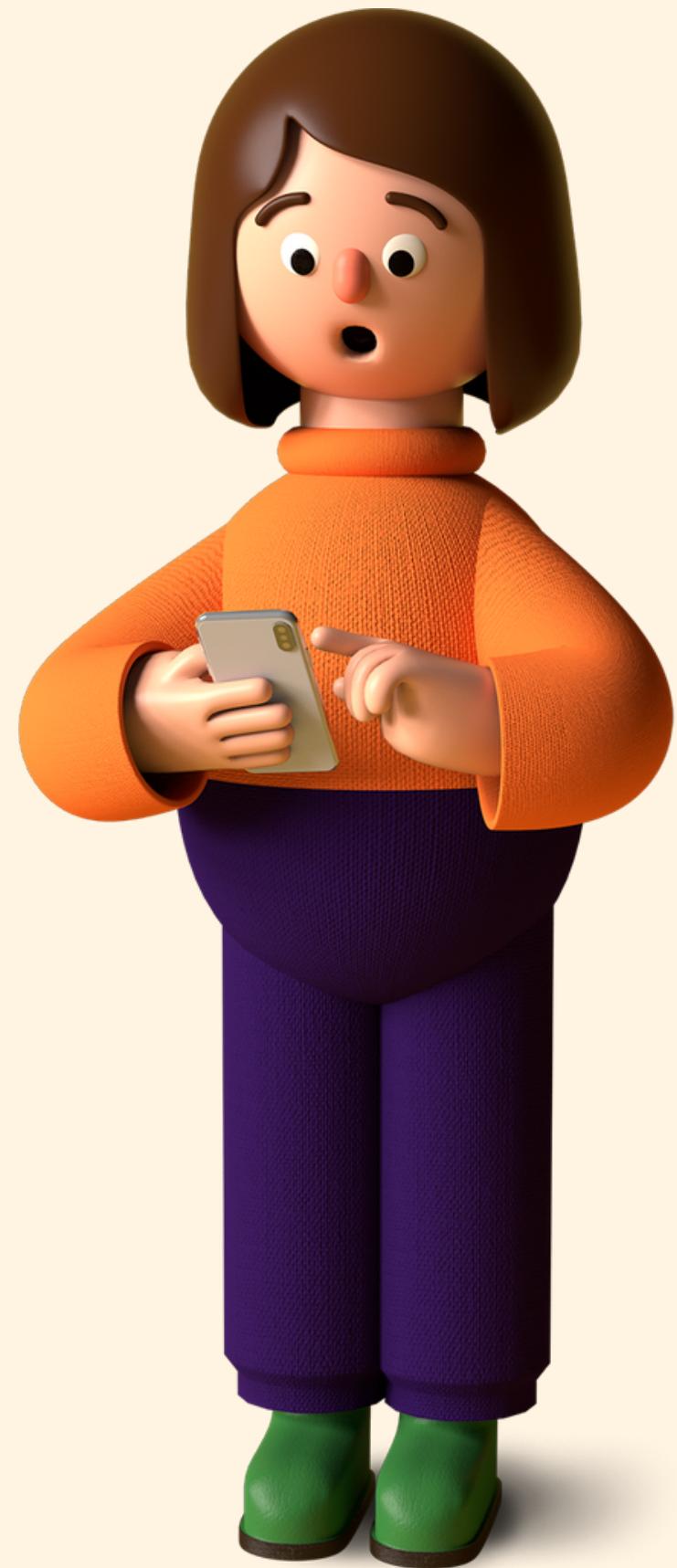


COMPLETE HEALTH CARE



**ENCRYPTED DATA
STREAM**

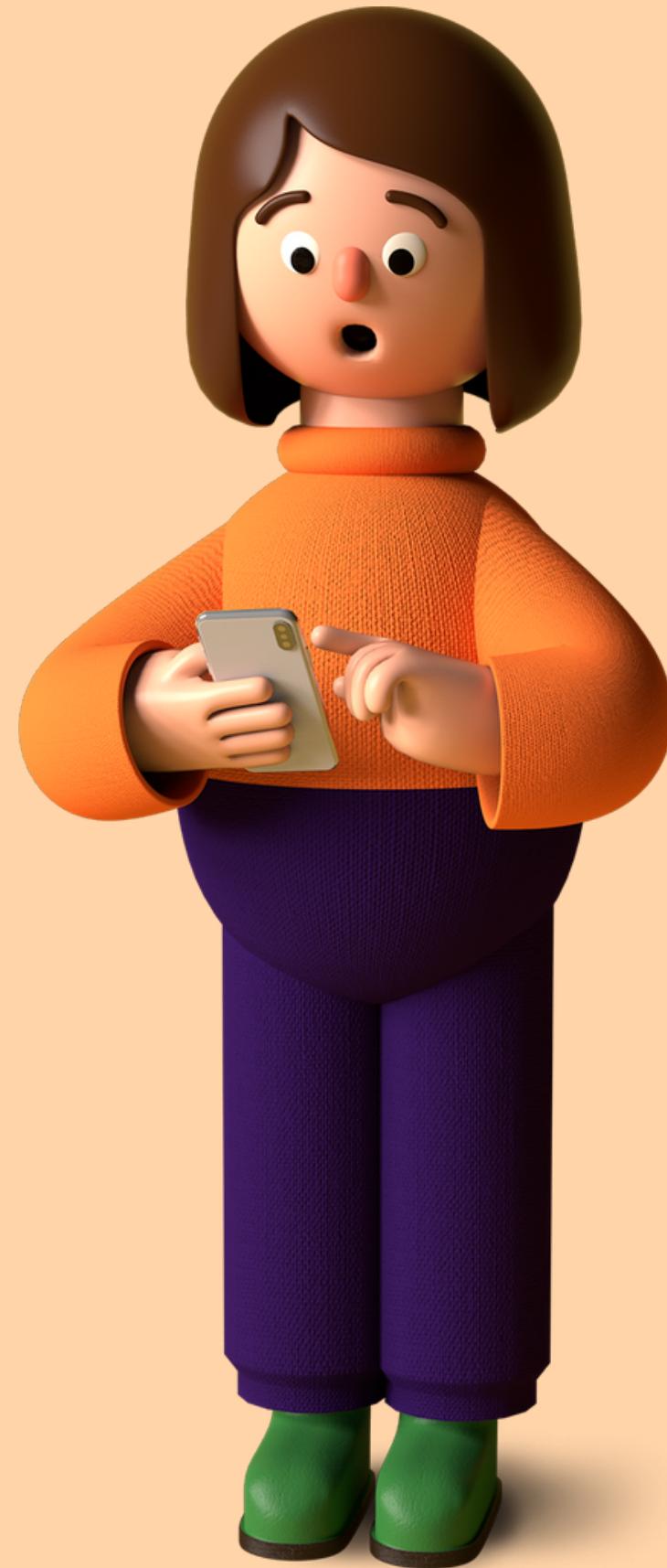
Transmitter



Bluetooth Signal



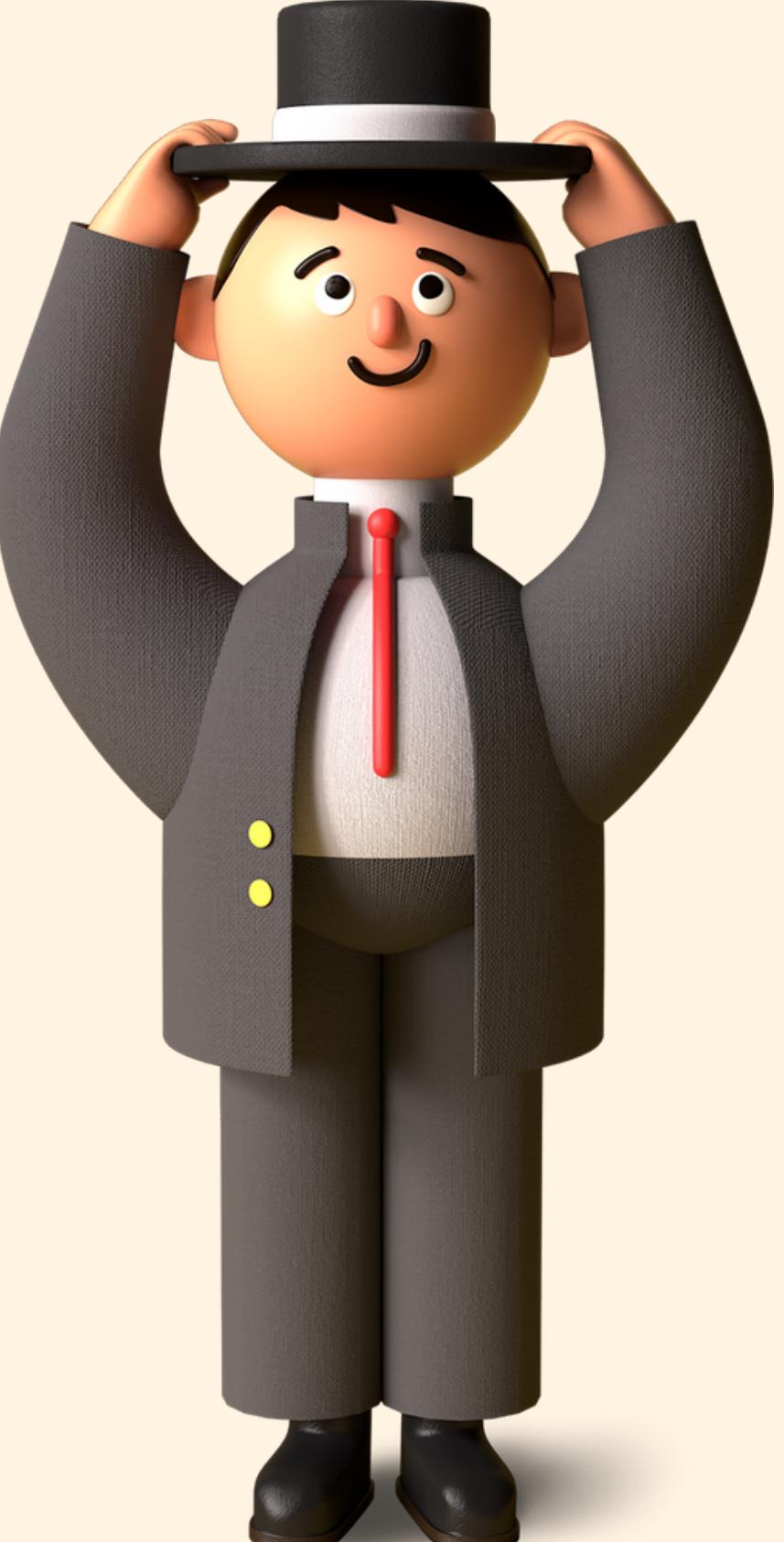
**Controller Module/
Receiver End**



INTRODUCTION



When a person is very sick or on complete bed rest he always needs to have someone around with him with this project I have got rid of this problem I have made a wireless system that works on Bluetooth and responded to their specific task



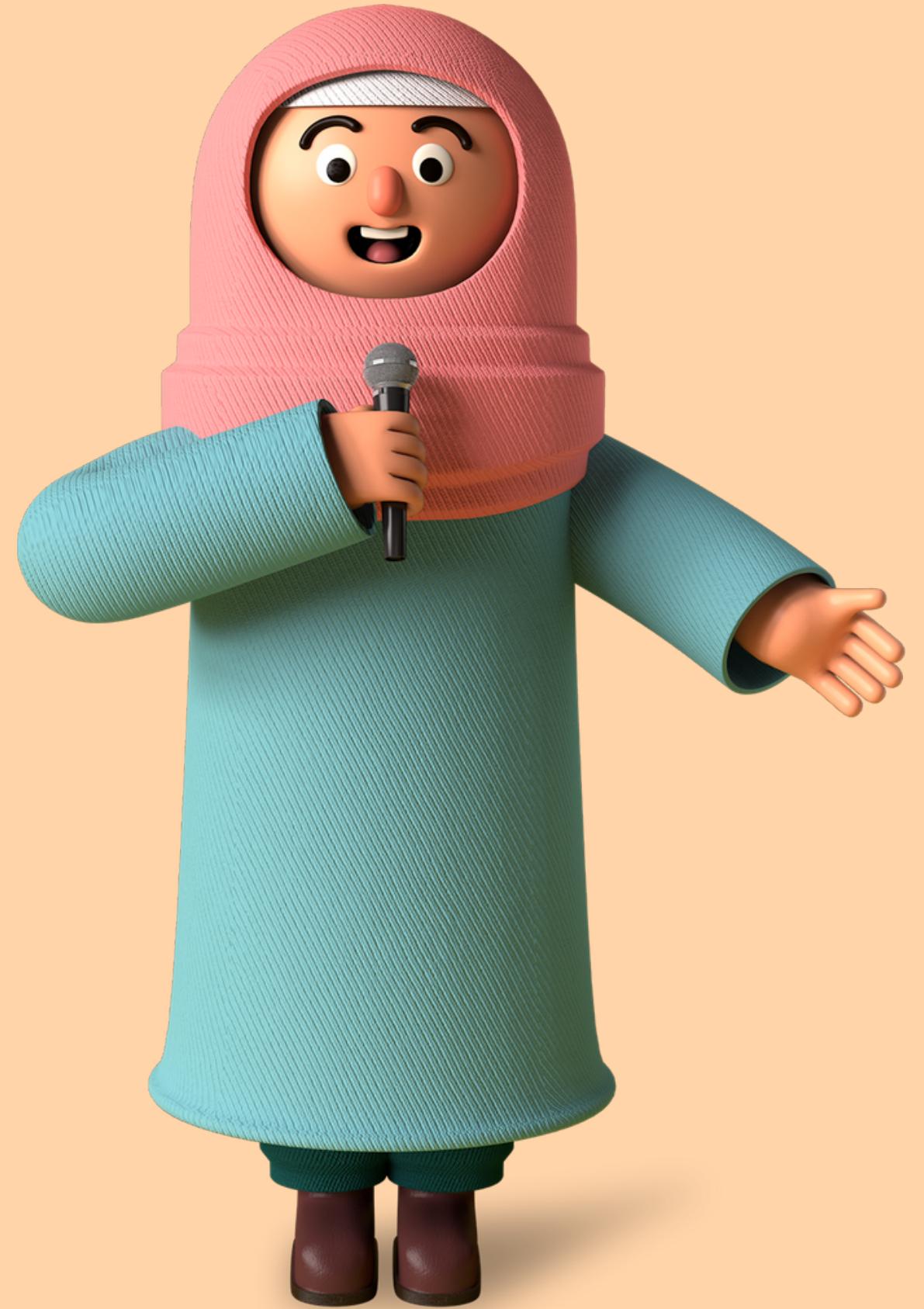
OEM (Original Equipment Manufacturer)

1-way only sends a command to the Health but there is no confirmation.

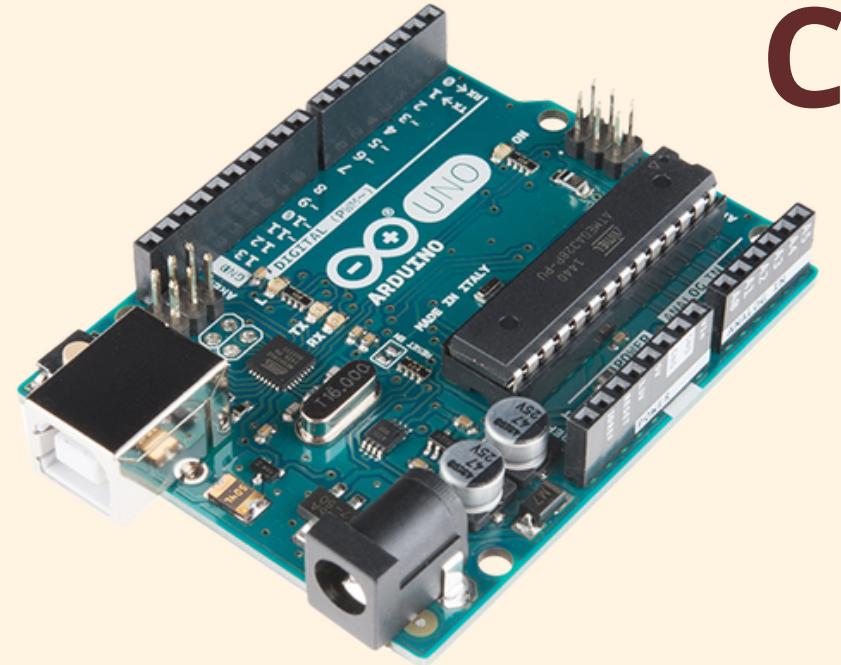
Instead, we proposed
2-way sends command as well as confirmation to the user

AIM

To make a wireless health system using Arduino and HC05 Bluetooth Module



COMPONENTS REQUIRED

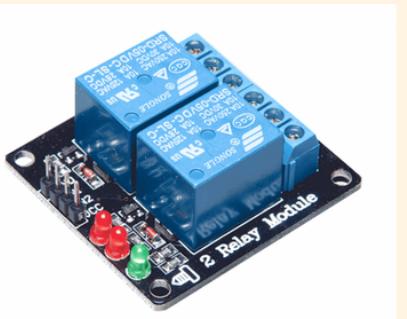
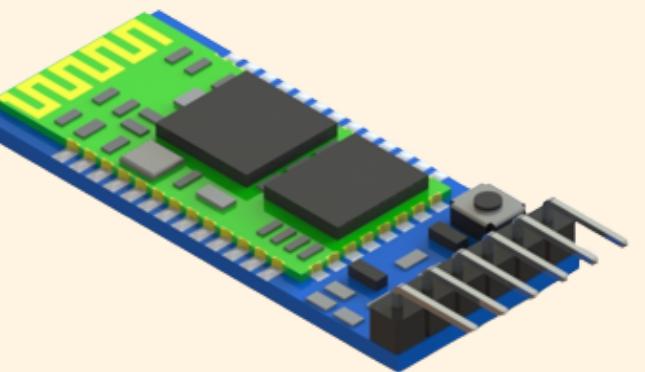


Arduino uno: a microcontroller that is the main processing unit

Servo Motor : a servo motor is used to make door lock

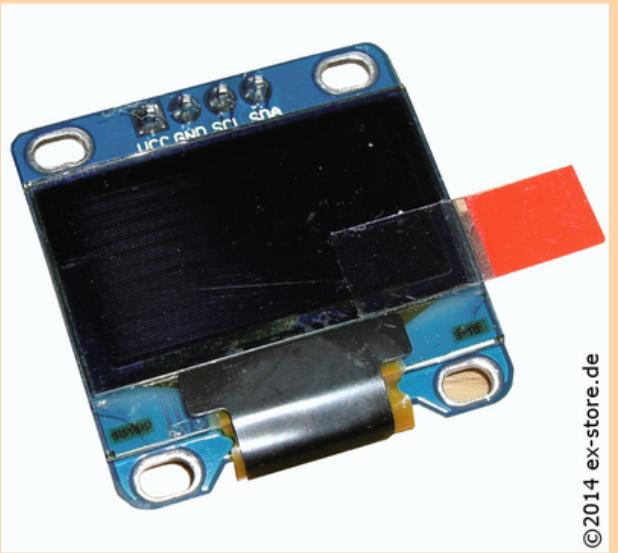


HC05 Bluetooth Module: uses serial communication to communicate with the electronics



Relay: used for switching and activating devices.

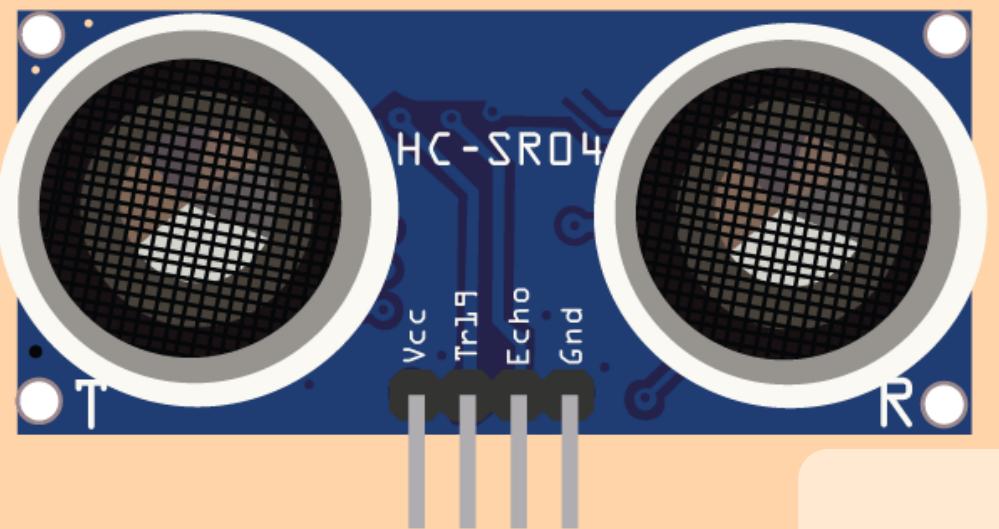




©2014 ex-store.de

Oled: This led is used to show the user if someone approaching or which device is on .

Pir sensor : This Sensor is used to detect motion

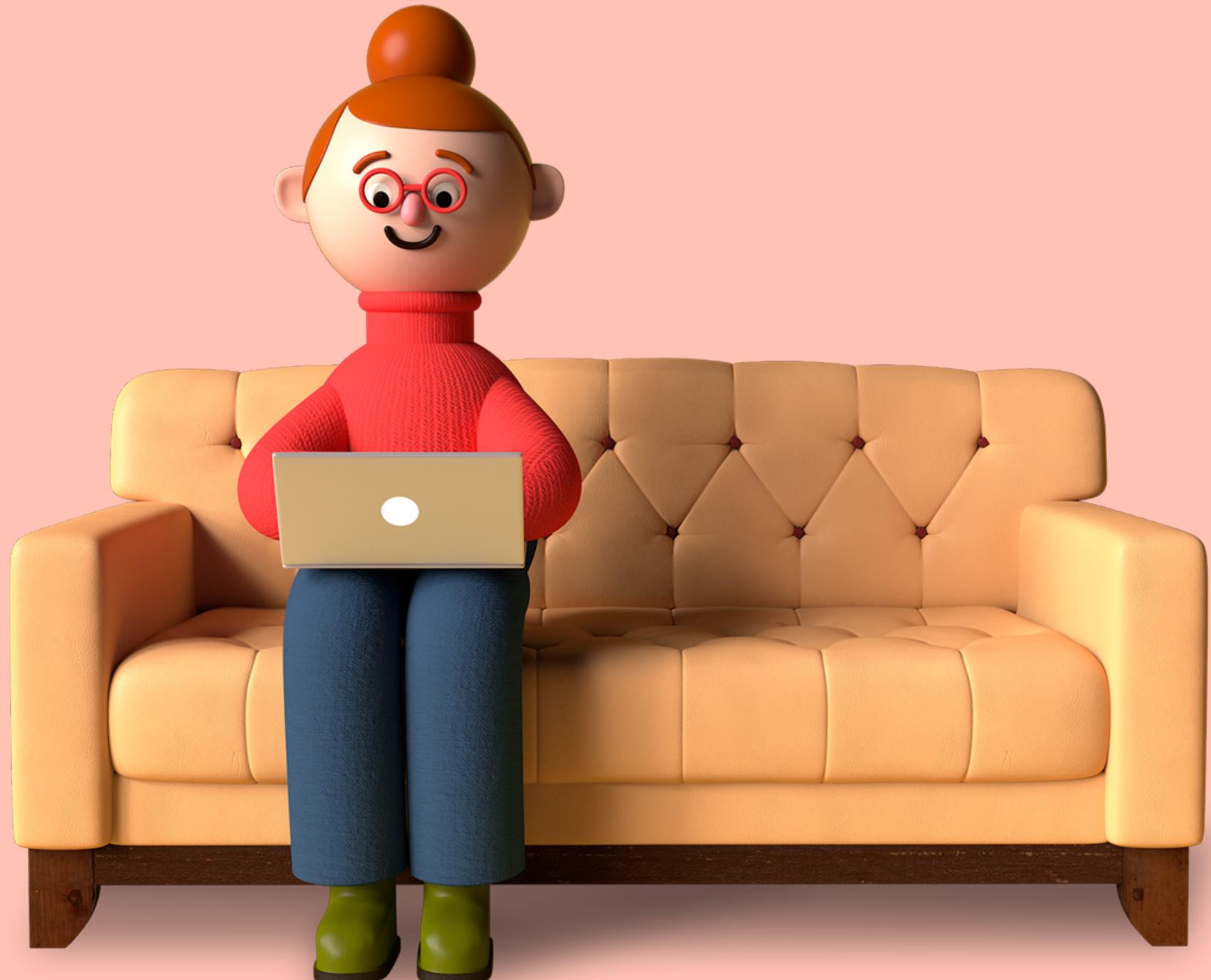


Diodes: This device is used to measure the distance

Water Motor: This device is used to deliver water to patient



FEATURES



01 No Personal
assistance required

02 Fully customizable

03 Respond to every single
thing

FEATURES



04

Wireless

05

Easy to implant

CONSTRUCTION

The circuit is divided into two parts:

1. Arduino module
2. Bluetooth module

So we made a connection between our HC05 module with our app whenever we send a signal through our phones it get received by HC05 module and after this with the help of Arduino The desired output is received

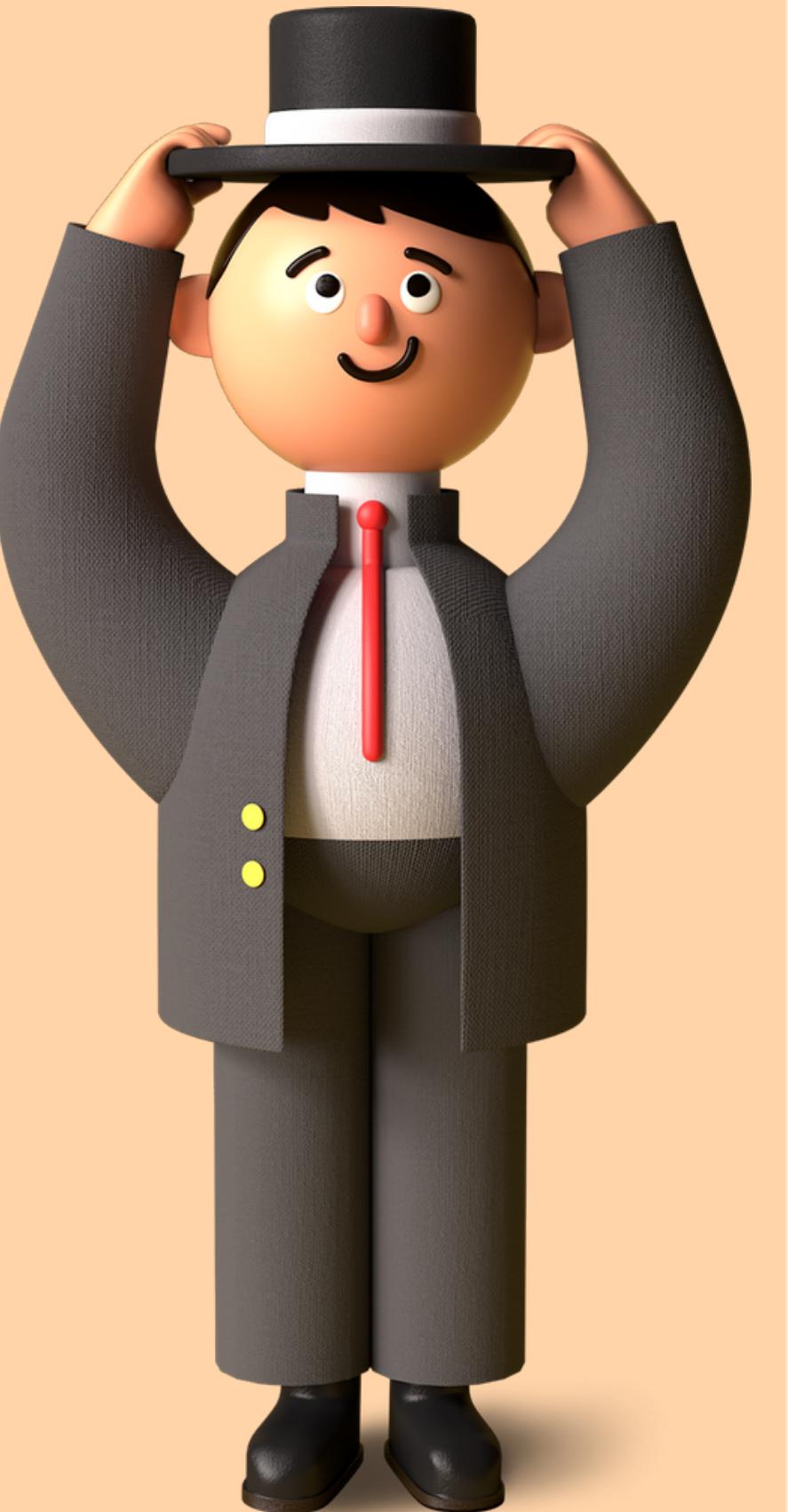


MIT APP INVENTER

MIT App Inventor is an online platform designed to teach computational thinking concepts through development of mobile applications. Students create applications by dragging and dropping components into a design view and using a visual blocks language to program application behavior.



CODE

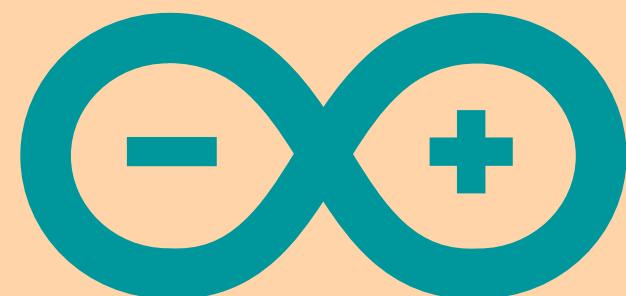


```
#include <Servo.h>
#define echoPin 9
#define trigPin 8
#include <SPL.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
int sensor = 10;
int state = LOW;
int val = 0;
long duration;
int distance;
int servoPin = 6;
int servoPin2 = 7;
Servo Servo;
Servo Servo2;
void setup()
{
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Servo.attach(servoPin);
    Servo2.attach(servoPin2);
    Serial.begin(9600);
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(sensor, INPUT);
    Serial.begin(9600);
}

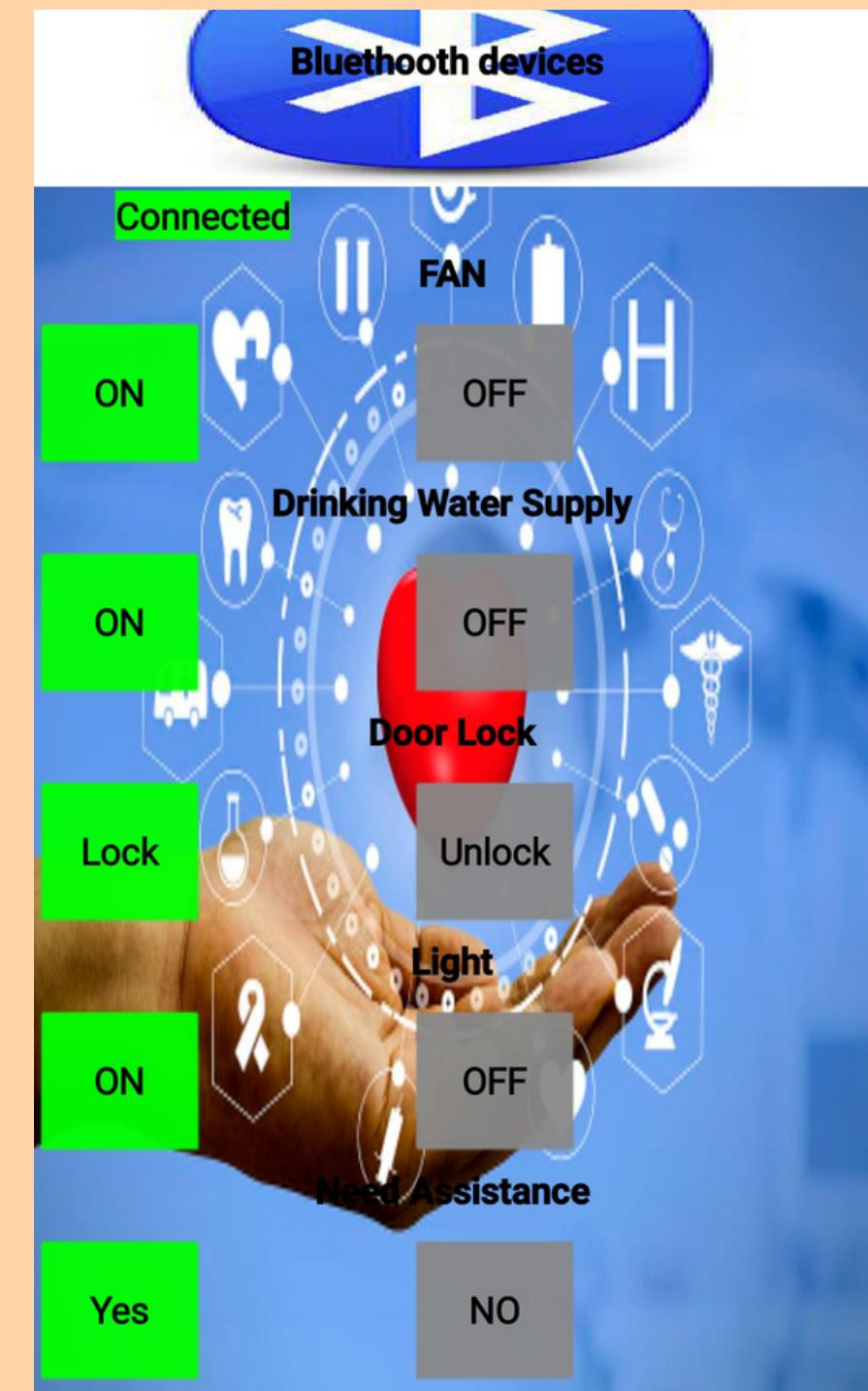
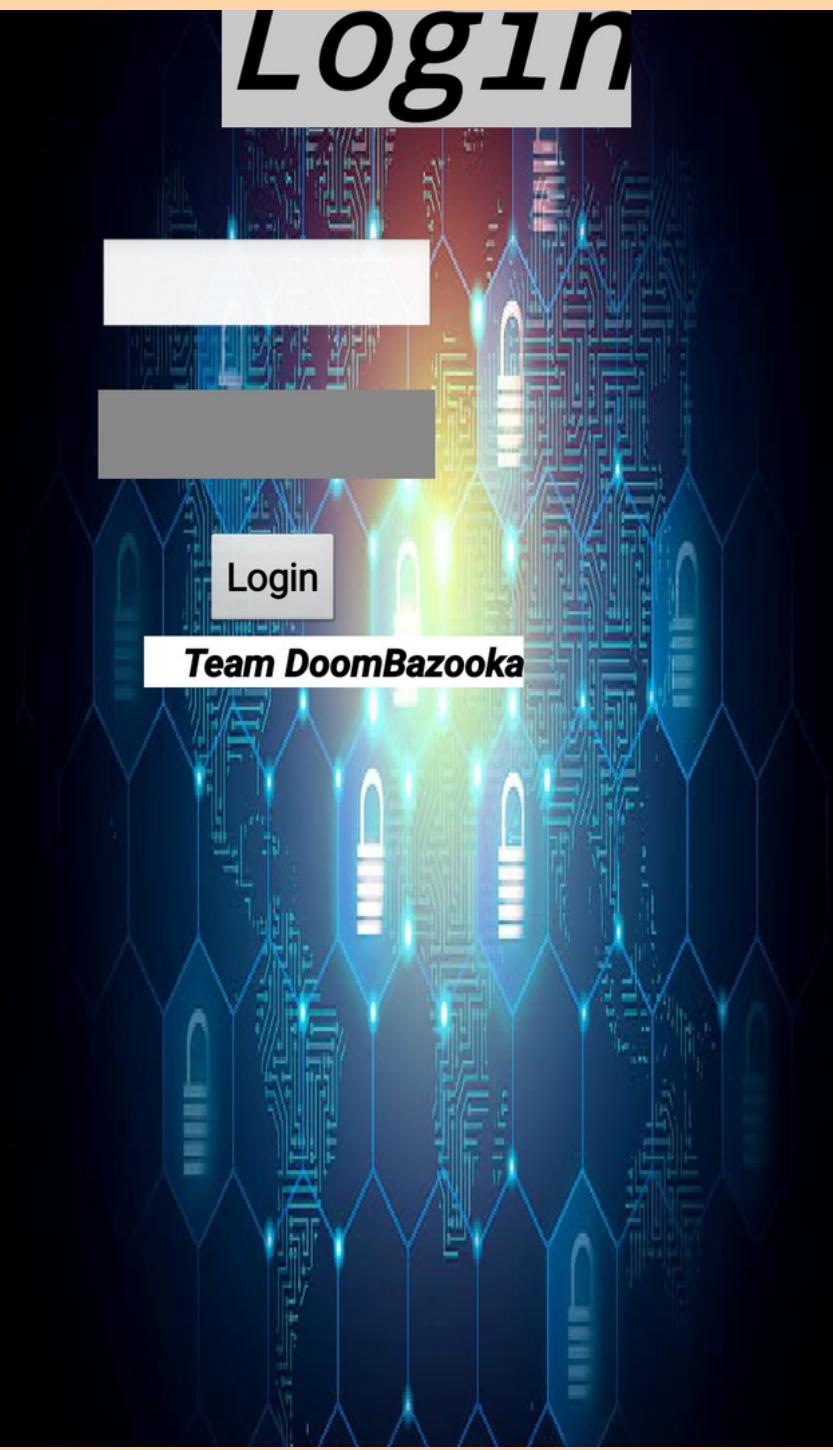
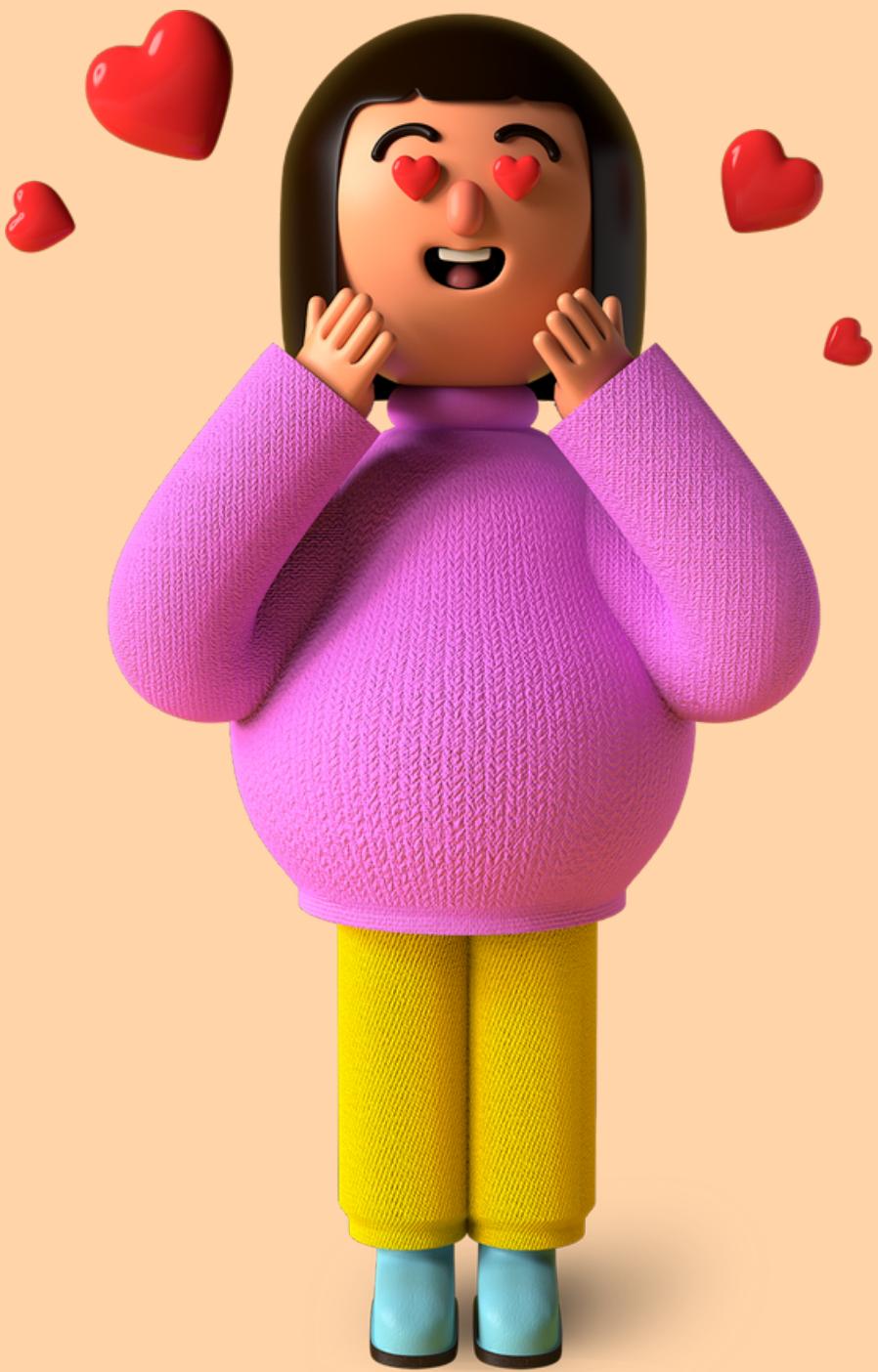
void loop()
{
    char command;
    while (Serial.available() > 0)
    {
        command = Serial.read();
        // Water
        if (command == 'W')
        {
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(0,0);
            display.println("Turning On Water supply");
            delay(5000);
            digitalWrite(2, HIGH);
        }
        else if (command == 'w')
        {
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(0,0);
            display.println("Turning On Water supply");
            delay(5000);
            display.clearDisplay();
            digitalWrite(2, LOW);
        }
        // fan
        else if (command == 'F')
        {
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(0,0);
            display.println("Turning On Fan");
            delay(5000);
            display.clearDisplay();
            digitalWrite(3, HIGH);
        }
        else if (command == 'f')
        {
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(0,0);
            display.println("Turning Off Fan");
            delay(5000);
            display.clearDisplay();
            digitalWrite(3, LOW);
        }
        // light
        else if (command == 'L')
        {
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(0,0);
            display.println("Turning On Light");
            delay(5000);
            display.clearDisplay();
            digitalWrite(4, HIGH);
        }
        else if (command == 'l')
        {
            display.setTextSize(1);
            display.setTextColor(WHITE);
            display.setCursor(0,0);
            display.println("Turning Off Fan");
            delay(5000);
            display.clearDisplay();
            digitalWrite(4, LOW);
        }
    }
}

// assistance
else if (command == 'A')
{
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Calling Assistance");
    delay(5000);
    display.clearDisplay();
    digitalWrite(5, HIGH);
}
else if (command == 'a')
{
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("cancelling assistance");
    delay(5000);
    display.clearDisplay();
    digitalWrite(5, LOW);
}
// servo 6,7
else if (command == 'S')
{
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Locking..");
    delay(5000);
    display.clearDisplay();
    Servo2.write(0);
    delay(1000);
    Servo1.write(0);
    delay(1000);
    Servo1.write(180);
    delay(1000);
}
else if (command == 's')
{
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Unlocking");
    delay(5000);
    display.clearDisplay();
    Servo2.write(0);
    delay(1000);
    Servo2.write(180);
    delay(1000);
}
// Ultrasonic Sound Sensor
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
if (distance<25){
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("An object is approaching");
    display.println(distance);
    delay(500);
    display.clearDisplay();
}
// PIR Sensor
val = digitalRead(sensor);
if (val == HIGH) {
    delay(500);
}

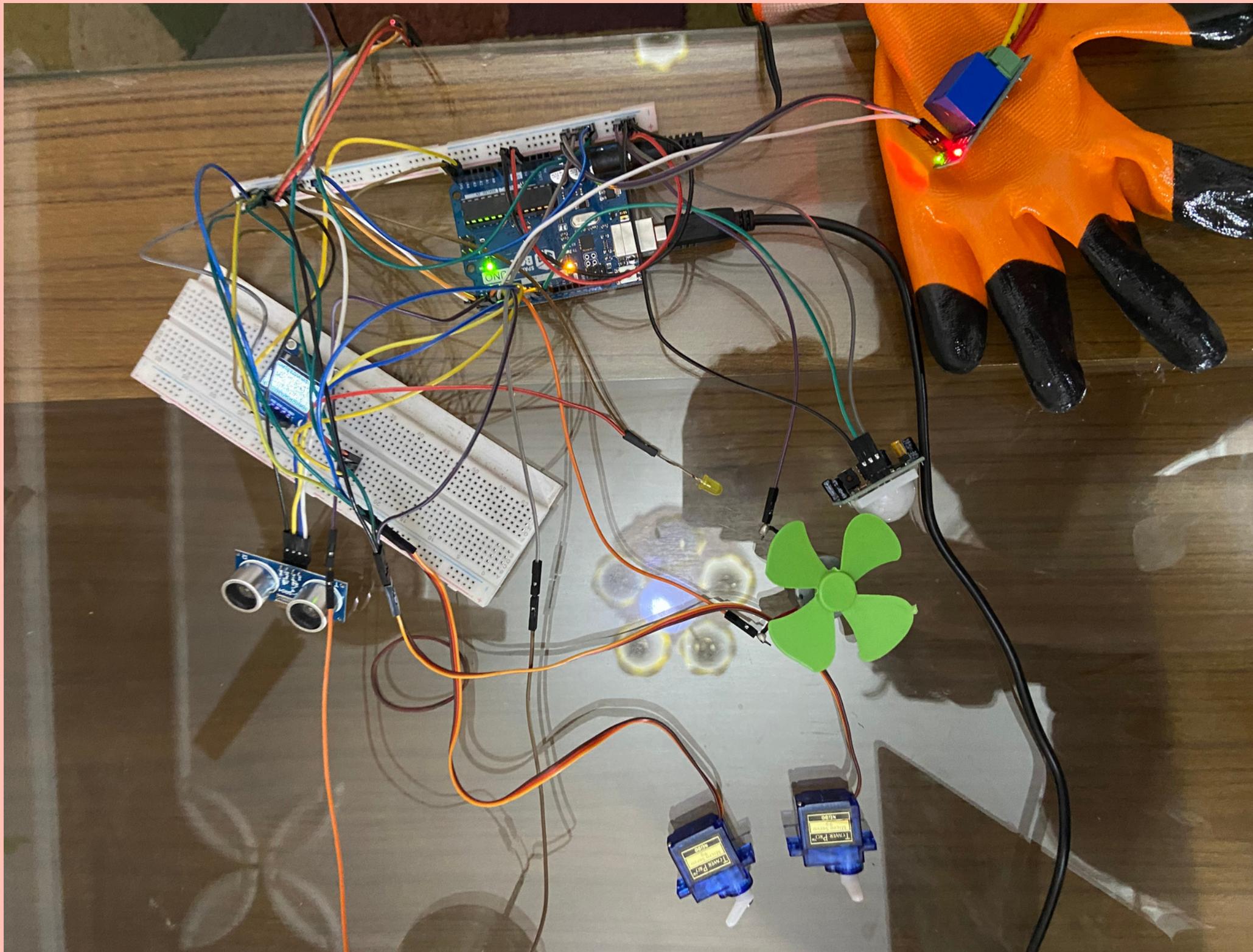
if (state == LOW) {
    Serial.println("Motion detected!");
    state = HIGH;
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Motion Detected");
    delay(500);
    display.clearDisplay();
}
else {
    delay(500);
    if (state == HIGH){
        Serial.println("Motion stopped!");
        state = LOW;
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.setCursor(0,0);
        display.println("Motion Stopped");
        delay(500);
        display.clearDisplay();
    }
}
}
```



REAL-LIFE VIEW OF THE APP- AUTO ACCESS



REAL MODEL



THANK YOU

Created and conceptualized by:

- Kunal Chauhan

