functions listed. These functions are user-defined and only activated when called in the setup and loop routine. When these functions are called, the Arduino processes all of the code in the function from top to bottom and then goes back to the next line in the sketch where it left off when the function was called. Functions are good because they allow us to run standard routines - over and over - without having to write the same lines of code over and over. We can simply call upon a function multiple times, and this will free up memory on the chip because the function routine is only written once. It also makes code easier to read.

All of that said, the only two parts of the sketch which are mandatory are the Setup and Loop routines. Code must be written in the Arduino Language, which is roughly based on C. Almost all

statements written in the Arduino language must end with a semicolon (;). Conditionals (such as if statements and for loops) do not need a semicolon (;). Conditionals have their own rules.

2.1.8 Hardware and Software Features

Typical -80dBm sensitivity Up to +4dBm RF transmit power Low Power 1.8V Operation ,1.8 to 3.6V I/O PIO control UART interface with programmable baud rate with integrated antenna with edge connector.

Default Baud rate: 38400, Data bits: 8, Stop bit: 1, Parity: No parity, Data control: has. Supported baud rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800. Given a rising pulse in PIO0, device will be disconnected. Status instruction port PIO1: low-disconnected, high-connected; PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s. Auto-connect to the last device on power as default. Permit pairing device to connect as default. Auto-pairing PINCODE: "0000" as default.

From the setup function, we go to the loop routine. This is the main routine of the sketch. This is not only where the main code goes, but it will be executed over and over, so long as the sketch continues to run.

Below the loop routine, there are often other functions listed. These functions are user-defined and only activated when called in the setup and loop routine. When these functions are called, the Arduino processes all of the code in the function from top to bottom and then goes back to the next line in the sketch where it left off when the function was called. Functions are good because they allow us to run standard routines - over and over - without having to write the same lines of code over and over. We can simply call upon a function multiple times, and this will free up memory on the chip because the function routine is only written once. It also makes code easier to read.

All of that said, the only two parts of the sketch which are mandatory are the Setup and Loop routines. Code must be written in the Arduino Language, which is roughly based on C. Almost all statements written in the Arduino language must end with a semicolon (;). Conditionals (such as if statements

and for loops) do not need a semicolon (;). Conditionals have their own rules and can be found under "Control Structures" on the Arduino Language page.

2.1.9 Plugging the Arduino into the Computer

Although the Arduino is plugged into the computer, it is not a true USB device. The board has a special chip that allows it to show up on the computer as a virtual serial port when it is plugged into a USB port. This is why it is important to plug the board in. When the board is not plugged in, the virtual serial port that the Arduino operates upon will not be present (since all of the information about it lives on the Arduino board).

Every single Arduino has a unique virtual serial port address. This means that every time we plug in a different Arduino board into the computer, we will need to reconfigure the serial port that is in use. Settings: Before we can start doing anything in the Arduino programmer, we must set board type and serial port.

2.1.10 Advantages of Microcontroller

- The simplest computer processor is used as the "brain" of the future system.
- Single purposes and low power consumption.
- Depending on the taste of the manufacturer, a bit of memory, a few A/D converters, timers, input/output lines etc. are added.
- Low cost and small packaging.
- Simple software able to control it all and which everyone can easily learn about has been developed.

2.2 Servo Motor

2.2.1 What is Servo Motor?

A servo motor is basically a DC motor (in some special cases it is AC motor) along with some other special purpose components that make a DC motor a servo. In a servo unit, we will find a small DC motor, a potentiometer, gear arrangement and an intelligent circuitry. The intelligent circuitry along with the potentiometer makes the servo to rotate according to our wishes. As we know, a small DC motor will rotate with high speed but the torque generated by its rotation will not be enough to move even a light load. This is where the gear system inside a servomechanism comes into the picture. The gear mechanism will take high input speed of the motor and at the output; we will get an output speed which is slower than original input speed but more practical and widely applicable.

Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. This output port of the potentiometer is connected with one of the input terminals of the error detector amplifier. Now an electrical signal is given to another input terminal of the error detector amplifier. Now difference between these two signals, one comes from potentiometer and another comes from external source, will be amplified in the error detector amplifier and feeds the DC motor. This amplified error signal acts as the input power of the DC motor and the motor starts rotating in desired direction. As the motor shaft progresses the potentiometer knob also rotates as it is coupled with motor shaft with help of gear arrangement. As the position of the potentiometer knob changes there will be an electrical signal produced at the potentiometer port. As the angular position of the potentiometer knob progresses the output or feedback signal increases. After desired angular position of motor shaft, the potentiometer knob is reaches at such position the electrical signal generated in the potentiometer becomes same as of external electrical signal given to amplifier. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer. As the input signal to the motor is nil at that position, the motor stops rotating. This is how a simple conceptual servo motor works.

Unlike dc motors, with servo motors we can position the motor shaft at a specific position (angle) using control signal. The motor shaft will hold at this position as long as the control signal not changed. This is very useful for controlling robot arms, unmanned airplanes control surface or any object that we want it to move at certain angle and stay at its new position. Servo motors may be classified according to size or torque that it can withstand into mini, standard and giant servos. Usually mini and standard size servo motors can be powered by Arduino directly with no need to external power supply or driver. Usually a servo motor comes with arms (metals or plastic) that are connected to the object required to move.

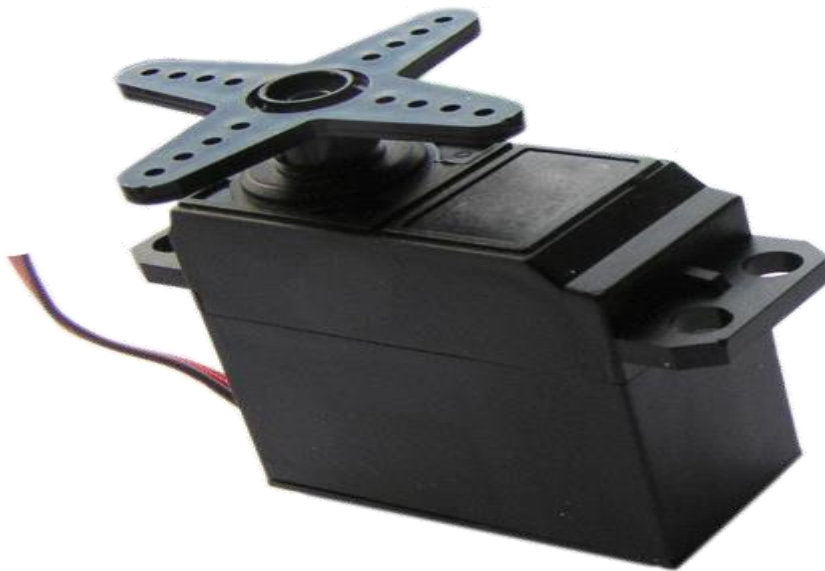


Figure 2.2 Servo Motor

2.2.2 How Does the Servo Motor Works?

A servo has 3 wires: Black wire: GND (ground), RED wire: +5v, Colored wire: control signal. The third pin accept the control signal which is a pulse-width modulation (PWM) signal. It can be easily produced by all micro- controllers and Arduino board. This accepts the signal from our controller that tells it what angle to turn to. The control signal is fairly simple compared to that of a stepper motor. It is just a pulse of varying lengths. The length of the pulse corresponds to the angle the motor turns to. For detailed and simplified explanation of Pulse Width Modulation

(PWM), Please click [here](#). The pulse width sent to servo ranges as follows: Minimum: 1 millisecond ---> Corresponds to 0 rotation angle. Maximum: 2 milliseconds ---> Corresponds to 180 rotation angle. Any length of pulse in between will rotate the servo shaft to its corresponding angle. For example, 1.5 ms pulse corresponds to rotation angle of 90 degrees.

Standard servo motor control using Arduino is extremely easy. This is because the Arduino software comes with a sample servo sketch and servo library that will get us up and running quickly

- Connect the black wire from the servo to the Gnd pin on the Arduino
- Connect the red wire from the servo to the +5V pin on the Arduino
- Connect the third wire (usually orange or yellow) from the servo to a digital pin on the Arduino.
- Future Electronics Egypt Ltd. (Arduino Egypt)

2.2.4 Important Notes

1- It is not a good idea to connect a motor of any kind directly to the Arduino because it usually requires more power than the board can provide.

2- In our example, the servo is being used to demonstrate code and is not encountering any resistance. Note that we should use a standard or small size if we are uncertain, check the servo's no load current rating (it should usually be under 150mA). 3-we may need an external source of 5 or 6 volts when connecting multiple servos. Four AA cells work well if we want to use battery power. Remember that we must connect the ground of the external power source to Arduino ground. Here is the example Sweep sketch distributed with Arduino:

```
#include <Servo.h> Servo myservo; // create servo object to control a servo int angle = 0;
```

```
// variable to store the servo position void setup()
```

```
Physical Output } myservo.attach(9); // attaches the servo on pin 10 to the servo object void loop()
```

```
{ } for(angle = 0; angle < 180; angle += 1) // goes from 0 degrees to 180 degrees {  
myservo.write(angle); delay(20); // in steps of 1 degree // tell servo to go to position in variable  
'angle' // waits 20ms between servo commands } for(angle = 180; angle >= 1; angle -= 1) // goes  
from 180 degrees to 0 degrees { } myservo.write(pos); delay(20); // tell servo to go to position in  
variable 'pos' // waits 20ms between servo commands.
```

2.3 Hc-05 Bluetooth Module

HC-05 Bluetooth Module is one of the most popular Bluetooth module used in embedded projects. It can be easily interfaced with Arduino Board, Raspberry Pi, and Microcontrollers through serial UART interface. HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm.

Specifications

- Frequency: 2.4GHz ISM Band
- Bluetooth Protocol: Bluetooth Specification v2.0 + EDR
- Emission Power: ≤4dBm, Class 2
- Modulation: GFSK (Gaussian Frequency Shift Keying)
- Operating Voltage: 3.3V
- Input Power Supply: 3.6V ~ 6V
- Security: Authentication & Encryption
- Working Temperature: -20°C ~ +75°C
- Dimensions: 35.7mm x 15.2mm x 5.6mm

2.3.1 Pins

On the bottom side there are labels for the signal direction and voltage levels:



Figure 2.3: Bluetooth Module HC-05

- **KEY:** according to the data sheet, I need to pull-up this pin while power-on-reset of the module to enforce AT mode. I have not been able to verify this yet. I have been told that some modules have this pin not connected at all?
- **VCC** is indicated in the range of 3.6V-6V. The module worked for me both with 3.3V and 5V
- **GND:** Ground
- **TXD:** serial *output* of the module, to be connected to RX of the microcontroller. Note that this signal is using 3.3V logic level
- **RXD:** serial *input* of the module, to be connected to the TX of the microcontroller. Note that this signal is using 3.3V logic levels.
- **STATE:** connected to LED2 (Pin32) of the module, but no meaning? At least on my module the pin was always low, regardless if paired or not.

2.3.2 Hardware Features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power

- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rat
- With integrated antenna

2.3.3 Software Features

- Default Baud rate: 38400, Data bits:8, Stop bit:1, Parity: No parity, Data control: has.
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:”0000” as default.
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

2.3.4 Schematic Diagram of HC-05 Module

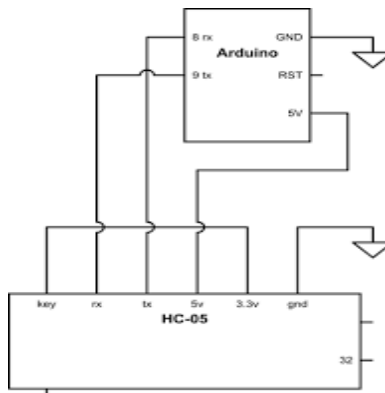


Figure 2.4: Schematic Diagram of HC-05

2.4 How does Bluetooth Work?

Bluetooth sends and receives radio waves in a band of 79 different frequencies (channels) centered on 2.45 GHz, set apart from radio, television, and cellphones, and reserved for use by industrial, scientific, and medical gadgets. Bluetooth's short-range transmitters are one of its biggest plus points. They use virtually no power and, because they don't travel far, are theoretically more secure than wireless networks that operate over longer ranges, such as Wi-Fi.

Bluetooth devices automatically detect and connect to one another and up to eight of them can communicate at any one time. They don't interfere with one another because each pair of devices uses a different one of the 79 available channels. If two devices want to talk, they pick a channel randomly and, if that's already taken, randomly switch to one of the others (a technique known as spread-spectrum frequency hopping). To minimize the risks of interference from other electrical appliances (and also to improve security), pairs of devices constantly shift the frequency they're using—thousands of times a second.

When a group of two or more Bluetooth devices are sharing information together, they form a kind of ad-hoc, mini computer network called a piconet. Other devices can join or leave an existing piconet at any time. One device (known as the master) acts as the overall controller of the network, while the others (known as slaves) obey its instructions. Two or more separate piconets can also join up and share information forming what's called a scatternet.

2.5 Overview of Existing System

The existing system largely consists of physical lock and keys. The problem with existing system is that the physical key is easy to manipulate also for multiple doors one needs to carry multiple keys. This increases the hassle of carrying multiple keys as well as the possibility of the keys being misplaced. Physical key also limits the number of people who can access the particular lock.

2.6 Existing Door Lock System

The main idea of designing digital locking system is to provide many modern security features than mechanical lock. The system has a Bluetooth Receiver by which the password can be received. This password should be entered through an application installed on Android mobile. When entered password matches with the password stored in the memory then the DC motor rotated clockwise so that the door is opened then it will be paused for some time and again rotated anti-clockwise so that door is closed. If we enter a wrong password, then appropriate message is displayed on the LCD. Applications are entrance doors, car doors, bank lockers etc. Many times people misplace their keys or accidentally lock their door. In these cases it is really difficult to enter the house. This project is designed to overcome this problem. Main concept behind this project is to open a door using a password entered through an Android application. As well as turning on the Buzzer when password is entered wrong for multiple times. To use the proposed system user needs to install a Bluetooth application and then perform the following steps:

- Open the installed application
- Connect the mobile device to the system via Bluetooth.
- Now pair the system with the mobile device using default pairing code.
- Enter the password and if the password is correct access is granted else access will be denied by the system.

2.7 Description of Blocks

The major blocks used in the said system are as follows:

- Bluetooth Decoder: The Bluetooth decoder is used for transmitting as well as receiving the serial data. The range of Bluetooth decoder depends upon its class. The Bluetooth decoder used in the proposed system has a range of 10 meters.

- DC Motor Driver This IC is used to convert 5v to 12v which is required to drive the DC motor, which is used to open the gate. The L293D is a quadruple half Bridge bidirectional motor driver IC that can drive current of up to 600mA with voltage range of 4.5 to 36 volts. It is suitable to drive small DC-Geared motors, bipolar stepper motor etc.
- DC Motor A DC motor is a mechanically commutated electric motor powered from direct current (DC). In DC motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. Opposite (North and South) polarities of magnet attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion. The DC motor used in the proposed system is of approximately 100 rpm.
- Relay acts as a switch which is used to control the 230-volt AC supply. This relay will be turned off if there is no person inside the room. This relay can be used to turn off the electrical appliances like fan, tubes etc.
- EEPROM Memory IC EEPROM stands for electrically erasable programmable read only memory. It is a secondary storage device that once written (programmed) can hold data even when the power is switched off. The EEPROM is a read only memory that can be electrically erased and reprogrammed. We have used EEPROM to store the password of our system.
- The power supply gives +5v and +12v supply to the circuit. The power supply consists of four stages namely transformer, rectifier, filter, and regulator. Transformer is a step-down transformer taking input of 230v AC and giving output of 15v at the secondary. This 15v AC is rectified by bridge rectifier consisting of four diodes, which converts the AC wave into fully rectified wave. The next stage is the filter stage consisting of capacitor, which converts the fully rectified wave into the DC wave with some ripple. Last stage is the regulator stage. Regulator removes the entire ripple and gives pure DC.

2.8 A Features of the Proposed System

The important features of the system are as follows: • Eliminates the hassle of carrying multiple keys.

- Provides higher security compared to traditional lock and key.
- As Bluetooth is used for data transfer, the range can be varied as required.
- The application for accessing the lock can be installed on any android device.
- Buzzer system helps to recognize wrong password.
- Password can be changed multiple times for better security.

Chapter 03

3.1 Design and Development of the System

The design and development of the system is given below: -

3.1.1 Block Diagram of Control System

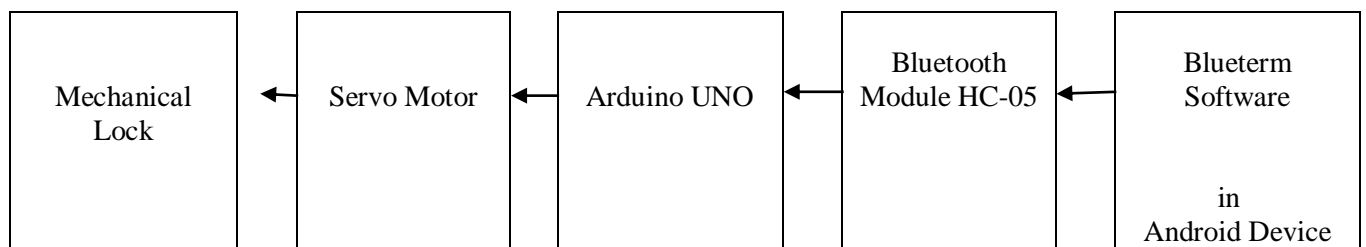


Figure 3.1 Block Diagram of the Control System

3.2 Description of Each Block

Mainly we use five major components for designing a full control system. The Bluetooth device receiving signal from android mobile phone. The Bluetooth device and the Servo-motor are connected to Arduino UNO board. The Arduino board is connected with power control circuit. The output will show by mechanical lock by locking or unlocking the door.

3.2.1 Equipment Used in this System

To design the project, we have used the following component:

A. Mechanical Lock with Rack and Pinion

Mechanical Lock is made with Servo motor, rack and pinion etc. The Servo motor is adjusted with rack and pinion. Here, to minimize the cost both the rack and pinion have been made by us. The

pinion has been made by cutting a large plastic gear wheel and the rack is made by an aluminum bar. Teeth of same size with the pinion have been made by a triangular file. The developed system is shown in the following figure.

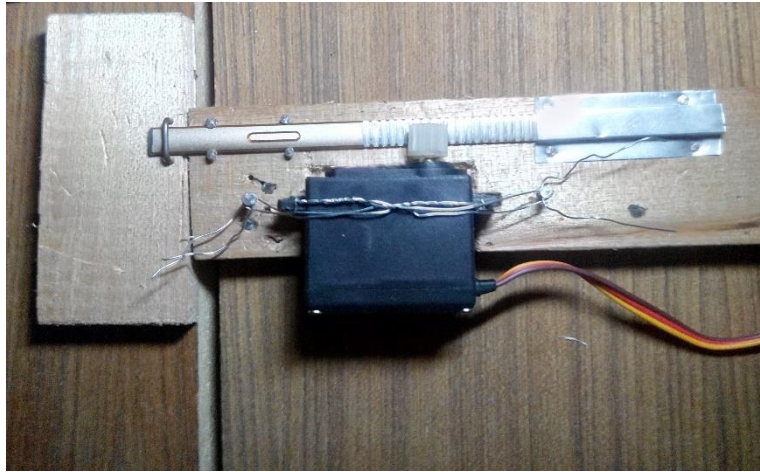


Figure (a) 3.2: Mechanical Lock (Locked)

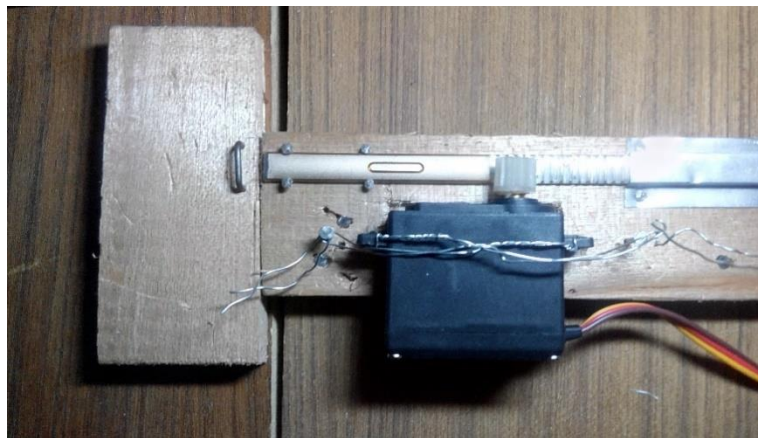


Figure (b) 3.2: Mechanical Lock (Unlocked)

B. Arduino Uno Board

The following Arduino UNO board (fig. 3.3) is used in our project.

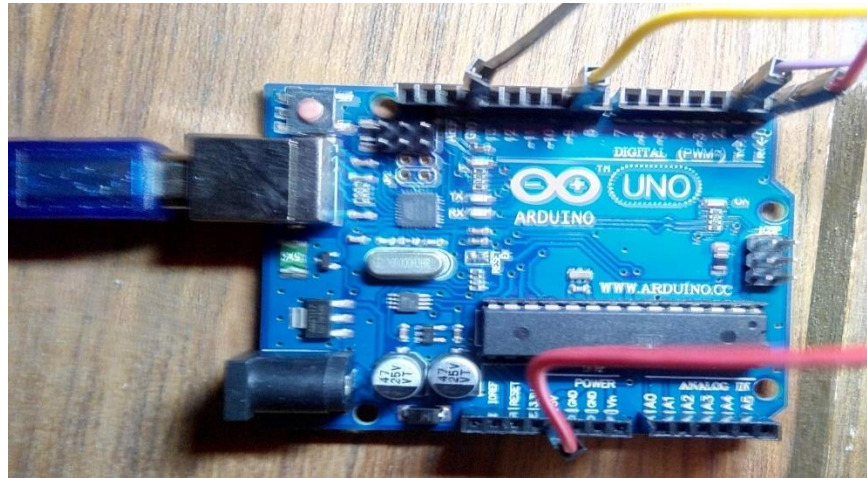


Figure 3.3: Arduino UNO

C. Bread Board

The following bread board (fig. 3.4) is used in our project.

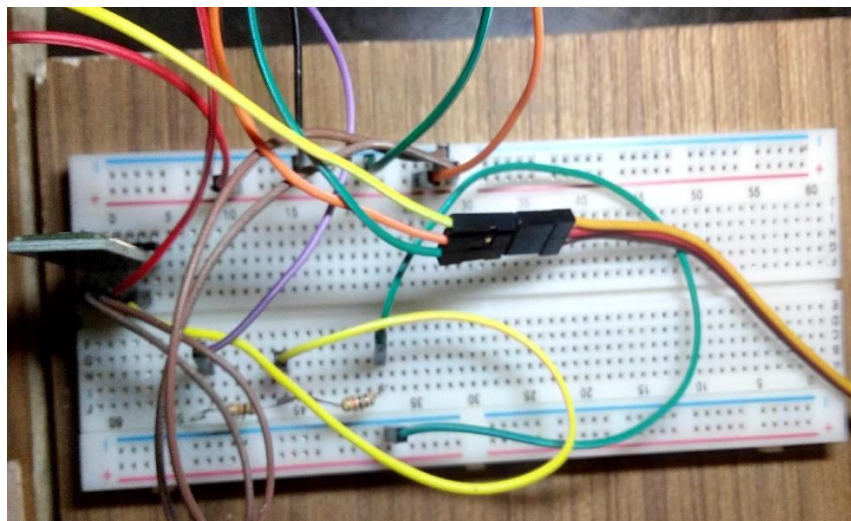


Figure 3.4: Bread Board

D. HC-05 Bluetooth Module

The following Bluetooth module (fig. 3.5) has been used in our project.

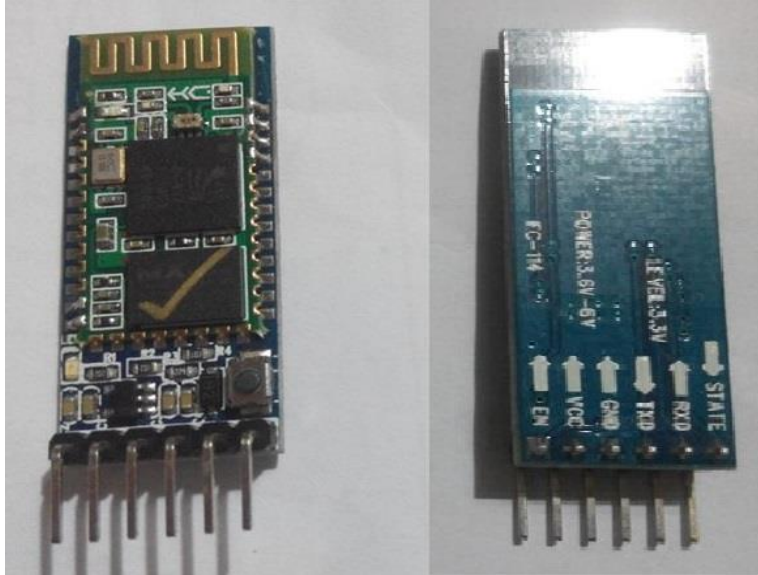


Figure 3.5: HC-05 Bluetooth Module

E. Resistances

The following 10k resistances (fig. 3.6) have been used in our project.



Figure 3.6: Resistances

F. Some Wires

The following wires (fig. 3.7) have been used in our project.

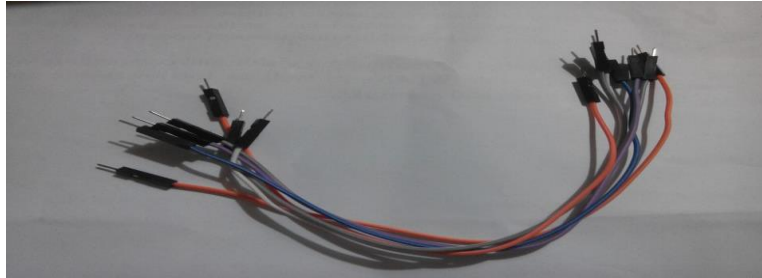


Figure 3.7: Jumper Wire

3.2.2 Connection of Arduino with Bluetooth Device

In the 3.2 figure, it has shown the connection of Arduino UNO and Bluetooth Module HC-05. Here we developed a connection of Bluetooth device with mobile phone. First, we think about android mobile software which will be appropriate for this project. We install Arduino Bluetooth Controller software on our mobile phone. Then we connect this software with HC-05 Bluetooth module. After connecting software, we developed the connection between Arduino and Bluetooth device. We run a program on Arduino board for implementing the project.

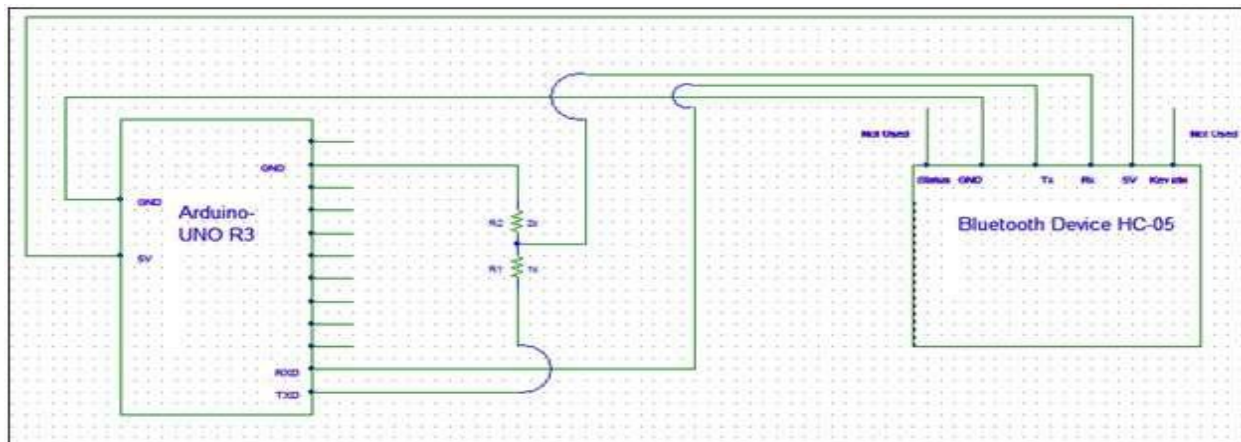


Figure 3.2 Circuit diagram of the Arduino with Bluetooth Device

3.2.3 Connection of Arduino with Servo Motor

In the 3.3 figure, it has shown the connection of Arduino UNO and servo motor. By the command in the Blueterm software on android device, the servo motor will be rotate right or left, and the door will be locked or unlocked.

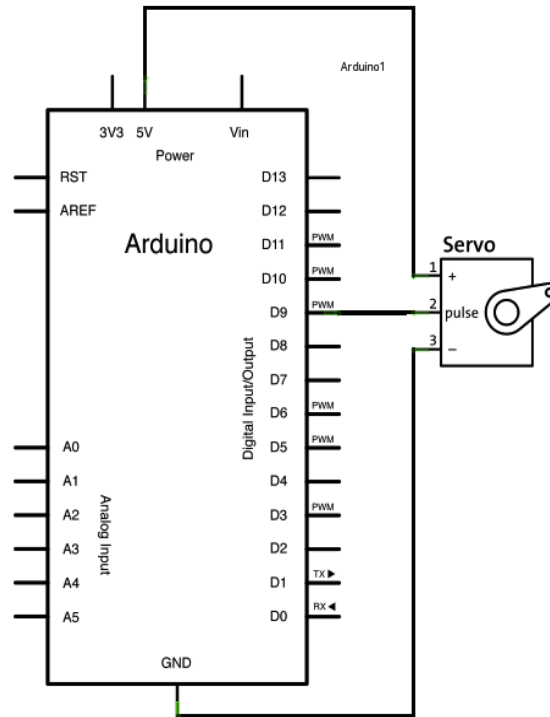


Figure 3.3 Circuit diagram of the Arduino with Servo-motor

3.3 Development of Software

Software design is divided into two parts. At first we write the Arduino program in Arduino software in pc or laptop. Then we compile it to the Arduino hardware. This Arduino command is control the Arduino hardware and other circuit connection. For making connection between Arduino and android mobile we need to install Blueterm software on our mobile phone.

3.3.1 Installing Arduino

Arduino runs on Windows. We have to go to the Arduino software web-site at <http://arduino.cc/en/Main/Software> and download the version of the software compatible with our system. We use Arduino 1.6.6 version.

3.3.2 Installing Blueterm

To connect the android device to the Bluetooth Module HC-05 for sending the command, we need to install Blueterm software in our android device. For installing Blueterm, we have to go to playstore and from there we install this android software.

3.3.3 Verifying the Hardware

Now that we have the Arduino IDE software installed, let's connect the computer to the Arduino board, load a small program, and verify that all components are working together. First, need to connect the USB cable to our Arduino board and then plug the other end of the USB cable into our computer.

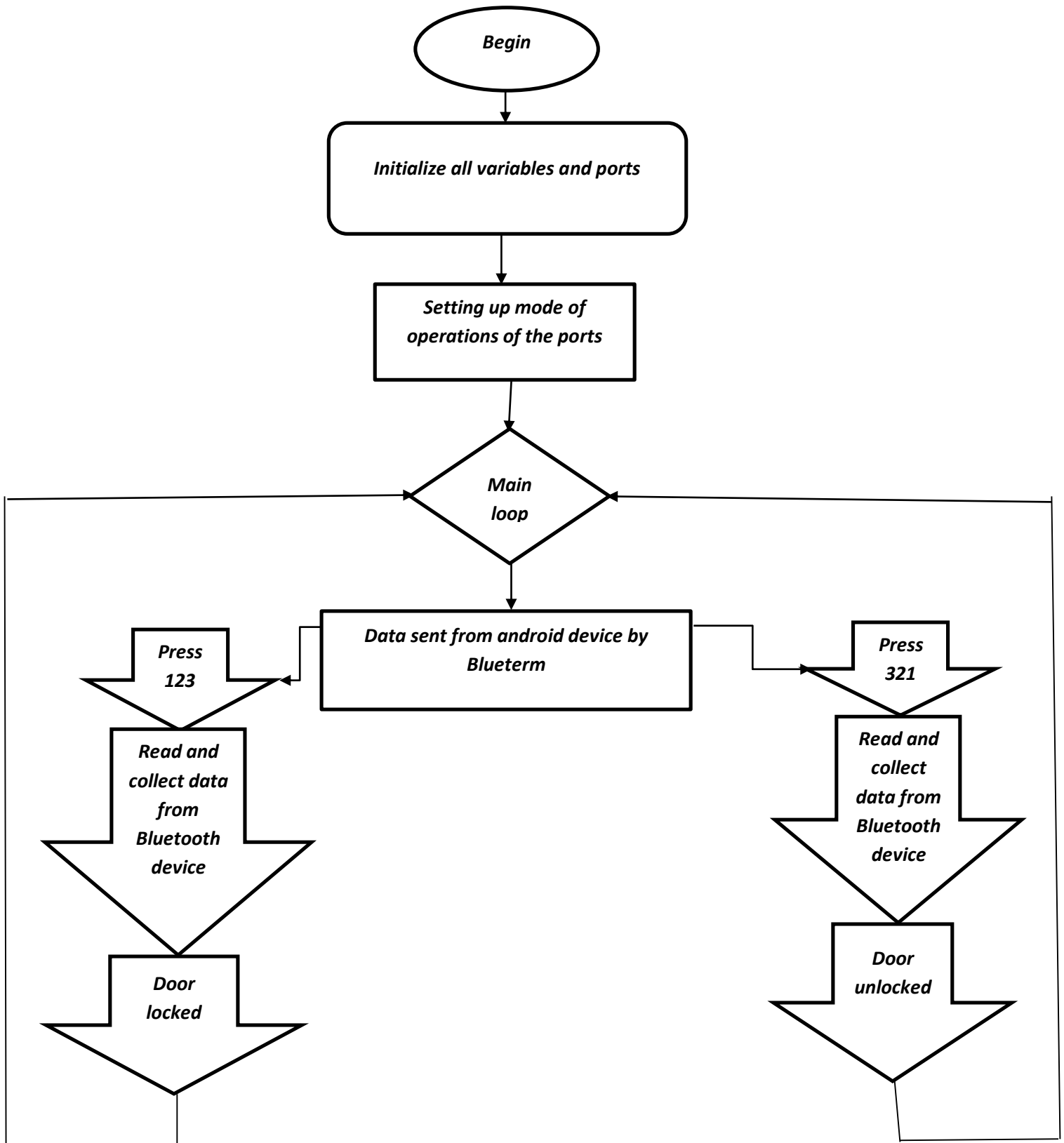
3.3.4 Arduino Language

The Arduino language is implemented in C/C++ and based in Wiring. When we write an Arduino sketch, we are implicitly making use of the Wiring library, which is included with the Arduino IDE. This allows us to make run able programs by using only two functions: setup () and loop (). As mentioned, the Wiring language is inspired by Processing, and the Arduino language structure is inherited from the Processing language, where the equivalent functions are called setup (). We need to include both functions in every Arduino program, even if we don't need one of them. Let's analyze the structure of a simple Arduino sketch using again the Blink example.

3.3.5 Logical Diagram

We have a logical diagram for this project. Here android device works as an input signal which start from beginning part. We have got output from the mechanical lock at the door as the door is locked or unlocked if the password is matched.

3.3.6 Flow Chart



At first, we have to initialize all the variables and ports. Then we have to setting up mode of operations of the ports. Then in the main loop we have to decide, we will lock or unlock the door. If we press 123 the door will be locked after read and collect data from Bluetooth device. Because this is the password for lock the door. On the other hand, if we want to unlock the door, we need to press 321 in Blueterm. Because this is the password for unlock the door.

3.3.7 Programming Codes

The overall codes of our project is given below

```
#include <Servo.h>
Servo myservo; // create servo object to control a servo
int j,i=0,sd;
char key[10],state='2';
bool flag =0; //flag = 0 means the door is unlocked and flag = 1 means door is locked
void setup() {
    // sets the pins as outputs:

    myservo.attach(9);
    myservo.write(100);
    delay(10);
    Serial.begin(9600);
    flag = 0;
}
void loop() {
    //Serial.println(flag);
    //Serial.println(state);
    if(Serial.available() > 0){
        sd = Serial.read();

        key[i]=sd;
        i=(i+1)%3;
        Serial.println(i);
        if(sd=='9') i=0;
        //flag=0;
    }
}
```

```

if(key[0]=='1' & key[1]=='2' & key[2]=='3')
{
    state = '1';
    for(j=0;j<10;j++) key[j]='0';
}
if(key[0]=='3' & key[1]=='2' & key[2]=='1')
{
    state = '0';
    for(j=0;j<10;j++) key[j]='0';
}

if (state == '1') {
    if(flag==1)
    {
        Serial.println("Door: Already Locked");
        i=0;
    }
    else {
        myservo.write(110);
        delay(1000);
        Serial.println("Door: Locked");
        flag=1;
        i=0;
    }
    myservo.write(100);
    state='2';
    delay(10);
}

else if (state == '0') {
    if(flag==0)
    {
        Serial.println("Door: Already Unlocked");
        i=0;
    }
    else {
        myservo.write(90);
        delay(1125);
        Serial.println("Door: Unlocked");
        flag=0;
        i=0;
    }
    myservo.write(100);
    state='2';
    delay(10);
}
//delay(5000);
}

```

Figure 3.4: The overall codes used in this project

Chapter 04

4.1 Implementation

All the parts are connected as circuit design. Then we upload the programming code in the Arduino and we get positive result. It works properly according to our design.

4.2 Results

When we Press the password '123' at Blueterm software, which is install in any smartphone the door will be locked.

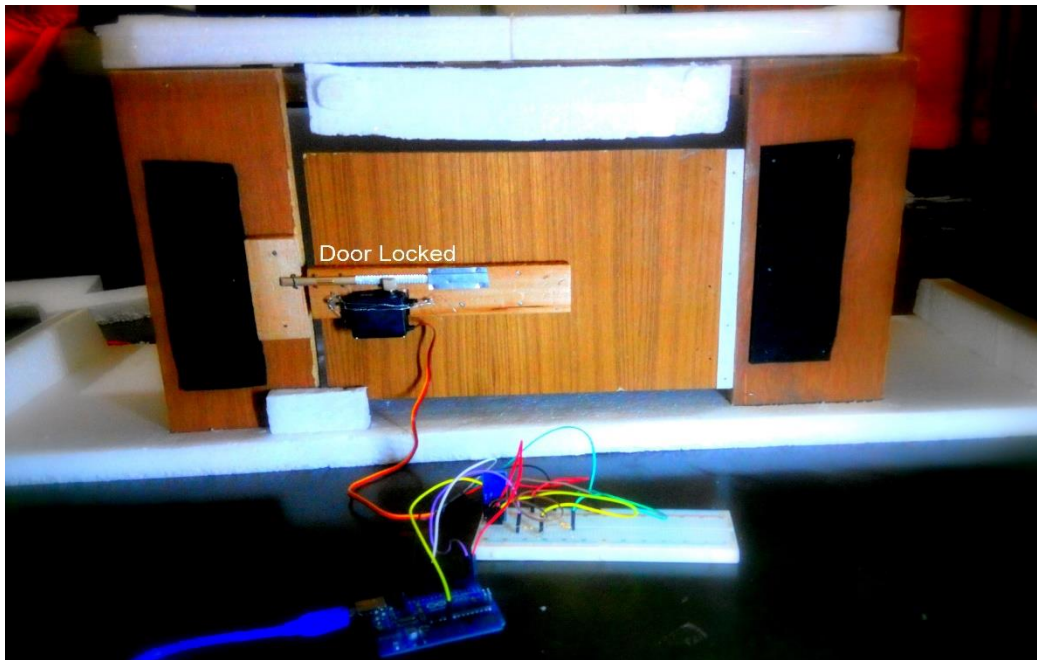


Figure 4.1 Processing of Locke the Door

And when we Press the password '321' at Blueterm software, which is install in any smartphone the door will be unlocked.

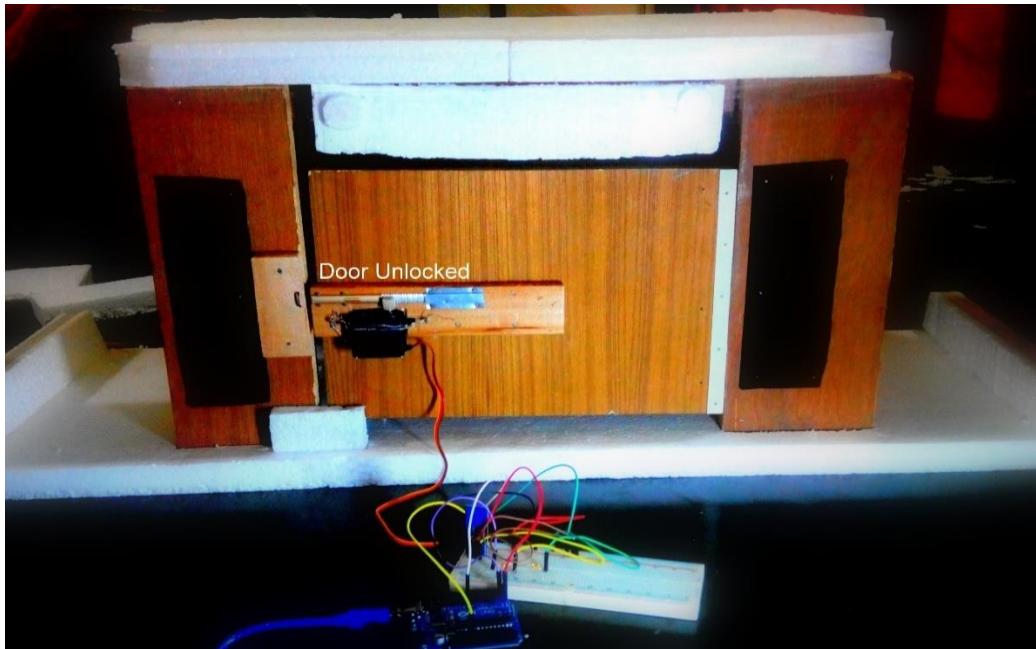


Figure 4.2 Processing of Unlock the Door

If the door is already locked and we press the password 123 to lock the, it is shown that the door is already locked; on the other hand, if the door is already unlocked and we press the password 321 to unlock the, it is shown that the door is already unlocked in the both software Arduino in the pc/laptop and Blueterm in the android device.

```
1
2
0
Door: Locked
1
2
0
Door: Unlocked
1
1
2
0
Door: Locked
1
1
2
0
Door: Already Locked
1
1
2
0
Door: Unlocked
1
1
2
0
Door: Already Unlocked
```

Figure 4.3 Outputs in Arduino when the door is already locked or unlocked

Chapter 5

5.1 Conclusion:

The door control system using smart-mobile has designed and developed for making our life easier and secured for entrance into houses or offices. To complete the entire project properly, we have worked seriously and to implement the project we have used the essential equipment. The equipment that are used to develop the whole project are Arduino Uno board, Bluetooth module, bread board, jumper wires, servo motor, rack and pinion and different software to run the equipment. By developing the individual units step by step the whole system has been assembled together. The entire project operation procedure has been shown by drawing the flow-chart. In this project paper, we have attached different figures of our project works that shows how serially we prepared the entire system. Finally, we have successfully completed our project works which is suitable for door control system using smart-mobile phone.

5.2 Future Work Scope:

In future, this project gives us an opportunity to do a big project. For the limitation of time and required fund we were able to develop just “DESIGN AND DEVELOPMENT OF A MOBILE CONTROLLED DOOR LOCK SYSTEM”. The system will work using Android Mobile Phone. It also works on Tab or Laptop. Day by day it’s need to develop the security of us. So, we have a big work scope in this sector. We hope that, we will be able to complete all the features needed for its ultimate applications.

REFERENCES

- [1] R.P.Jain, Digital Electronics, Tata McGraw Hill.
- [2] Ramakant Gayakwad, Op-Amps and Linear Integrated Circuits.
- [3] Robert L. Boylestad, Louis Nashelsky, Electronic Devices and Circuit Theory.
- [4] Kenneth J. Ayala, The 8051 Microcontroller Architecture, Programming and Applications.
- [5] <https://www.circuitstoday.com>
- [6] <https://www.discovercircuits.com>
- [7] <https://www.electronicsforu.com>
- [8] <https://www.electronic-diagrams.com>
- [9] <https://www.circuitlake.com>
- [10] <https://www.wikipedia.org>