A Report on

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO AND**

**HC-SR04 ULTRASONIC DISTANCE SENSOR USING RPI PICO**

for

**Mini Project 1-b (REV- 2019 ‘C’ Scheme) of Second Year, (SE Sem-IV)**

in

**Electronics & Telecommunication Engineering**

by

1. Mohd Aman (28)
2. Purva Shete (34)
3. Shreyash Tiwari (43)
4. Riddhesh Veling (45)

Under the guidance of

**Ms. Savita Kulkarni**

**St. Francis Institute of Technology**



**UNIVERSITY OF MUMBAI**

**AY 2022-2023**

**CERTIFICATE**

This is to certify that the project entitled **Password Based Smart Locking System Using Arduino and HC-SR04 ultrasonic distance sensor using Rpi Pico** is a bonafide work of

1. Mohd Aman
2. Purva Shete
3. Shreyash Tiwari
4. Riddhesh Veling

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Mini Project 1-B (REV- 2019 ‘C’ Scheme) of Second Year, (SE Sem-IV)** in **Electronics & Telecommunication Engineering** as laid down by **University of Mumbai** during academic year **2022-23**

(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

**Examiner/Reviewer-1 Examiner/ Reviewer -2**

**Name of Guide**

**Guide Head of Department**

**INDEX**

|  |
| --- |
| **1. INTRODUCTION**  1.1 Need……………………………………………………………………………… 5  1.2 Definition………………………………………………………………………… 6  **2. MINIPROJECT DESIGN (PRINCIPLE AND WORKING)**  2.1 Block Diagram……………………………………………………………….. 7  2.2 Block Diagram Description…………………………………………………... 8  2.3 Circuit diagram and working…………………………………………………. 9  **3. COMPONENTS/TOOLS TO BE USED**  3.1 Components (with features related to project)………………………………... 11  3.2 Software tools Description……………………………………………………. 17  **4. MINIPROJECT IMPLEMENTATION STEPS USING ARDUINO**  4.1 Implemented Steps (Snapshots/photos…)…………………………………….. 20  4.2 Simulation Results……...……………………………………………………… 23  4.3 Results & discussion…………………………………………………………… 24  **5. MINIPROJECT IMPLEMENTATION STEPS USING RASPBERRIY PI**  5.1 Implemented Steps (Snapshots/photos)………………………………………... 25  5.2 Simulation Results if any………………………………………………………. 27  5.3 Results & discussion…………………………………………………………… 28  **References**……………………………………………………………………... 29 |

**LIST OF FIGURES**

[Fig 2. 1. 1 Block diagram for Smart Lock using Arduino UNO 7](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477537)

[Fig 2. 1. 2 Block diagram for Ultrasonic Sensor using Rpi Pico 7](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477538)

[Fig 2. 3. 1 Circuit diagram for Smart Lock using Arduino UNO 9](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477554)

[Fig 2. 3. 2 Circuit diagram for Ultrasonic Sensor using Rpi Pico 10](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477555)

[Fig 3. 1. 1 Arduino UNO 11](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477682)

[Fig 3. 1. 2 Bluetooth HC-05 Module 11](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477683)

[Fig 3. 1. 3 Servo Motor 12](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477684)

[Fig 3. 1. 4 Android Mobile Phone 12](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477685)

[Fig 3. 1. 5 Jumper Wires 13](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477686)

[Fig 3. 1. 6 Power Bank 13](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477687)

[Fig 3. 1. 7 Raspberry Pi Pico 14](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477688)

[Fig 3. 1. 8 SSD1306 OLED Display 14](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477689)

[Fig 3. 1. 9 HC-SR04 Ultrasonic Sensor 15](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477690)

[Fig 3. 1. 10 Breadboard 15](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477691)

[Fig 3. 1. 11 Jumper Wires 16](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477692)

[Fig 3. 1. 12 Micro USB Cable 16](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477693)

[Fig 3. 2. 1 Arduino IDE 17](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477710)

[Fig 3. 2. 2 Thonny IDE 18](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477711)

[Fig 3. 2. 3 MIT App Inventor 19](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477712)

[Fig 4. 1. 1 Circuit Connection 20](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477720)

[Fig 4. 1. 2 Control Page UI 20](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477721)

[Fig 4. 1. 3 App Login UI 20](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477722)

[Fig 4. 1. 4 Login Page Code 21](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477723)

[Fig 4. 1. 5 Control Page Code 21](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477724)

[Fig 4. 1. 6 Arduino code 22](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477725)

[Fig 4. 2. 1 When Lock button is pressed on mobile application 23](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477762)

[Fig 4. 2. 2 When Unlock button is pressed on mobile application 23](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477763)

[Fig 5. 1. 1 Circuit Connection 25](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477769)

[Fig 5. 1. 2 Raspberry Pi code 25](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477770)

[Fig 5. 1. 3 Raspberry Pi Code 26](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477771)

[Fig 5. 2. 1 Object at distance of 10cm 27](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477777)

[Fig 5. 2. 2 Object at distance of 16 cm 27](https://d.docs.live.net/8d200e223842800f/Desktop/Mini%20Project/B7%20Mini%20Project%20Report.docx#_Toc131477778)

**1. INTRODUCTION**

**1.1 NEED:**

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO**

If you are still using the key-based system. The basic problem with it is that every time we need to carry a key to unlock the door and you are likely to land in a big problem if your key gets lost or stolen. If it is lost, then we must break the locking system which also result in wastage of money. For old age people it is difficult to open key based lock and there can only be a unique key for single lock and for different lock you have different keys. Furthermore, carrying many keys is also a burden. The electronic wireless lock system is not safe either as there is a high risk being hacked.

For your safety and security, we bring to you a DIY smart lock that has the capability to remove all these security threats and problems.

**ULTRASONIC DISTANCE SENSOR USING RASPBERRY PI PICO**

The need for ultrasonic distance sensors in daily life is evident from the various applications that use them. They provide an accurate and reliable way to measure distances and detect objects, making them an essential component in many industries. Ultrasonic distance sensors are also relatively inexpensive and easy to use, making them accessible to a wide range of users.

In summary, ultrasonic distance sensors have become an integral part of our daily lives, improving safety and efficiency in various applications.

**1.2 DEFINITION**

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO**

In this project, we have made an application consisting of a password which when entered correctly connects to the Bluetooth module (HC-05), this module is also password locked, if connected to a device will allow the user to lock or unlock the door with the help of a servo motor. An Arduino Atmega board is used to connect the servo and Bluetooth module. The application is made using the MIT App Inventor.

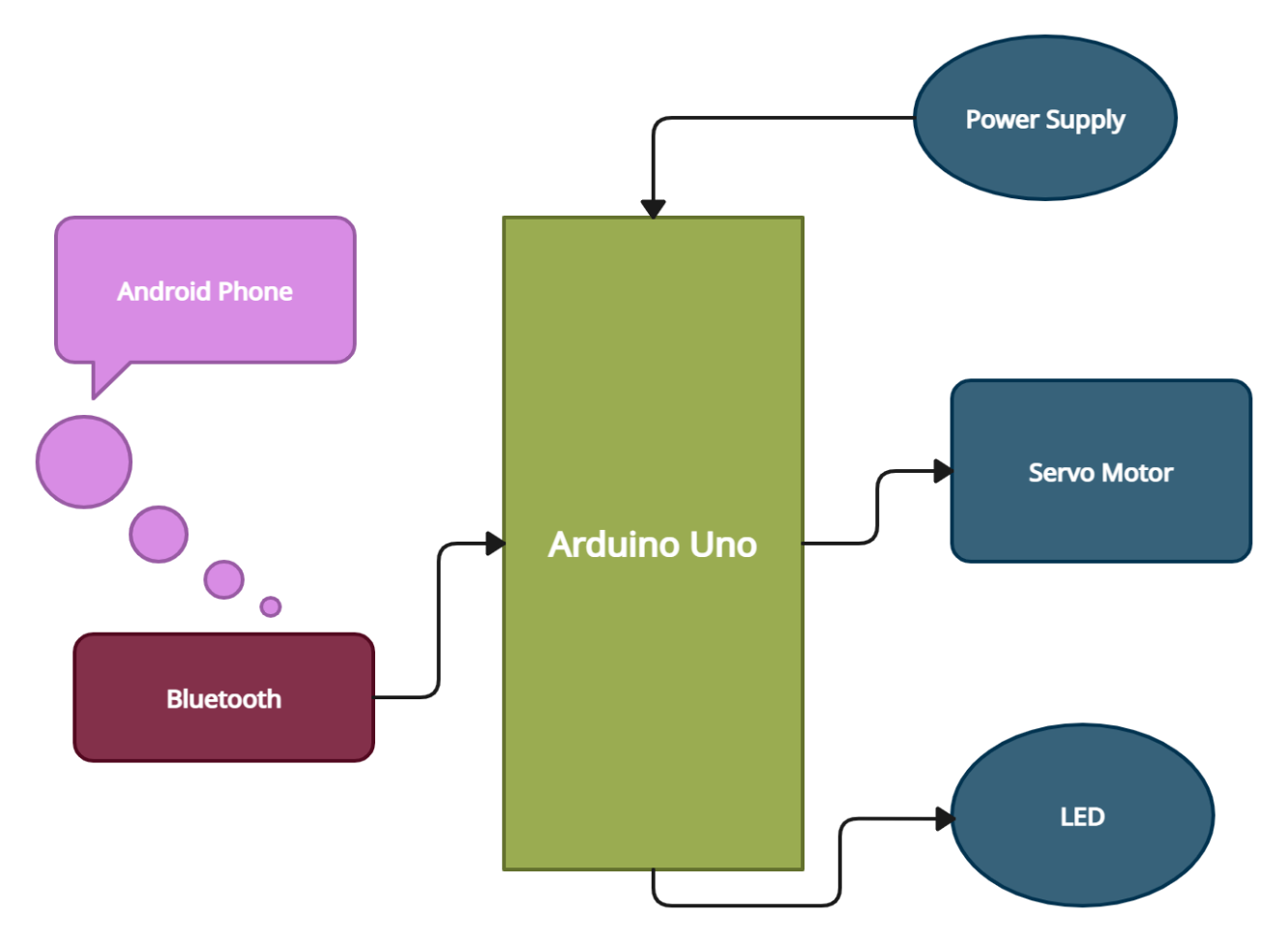
**ULTRASONIC DISTANCE SENSOR USING RASPBERRY PI PICO**

This project involves an ultrasonic distance sensor using Raspberry Pi Pico that uses ultrasonic waves to measure the distance between the sensor and an object. The Raspberry Pi Pico, a low-cost microcontroller board, is used to interface with the ultrasonic sensor and process the distance measurement data. The system can be used in a wide range of applications such as obstacle detection, liquid level measurement, and parking sensors. The main goal of the project is to provide accurate and reliable distance sensing capabilities using a cost-effective and easy-to-implement solution.

**2. MINI PROJECT DESIGN (PRINCIPLE AND WORKING)**

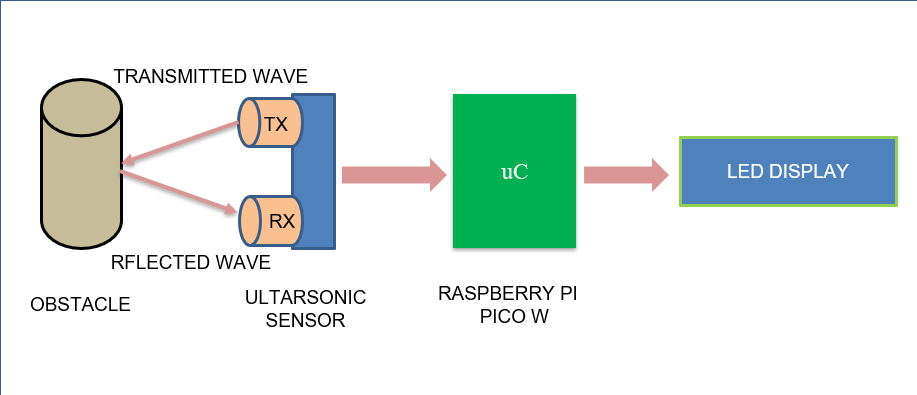
**2.1** **BLOCK DIAGRAM:**

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO**

****

**Fig 2. 1. 1 Block diagram for Smart Lock using Arduino UNO**

**ULTRASONIC DISTANCE SENSOR USING RASPBERRY PI PICO**



**Fig 2. 1. 2 Block diagram for Ultrasonic Sensor using Rpi Pico**

**2.2 BLOCK DIAGRAM DESCRIPTION:**

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO**

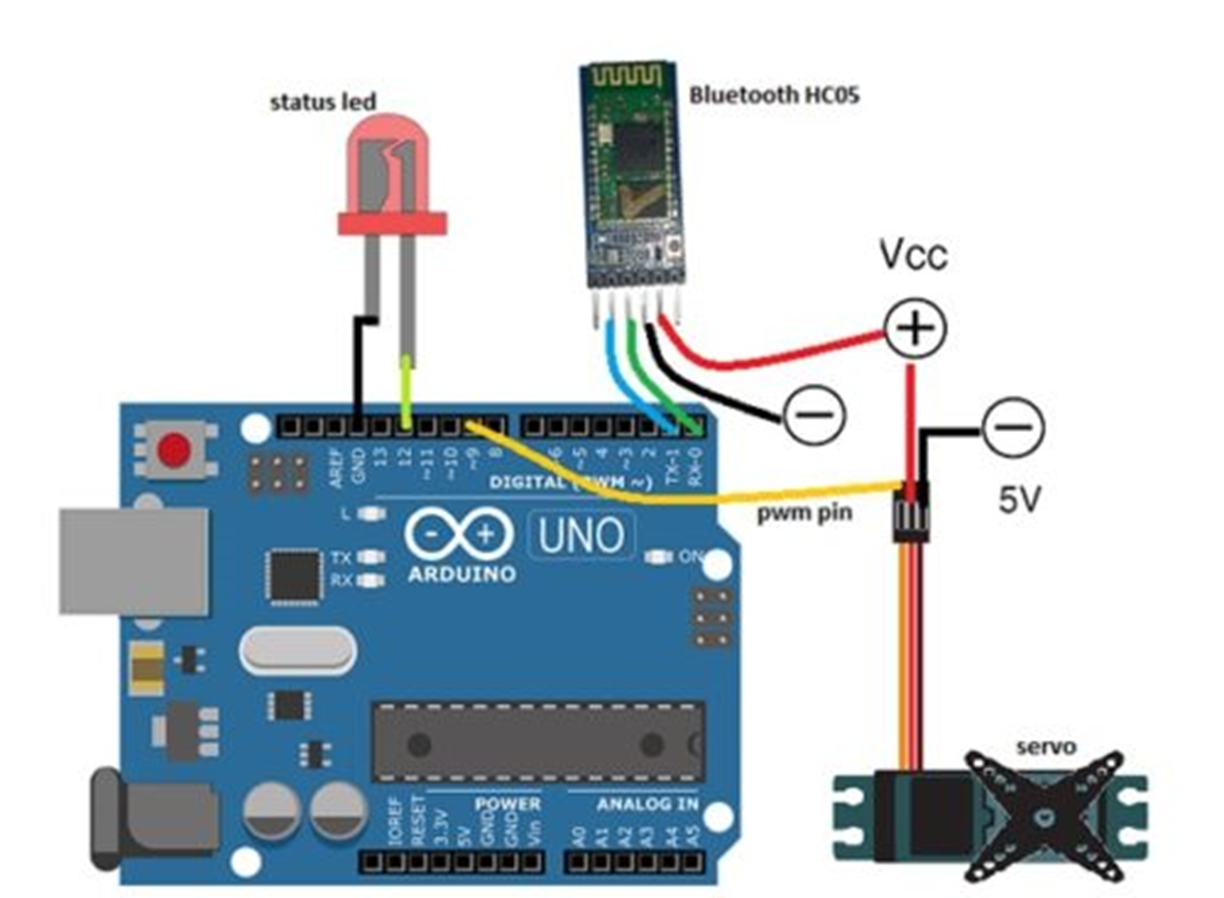
* The Android Phone’s Bluetooth signal is detected by the Bluetooth (HC-05).
* Bluetooth (HC-05) sends signal to Arduino.
* Arduino passes signal to servo motor to rotate it accordingly.

**ULTRASONIC DISTANCE SENSOR USING RASPBERRY PI PICO**

* The HC-SR04 ultrasonic sensor sends the ultrasonic waves towards an object/obstacle.
* The reflected waves are received by the HC-SR04 sensor and signal is sent to Raspberry Pi Pico.
* The Raspberry Pi Pico module then sends the required data to the SSD1306 OLED Display.
* The SSD1306 OLED Display then displays the distance in centimeters.

**2.3 CIRCUIT DIAGRAM AND WORKING:**

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO**



**Fig 2. 3. 1 Circuit diagram for Smart Lock using Arduino UNO**

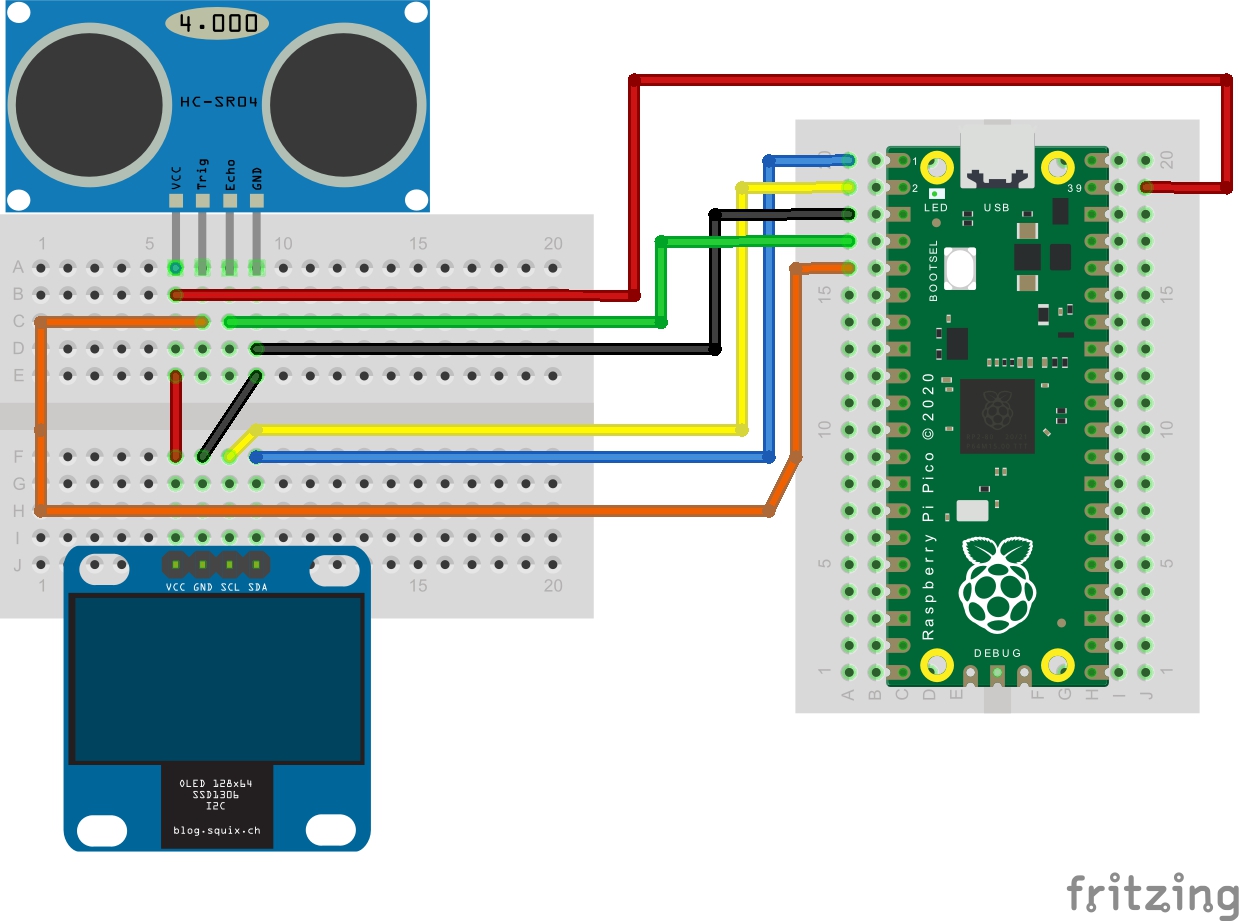
**WORKING:**

Once the user enters correct password and logs in to the app, he can access the control screen where there are 3 buttons. The ‘Bluetooth Button’ is used to scan the nearby Bluetooth devices and connect to them. The user needs to select HC-05 from the list. When successfully paired, the status is shown on the screen and the blinking of HC-05 module also stops.

The ‘Unlock Button’ and ‘Lock Button’ is used to control the door lock (servo motor). When Unlock button is pressed, the letter ‘U’ goes to the Arduino as an indicator that the servo must be rotated by 120° in clockwise direction and unlock the door lock. Similarly, when Lock button is pressed, the letter ‘L’ goes to the Arduino as an indicator that the servo must be rotated by 120° in anti-clockwise direction to return to its original position and lock the door lock.

Also, the LED shows the lock and unlock status. When the LED is on, the door is locked and when off, the door is unlocked.

**ULTRASONIC DISTANCE SENSOR USING RASPBERRY PI PICO**

****

**Fig 2. 3. 2 Circuit diagram for Ultrasonic Sensor using Rpi Pico**

**WORKING:**

The program starts by initializing the GPIO pins of the Raspberry Pi Pico and the HC-SR04 sensor.

The program then sends a trigger signal to the HC-SR04 sensor by setting the trigger pin to LOW for a brief period, then back to HIGH.

The HC-SR04 sensor emits an ultrasonic wave signal that bounces off the nearest object and returns to the sensor. The time taken for the signal to travel from the sensor to the object and back is measured.

The program reads the time taken for the signal to travel and calculates the distance based on the speed of sound and the time taken.

The distance is displayed on the SSD1306 OLED display in centimeters.

The program continues to run in a loop, continuously measuring and displaying the distance.

**3. COMPONENTS/TOOLS TO BE USED**

**3.1** **COMPONENTS:**

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO**

**1) Arduino UNO:** Arduino UNO is a microcontroller board based on the ATmega328P.



**Fig 3. 1. 1 Arduino UNO**

**2)** **Bluetooth HC-05:** The Bluetooth HC05 module is a small electronic device that allows for wireless communication between devices over short distances. The HC05 module supports Bluetooth 2.0.



**Fig 3. 1. 2 Bluetooth HC-05 Module**

**3)** **Servo Motor:** A servo motor is a type of motor that is designed to be highly precise and controllable. It uses feedback control to maintain a specific position, speed, or torque. The motor consists of a rotor, a stator, and a control circuit. The control circuit measures the motor's position and adjusts the voltage and current supplied to the motor to keep it in the desired position.



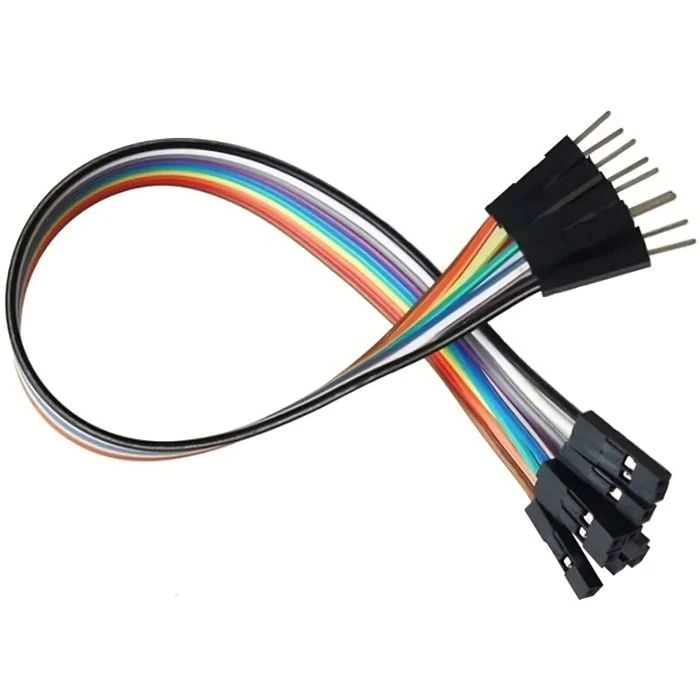
**Fig 3. 1. 3 Servo Motor**

**4)** **Android Device:** Used as an ‘User Interface’



**Fig 3. 1. 4 Android Mobile Phone**

**5)** **Jumper Wires:** Jumper wires are simple electrical cables with connector pins or clips at both ends, used to connect electronic components on a breadboard or circuit board. They are usually made of flexible, stranded copper wire with insulation, and come in a variety of colors, lengths, and gauges.



**Fig 3. 1. 5 Jumper Wires**

**6)** **Power Bank (Source):** A power bank is a portable device that stores electrical energy and can be used to charge electronic devices such as smartphones, tablets, and laptops.



**Fig 3. 1. 6 Power Bank**

**ULTRASONIC DISTANCE SENSOR USING RASPBERRY PI PICO**

**1)** **Raspberry Pi Pico:** This version of raspberry PI has all the features which are more than enough for our project. It is also very cost effective and user friendly.



**Fig 3. 1. 7 Raspberry Pi Pico**

**2)** **SSD1306 OLED Display:** The SSD1306 OLED display is highly customizable, allowing users to control various aspects of the display, such as brightness, contrast, and font size. It is used to display the distance measured.



**Fig 3. 1. 8 SSD1306 OLED Display**

**3)** **HC-SR04 Ultrasonic Sensor:** The transmitter emits a burst of ultrasonic sound waves that bounce off objects and reflects them back to the receiver. The receiver then detects the reflected waves and calculates the distance between the sensor and the object based on the time it takes for the waves to travel back and forth.



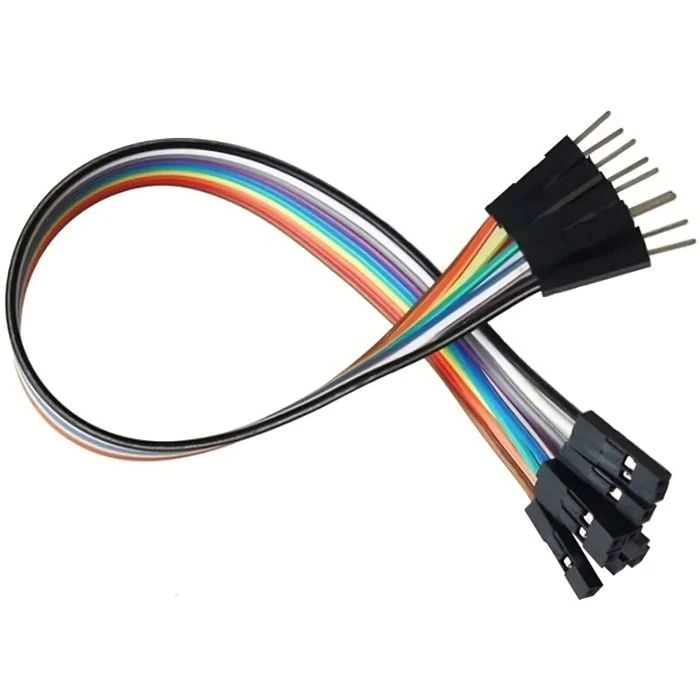
**Fig 3. 1. 9 HC-SR04 Ultrasonic Sensor**

**4)** **Breadboard:** The breadboard provides a convenient platform for testing and modifying electronic circuits, allowing users to easily add or remove components and make changes to the circuit layout without damaging the components.



**Fig 3. 1. 10 Breadboard**

**5)** **Jumper Wires:** Jumper wires are simple electrical cables with connector pins or clips at both ends, used to connect electronic components on a breadboard or circuit board. They are usually made of flexible, stranded copper wire with insulation, and come in a variety of colors, lengths, and gauges.



**Fig 3. 1. 11 Jumper Wires**

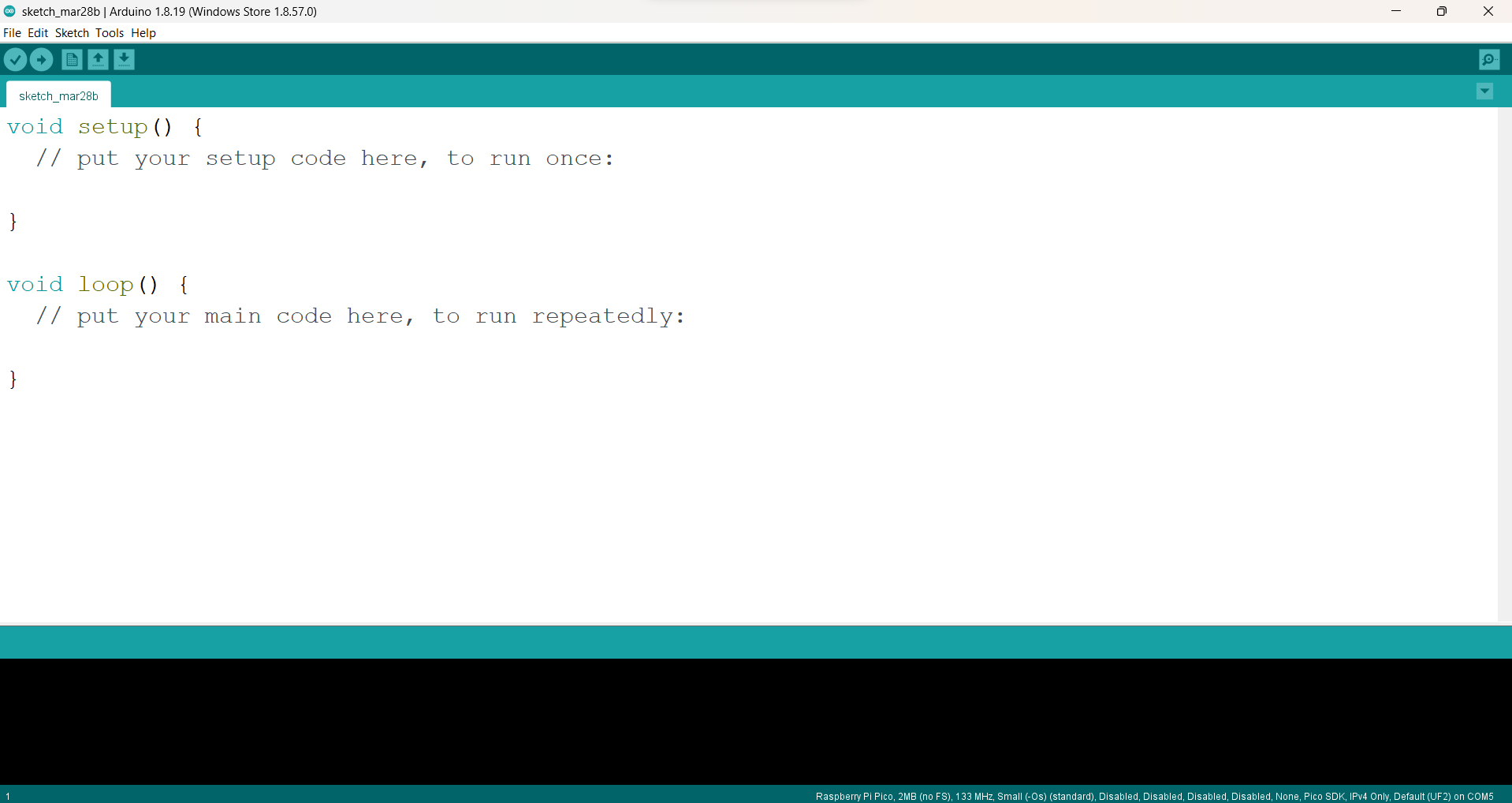
**6)** **Micro USB Cable:** Jumper wires are simple electrical cables with connector pins or clips at both ends, used to connect electronic components on a breadboard or circuit board. They are usually made of flexible, stranded copper wire with insulation, and come in a variety of colors, lengths, and gauges.



**Fig 3. 1. 12 Micro USB Cable**

**3.2** **SOFTWARE USED:**

**1)** **Arduino IDE:**

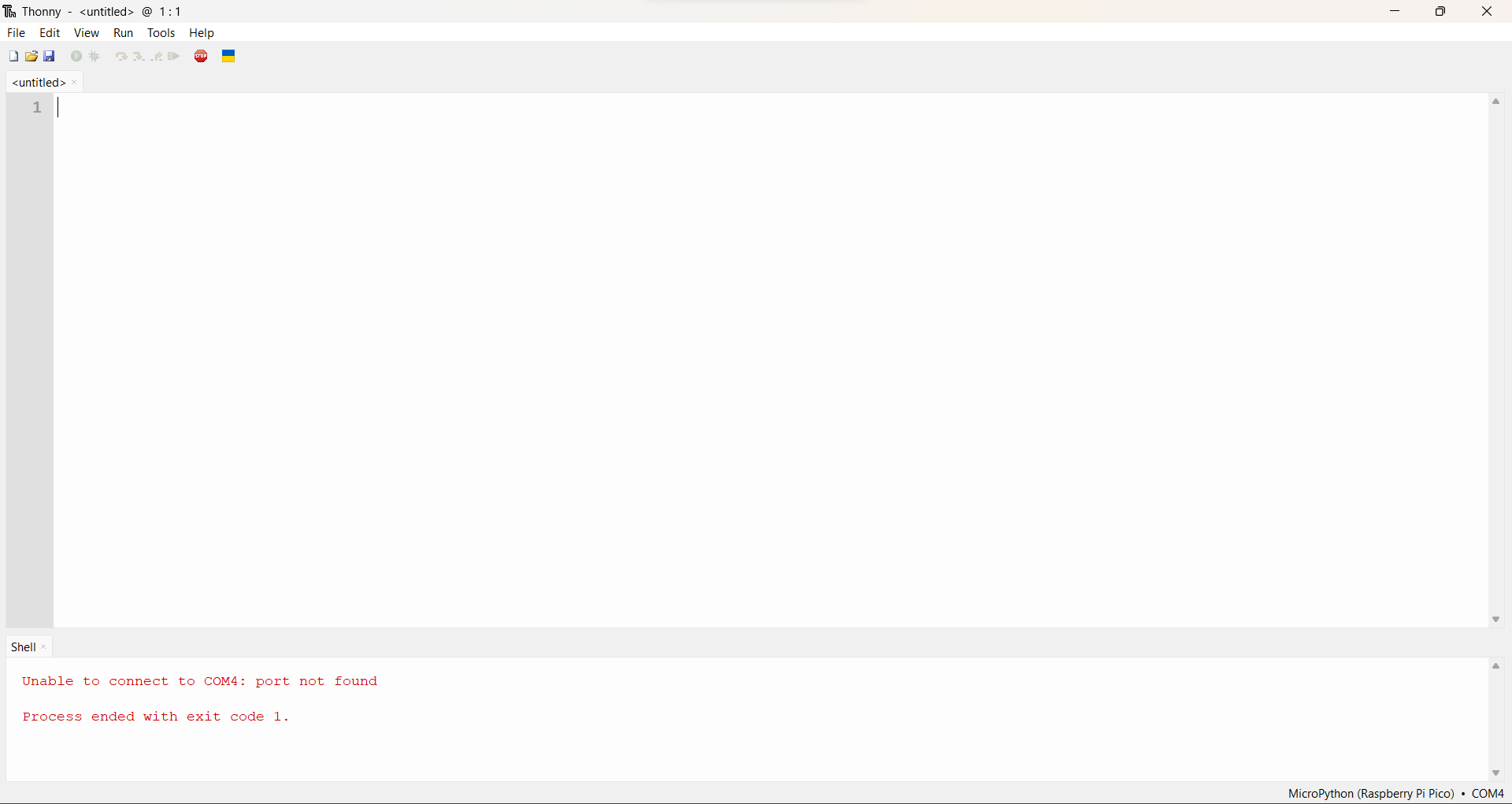
****

**Fig 3. 2. 1 Arduino IDE**

Arduino IDE is an open-source software development environment that allows users to program Arduino boards. Here are some key points about Arduino IDE:

* It is a software platform used for programming and uploading code to Arduino boards.
* The platform is written in Java and is available for Windows, Mac, and Linux operating systems.
* The platform provides a simplified programming language based on C++ that is easy to learn and use for beginners.
* The platform allows users to write and upload code to the Arduino board without the need for a separate compiler or programmer.
* The platform provides a simple, user-friendly interface for writing, editing, and uploading code.
* The platform includes a range of libraries that users can import to add additional functionality to their projects.
* The platform supports a range of Arduino boards, including the Arduino Uno, Nano, Mega, and others.
* The platform provides a serial monitor that allows users to monitor and debug their code while it is running on the board.
* The platform provides a range of example projects that users can use as a starting point for their own projects.
* The platform is free and open-source software, meaning that anyone can use, modify, and distribute it.

**2)** **Thonny IDE:**

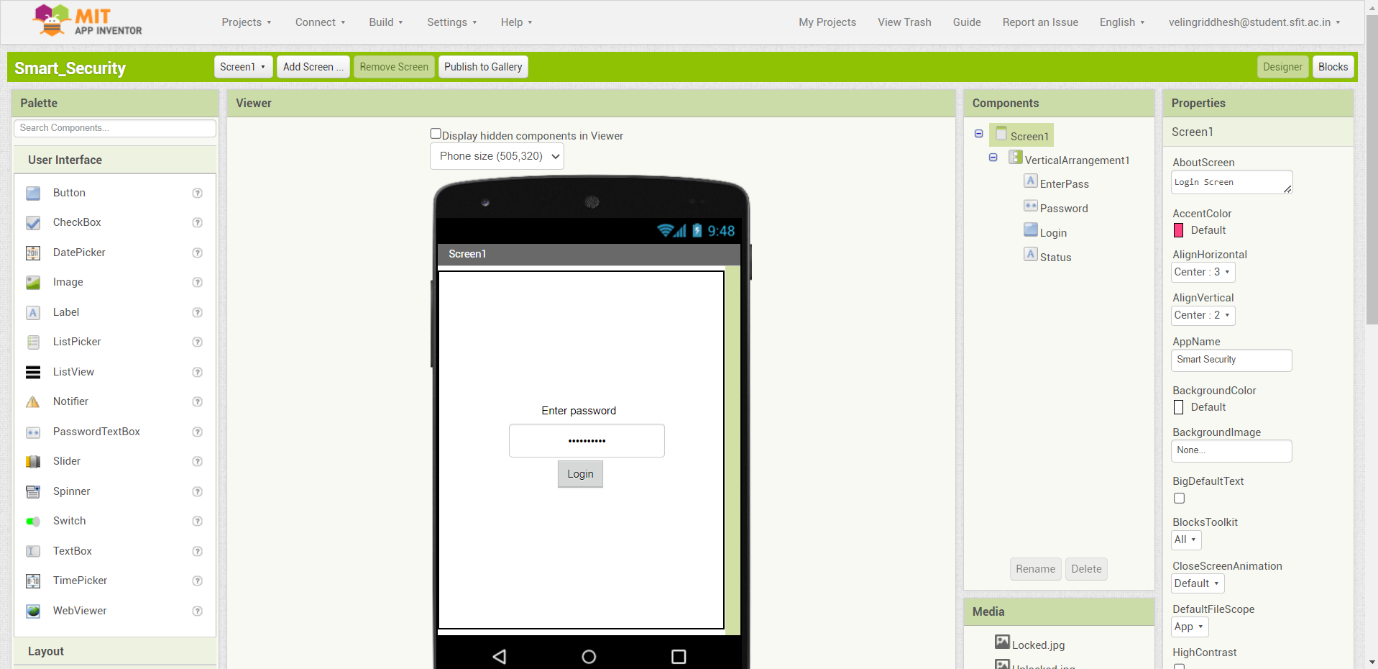
****

**Fig 3. 2. 2 Thonny IDE**

Thonny IDE is a Python Integrated Development Environment (IDE) that allows users to write, test, and debug Python code. Here are some key points about Thonny IDE:

* It is a lightweight software platform that is easy to install and use.
* The platform is written in Python and is available for Windows, Mac, and Linux operating systems.
* The platform provides a simple, user-friendly interface that is easy to navigate and use.
* The platform includes a range of tools that make it easy to write, test, and debug Python code.
* The platform includes a code editor with features such as syntax highlighting, code folding, and code completion.
* The platform includes a debugger that allows users to step through their code line by line and track the values of variables.
* The platform provides a shell that allows users to run Python commands and test code snippets.
* The platform includes a range of plug-ins that add additional functionality, such as support for microcontrollers and remote debugging.
* The platform is free and open-source software, meaning that anyone can use, modify, and distribute it.
* Thonny IDE is designed to be easy to use for beginners, but also includes advanced features that make it suitable for professional developers.

**3)** **MIT App Inventor:**

****

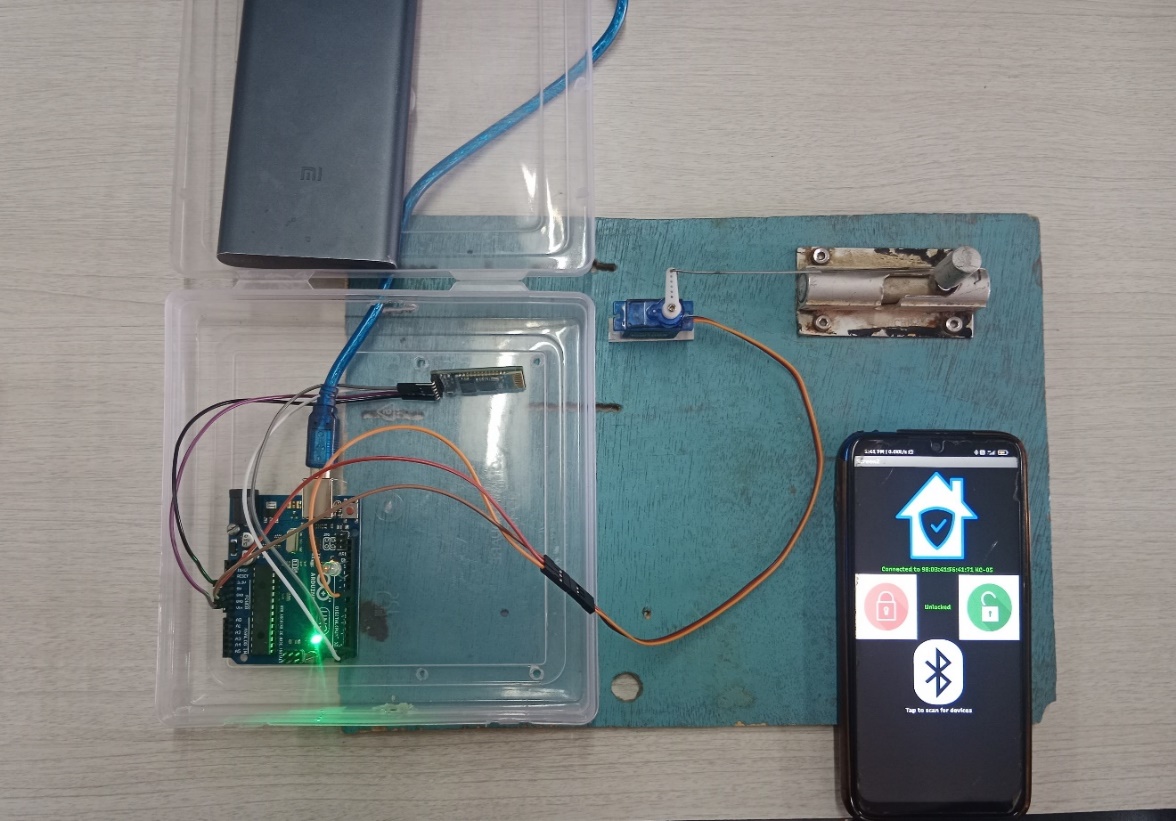
**Fig 3. 2. 3 MIT App Inventor**

MIT App Inventor is a cloud-based software development platform that enables users to create mobile apps for Android devices without prior programming knowledge. Here are some key points about MIT App Inventor:

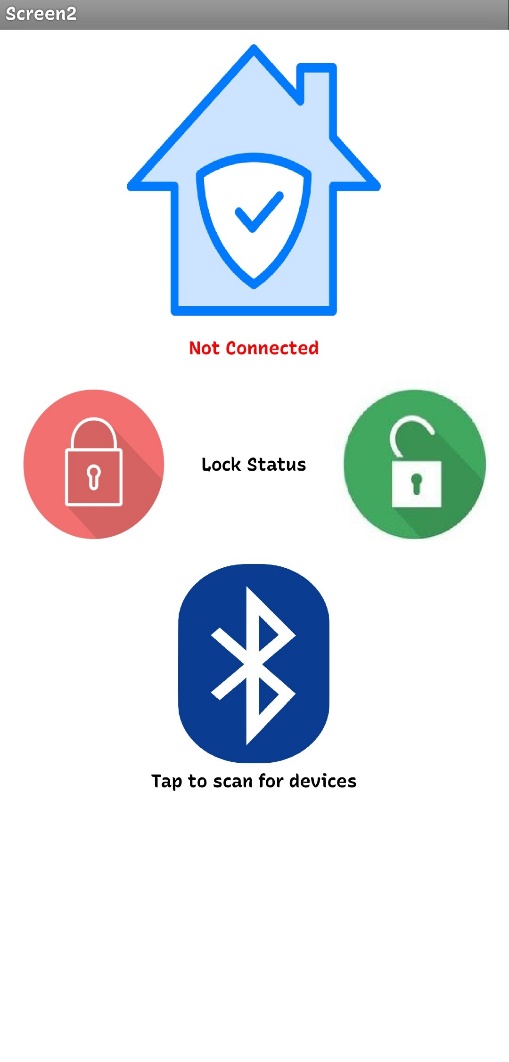
* It was originally created by Google but is now maintained by MIT.
* It uses a visual, drag-and-drop interface for building apps, making it easy for beginners to use.
* Users can create apps by dragging and dropping components onto a canvas and then programming their behaviour using blocks of code.
* The blocks-based programming language used by MIT App Inventor is like Scratch, a visual programming language aimed at children.
* MIT App Inventor provides a range of pre-built components, such as buttons, text boxes, and sensors, that can be used to create functional apps.
* Users can also import custom components or create their own.
* Apps built with MIT App Inventor can access a range of phone features, such as the camera, GPS, and accelerometer.
* MIT App Inventor provides a live testing feature that allows users to test their apps in real-time on an Android device connected to their computer.
* Once an app is complete, users can export it as an APK file and distribute it through the Google Play Store or other app stores.
* MIT App Inventor is free and open-source software, meaning that anyone can use, modify, and distribute it.

**4. MINIPROJECT IMPLEMENTATION STEPS USING ARDUINO**

**4.1** **IMPLEMENTED STEPS:**

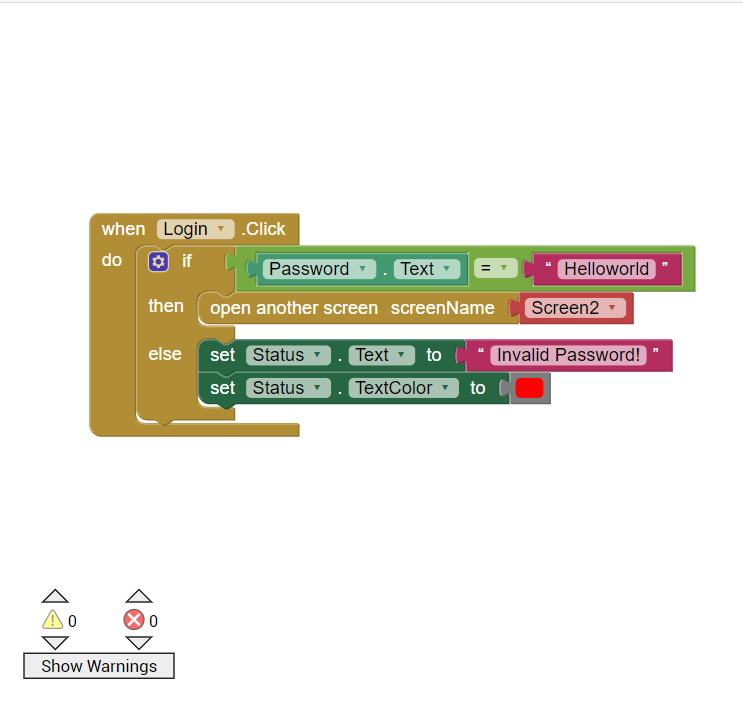
****

**Fig 4. 1. 1 Circuit Connection**

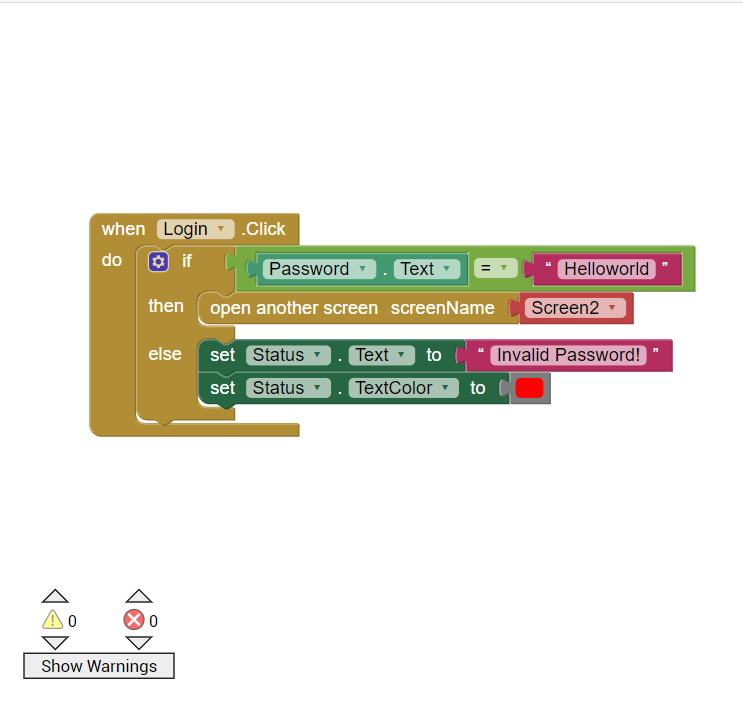


**Fig 4. 1. 3 App Login UI**

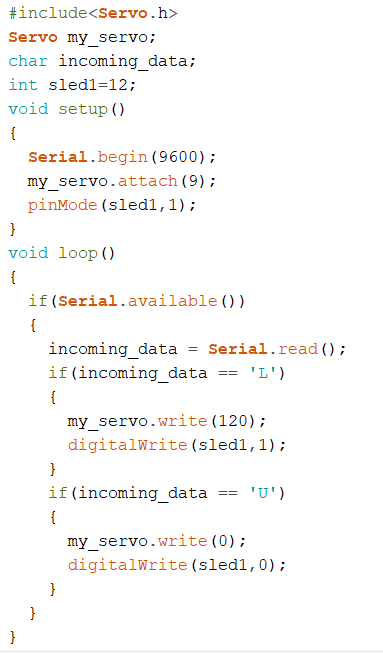
**Fig 4. 1. 2 Control Page UI**



**Fig 4. 1. 4 Login Page Code**



**Fig 4. 1. 5 Control Page Code**

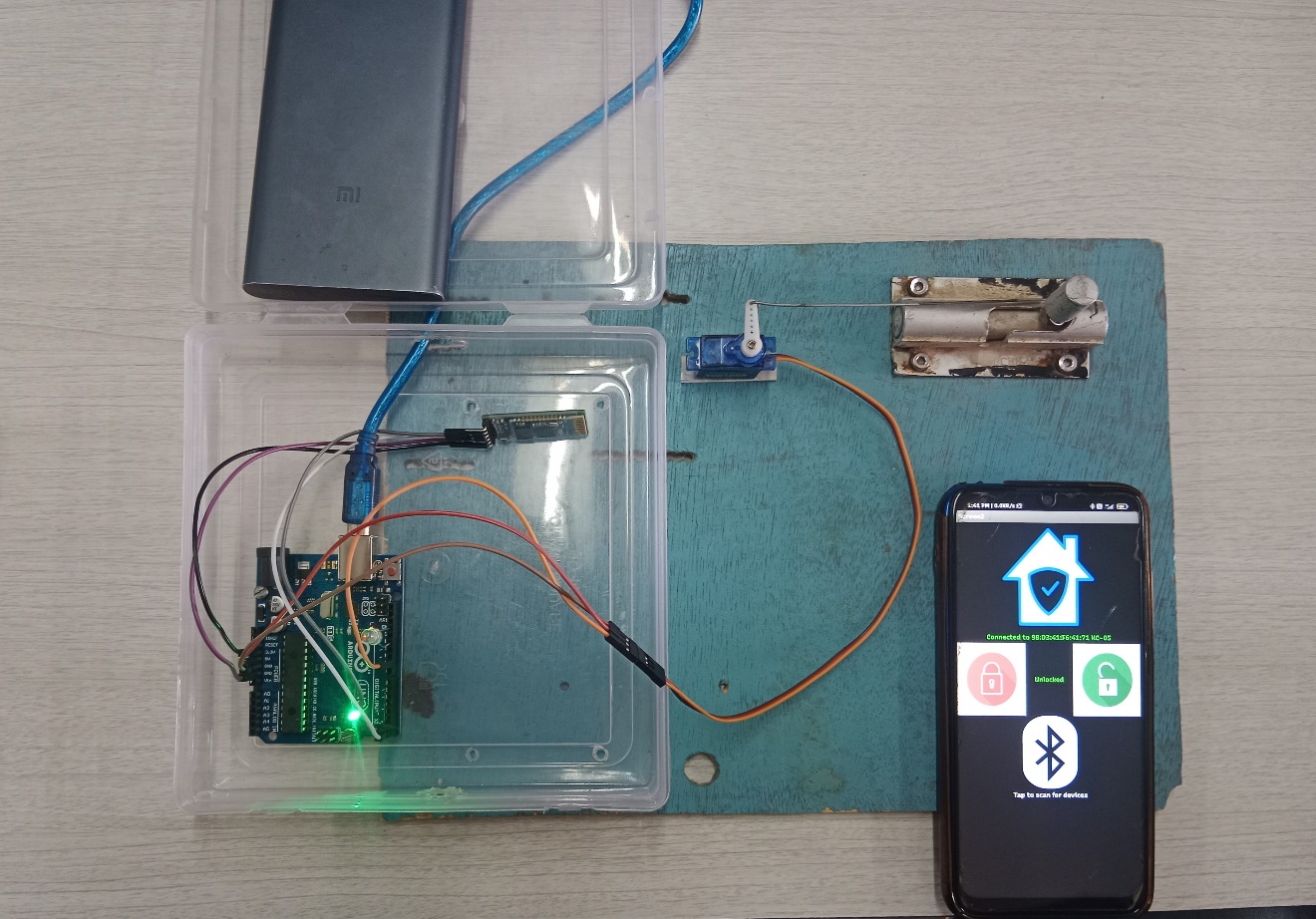
****

**Fig 4. 1. 6 Arduino code**

**4.2 RESULTS:**



**Fig 4. 2. 1 When Lock button is pressed on mobile application**



**Fig 4. 2. 2 When Unlock button is pressed on mobile application**

**4.3 RESULTS & DISCUSSION:**

The circuit diagram connection is as follows :

|  |  |
| --- | --- |
| **Arduino Pins** | **Components and Pin** |
|  |  |
| Pin 9(PWM) | Servo : Yellow Wire (signal input wire) |
| Rx | Bluetooth Module : TX |
| Tx | Bluetooth Module : RX |
| Pin 12 | LED |
| 5V | Servo : Red Wire (VCC wire) |
| 3.3 V | Bluetooth Module : VCC |
| GND | Servo, Bluetooth Module & LED : GND |

The user must enter the secret password on the ‘App Login UI’ page and then he can access the control page. On the ‘Control Page the user first has to connect the device’s Bluetooth with HC-05 module. Then the user can lock and unlock the door by pressing the designated buttons.

The current status of the door lock can be monitored on the control page or through the Status LED from the circuit.

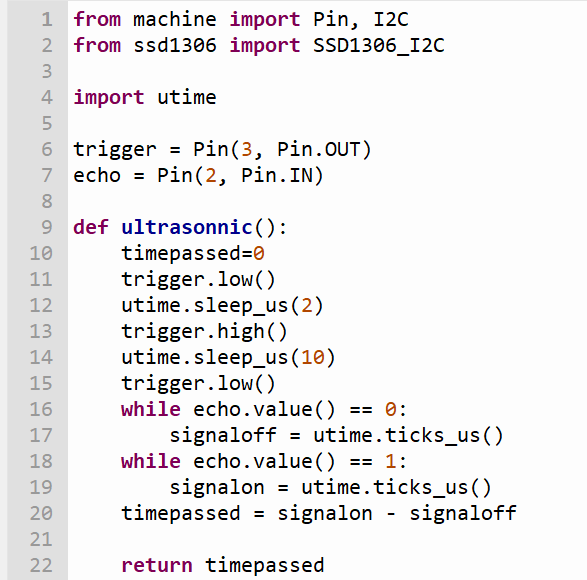
In the code, the required servo library is first imported. Then an object is created for the servo motor operations and the required variables for the incoming data & LED pin are created. As the LED is connected on pin 12, sled1 variable is given value 12. In the setup() function, the baud rate is set to 9600 and the servo motor is attached to PWM pin 9 of Arduino board and the LED pin is set high. In the loop() function, the serial connection with Bluetooth module is first is checked and when true, the data received is stored in the incoming\_data variable created. If the incoming data is letter ‘L’, the servo motor is rotated by 120 degrees (from rest position) and the LED is turned ON. This is the Lock condition. Whereas, if the incoming data is letter ‘U’, the servo motor is rotated by 0 degrees (i.e., back to its rest position) and the LED is turned off. This is the Unlock condition.

**5. MINIPROJECT IMPLEMENTATION STEPS USING RASPBERRY PI**

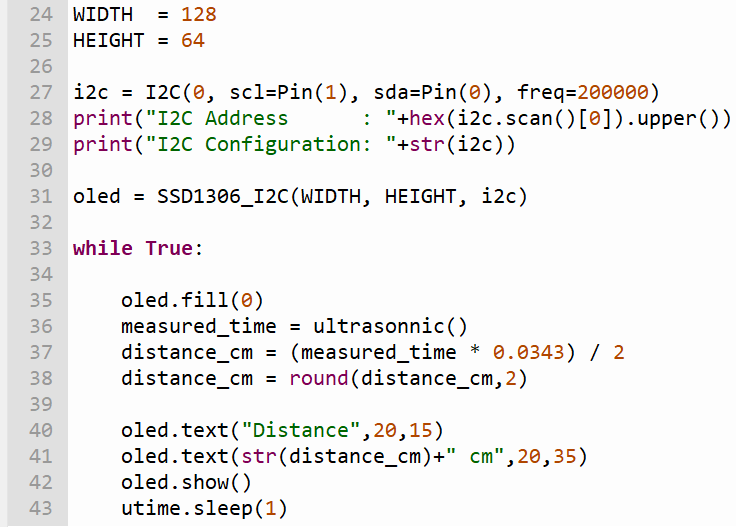
**5.1** **IMPLEMENTED STEPS:**

****

**Fig 5. 1. 1 Circuit Connection**

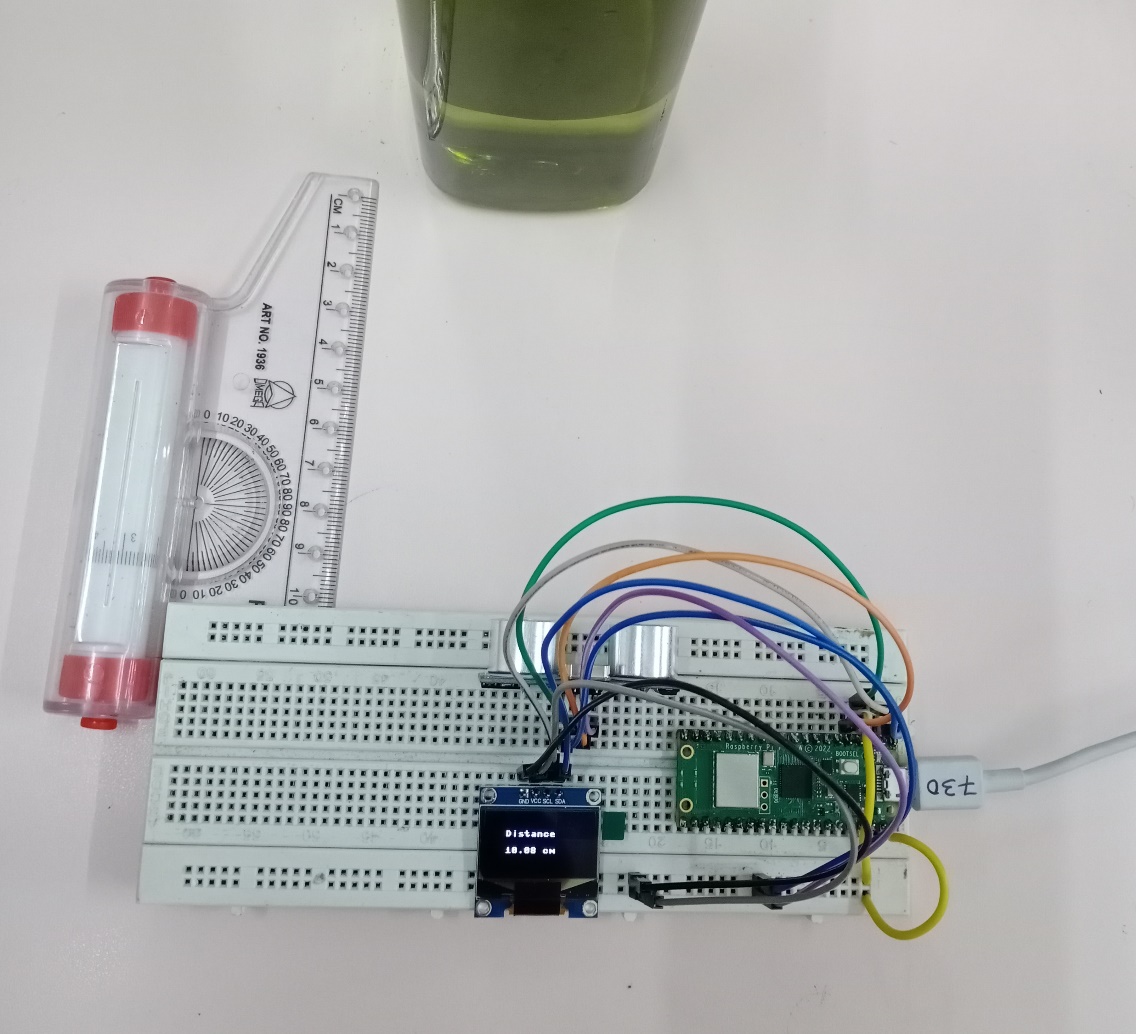
****

**Fig 5. 1. 2 Raspberry Pi code**

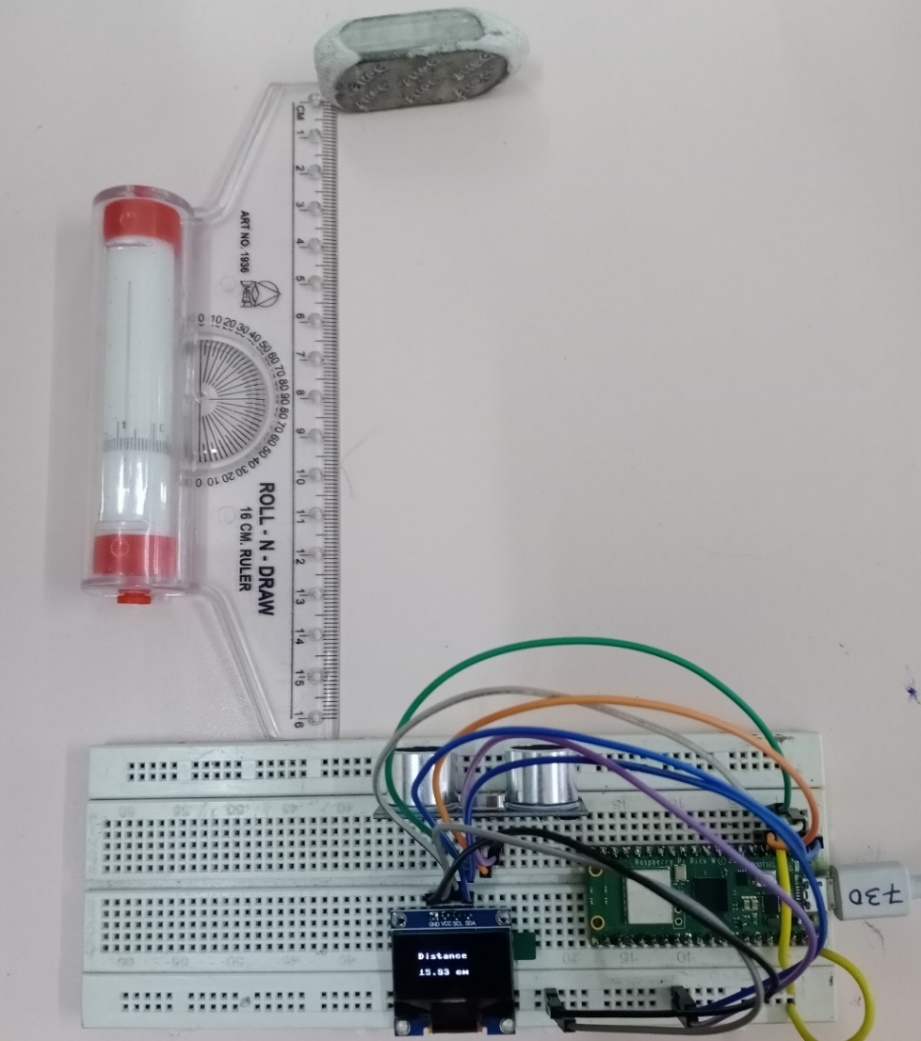


**Fig 5. 1. 3 Raspberry Pi Code**

**5.2 RESULTS:**

****

**Fig 5. 2. 1 Object at distance of 10cm**



**Fig 5. 2. 2 Object at distance of 16 cm**

**5.3 RESULTS AND DISCUSSION:**

|  |  |
| --- | --- |
| **Rpi Pico Pins** | **Components and Pin** |
|  |  |
| GP0 | OLED Display : SDA |
| GP1 | OLED Display : SCL |
| GP2 | HC-SR04 : Echo |
| GP3 | HC-SR04 : Trig |
| VCC | OLED Display & HC-SR04 : VCC |
| GND | OLED Display & HC-SR04 : GND |

After making all the connections, as soon as VCC of OLED Display and HC-SR04 Sensor is connected to the VCC of Rpi Pico, we can see the distance getting stabilized on the OLED Display screen.

We can verify the distance measuring ability of the sensor by using a scale between the obstacle and sensor. Its accuracy is up to ± 5cm. The sensor can measure obstacles from anywhere between 4cm to 400cm.

As the circuit is mounted on the breadboard, the circuit is portable using a powerbank as power supply.

In the code, the first three lines import necessary modules from the microcontroller's library. Specifically, it imports the Pin and I2C classes from the machine module and the SSD1306\_I2C class from the ssd1306 module. The next three lines define two pins: trigger, and echo, which are connected to the ultrasonic sensor. The trigger pin sends a signal to the sensor to start a measurement, and the echo pin receives a signal back from the sensor. The ultrasonic function is defined to calculate the time it takes for the ultrasonic signal to travel from the sensor to an object and back. It sends a pulse on the trigger pin for 10 microseconds, which starts the measurement. The echo pin will then go high when the signal reaches an object, and it will go low again when the signal returns to the sensor. The time difference between these two events is measured and returned. An I2C object is created with the I2C class, which initializes the I2C communication protocol with a specified frequency on two pins SDA and SDA. An SSD1306\_I2C object is created with the WIDTH, HEIGHT, and i2c parameters, which initializes the OLED display using the I2C communication protocol. The distance is calculated by multiplying the time taken by the ultrasonic signal by the speed of sound (0.0343 cm/microsecond) and dividing by 2 since the signal must travel to the object and back. The distance is then rounded to two decimal places. The distance is displayed on the OLED display using the text method, and the display is updated with the show method. The loop waits for 1 second using the sleep method in the utime module before repeating the distance measurement and display update.

**REFERENCES:**

**PASSWORD BASED SMART LOCKING SYSTEM USING ARDUINO**

<https://www.electronicsforu.com/electronics-projects/otp-based-smart-wireless-locking-system>

<https://www.instructables.com/Arduino-AND-Bluetooth-HC-05-Connecting-easily/>

HC-05 Bluetooth Module with Arduino-MIT App Inventor :

<https://youtu.be/aQcJ4uHdQEA>

**ULTRASONIC DISTANCE SENSOR USING RASPBERRY PI PICO**

<https://how2electronics.com/hc-sr04-ultrasonic-distance-sensor-with-raspberry-pi-pico/>

Getting started with Raspberry Pi Pico:

<https://www.youtube.com/watch?v=vO_2XWJDF70>