# PROJECT REPORT IN COMPUTER

# DUCKY HOME


**Submitted by**

Autthaphan Poonsawat M.5/2 no.5
Thanawat sophunpanchuen M.5/2 no.16
Chitsanupong Khadpool M.5/2 no.30
Kamolwan Sapsupasiri M.5/2 no.51

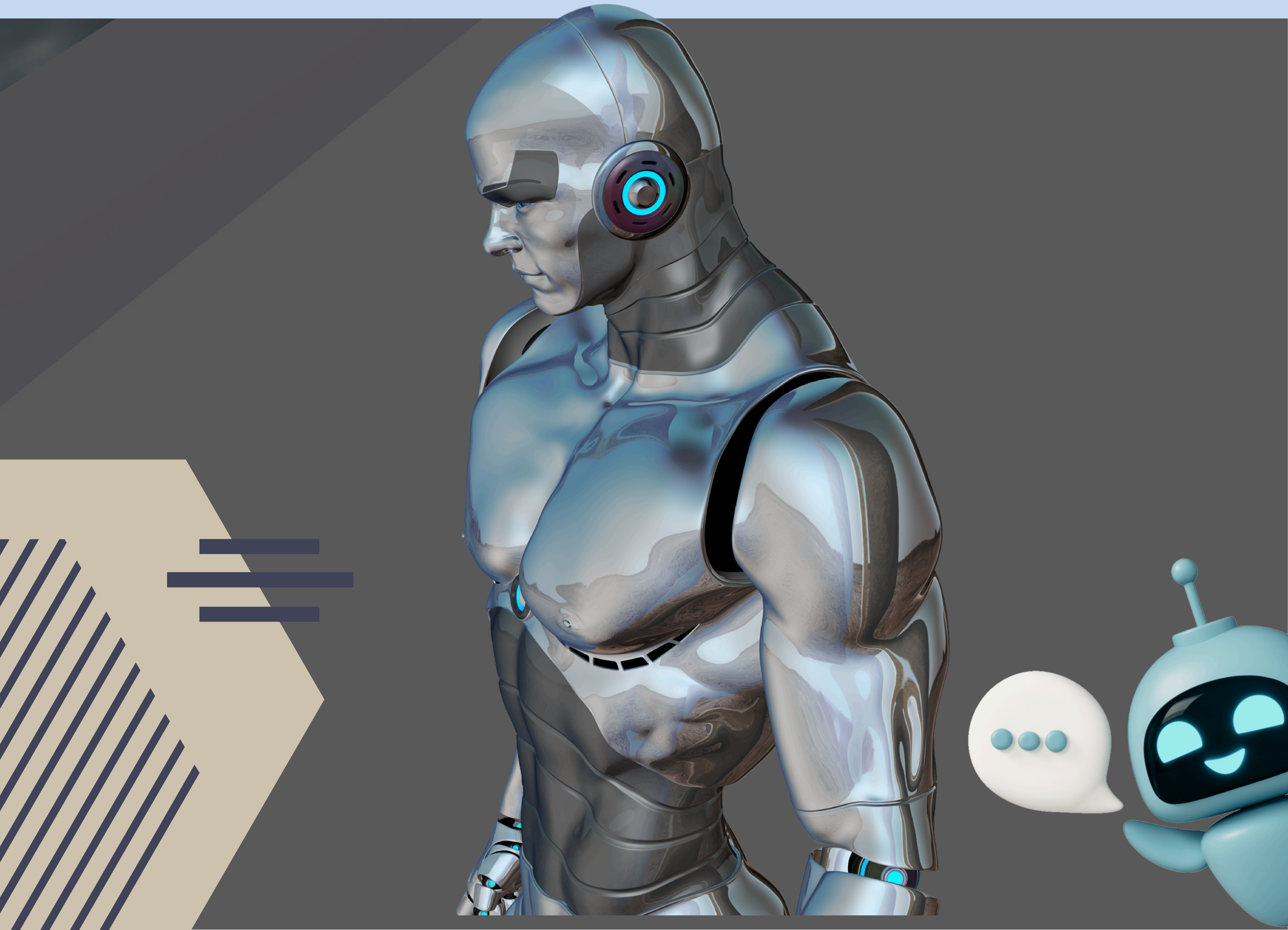
Submitted to:

Master Ronnie Molina
Master Athipan Thoongsook

# TABLE OF CONTENTS

# INTRODUCTION

A modern smart home seamlessly integrates advanced technology to provide enhanced convenience, security, and energy efficiency. Key features include soil moisture sensors that monitor garden conditions and send alerts or automatically activate irrigation systems when watering is required. Smart doors offer added convenience by automatically opening from the inside, while robust security systems enable remote locking and monitoring. Lights, fans, and other appliances can be controlled through smartphone apps or voice commands, giving homeowners the power to manage their homes effortlessly, whether at home or away.



# OBJECTIVE

It sounds like you're outlining some goals or areas of focus. Here's how you might structure them for clarity and direction:

1. Develop and Study Programming Code

Dedicate time daily or weekly to write, debug, and review code.

Work on small projects or algorithms to improve skills.

Explore multiple programming languages or frameworks depending on your interests and goals.

2. Extend Knowledge Through Continuous Learning

Take online courses (e.g., from platforms like Coursera, Udemy, or freeCodeCamp).

Read books, articles, or technical blogs.

Follow forums or communities like Stack Overflow or GitHub for real-world problem-solving discussions.

Stay updated on tech trends by reading newsletters or attending webinars.

3. Learn Development Skills to Apply in the Future

Focus on practical, in-demand skills such as software development, mobile app creation, web development, or game design.

Build projects that reflect real-world applications or portfolio pieces.

Collaborate with peers or contribute to open-source projects to gain team-based development experience.

4. Simulate Smart Home Convenience Systems

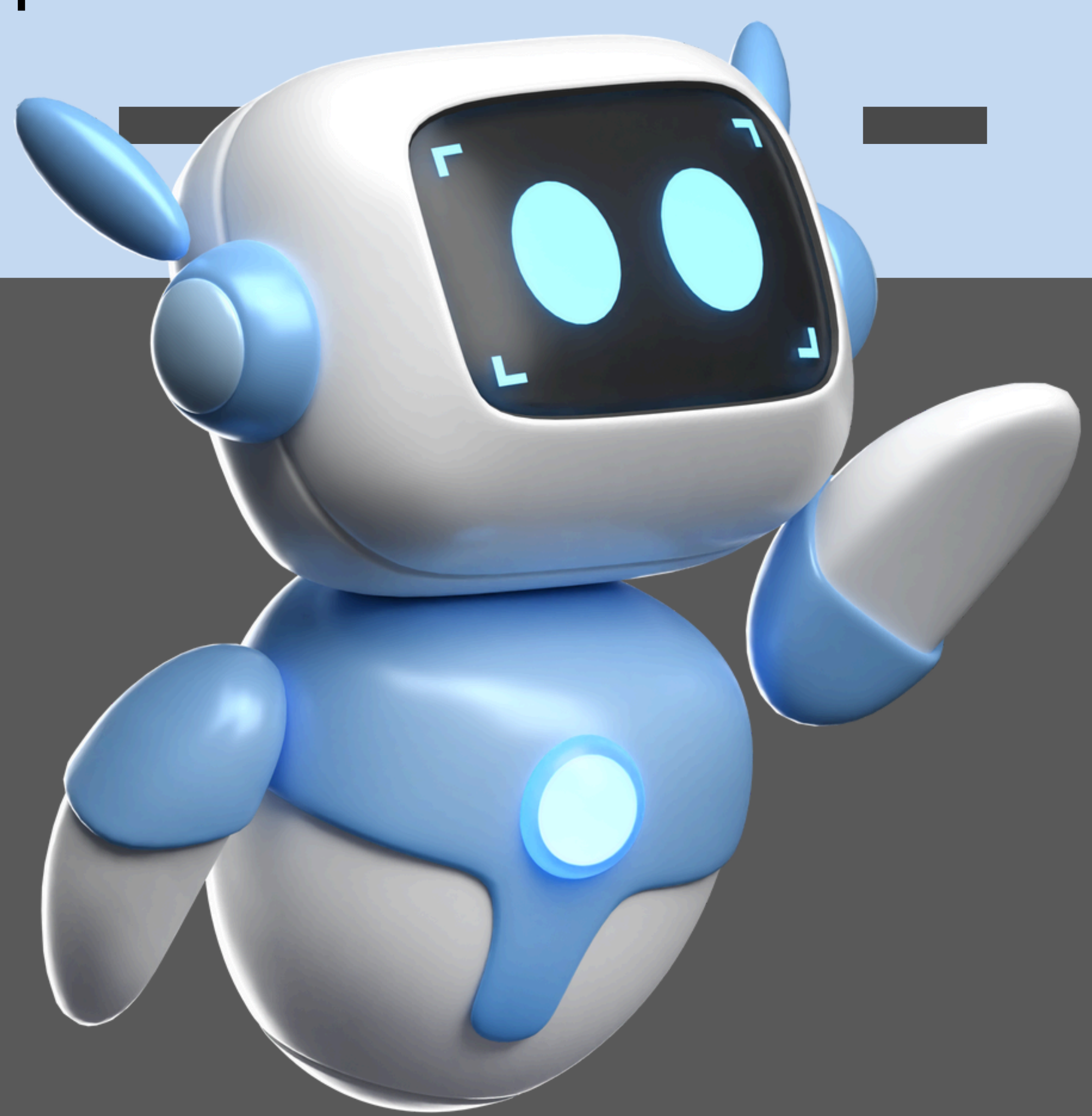
Learn about IoT (Internet of Things) frameworks and protocols (e.g., MQTT, Zigbee).

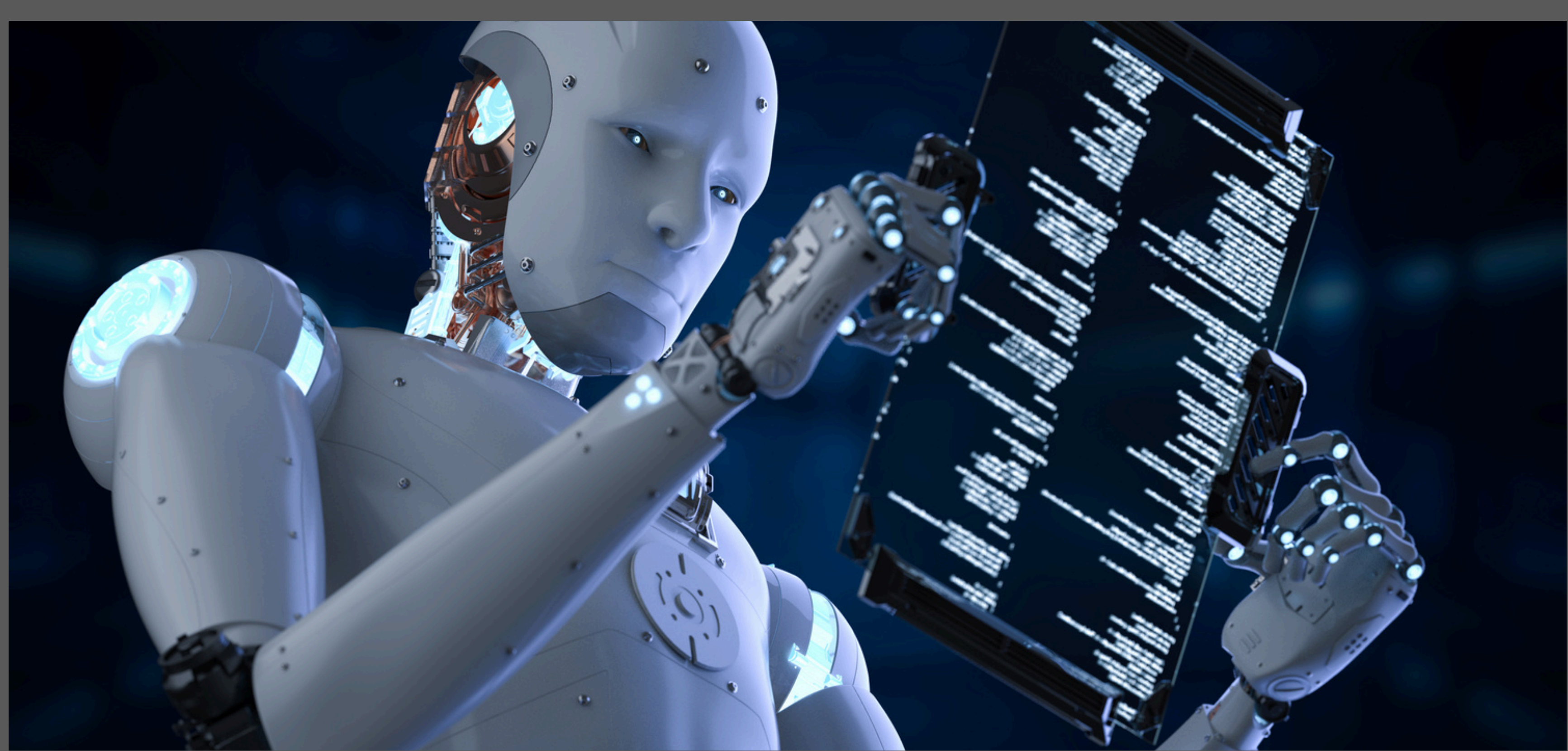
Experiment with hardware like Raspberry Pi, Arduino, or other microcontrollers.

Develop basic control applications, such as light automation, temperature monitoring, or voice assistant integrations.

Study security considerations for IoT devices.

Would you like detailed resources, example projects, or further assistance with one of these points?





# KEY COMPONENTS

## Sensor

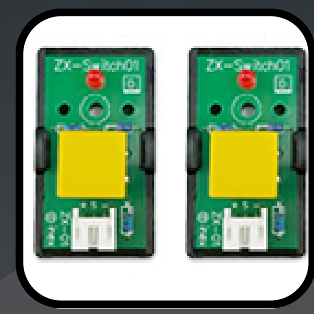
**-Switch Sensor**
Used to open the door manually from outside or inside the house.
**-Servo Motor**
Used to implement an automated door opening and --closing system.
**-Ultrasonic Sensor**
Detects objects to trigger the automatic opening of the door.
**-Soil Moisture Sensor**
Measures the moisture level of the soil in plants and sends a signal to an OLED screen to notify when watering is needed.

## ACTUATORS

**-Buzzer**
Emits an alert sound when the soil lacks sufficient moisture.
**-Neopixel**
Provides lighting inside the house.

## CONTROL UNIT

Used to program and control the operation of the house
For programming and managing the functionality of all systems in the home.

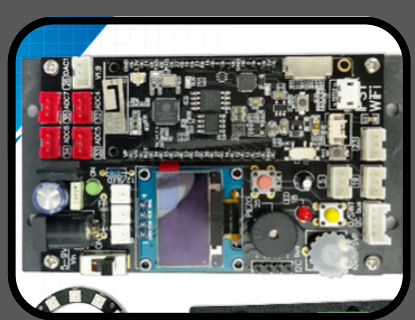


Switch Sensor

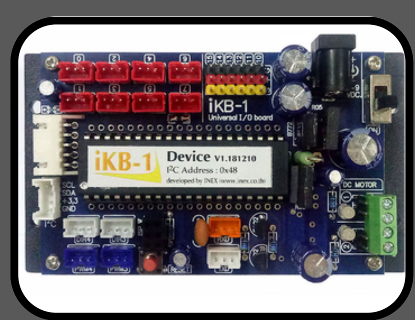

ultrasonic

Servo

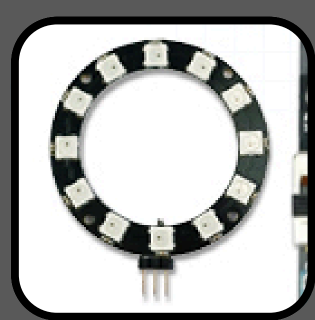
Soil Moisture Sensor



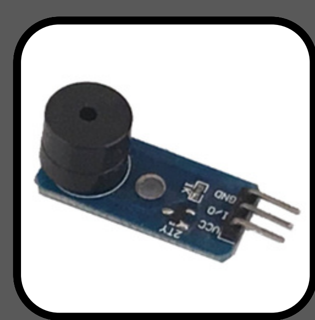
IPST-BOARD



ikb 1



Neopixel



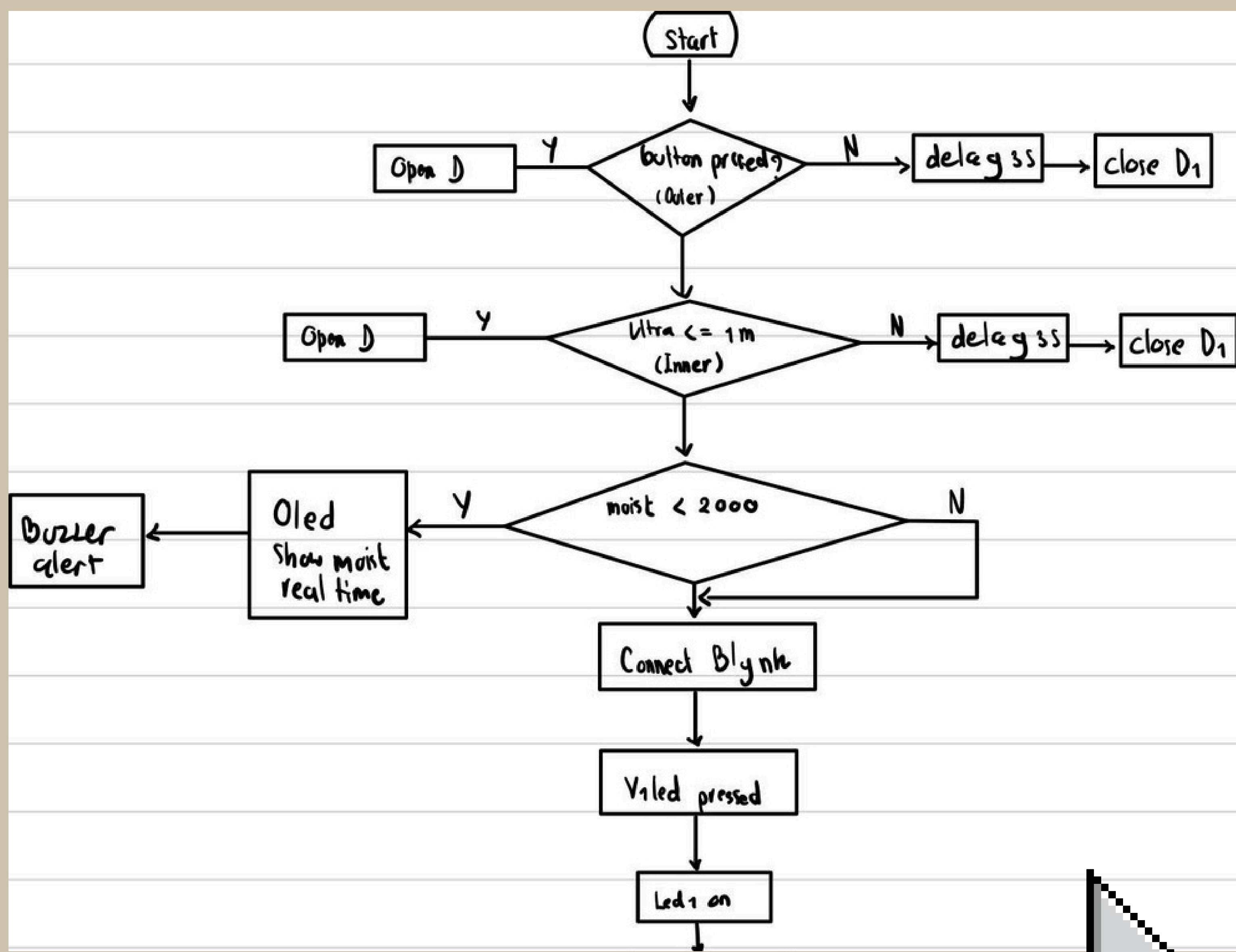
Buzzer

# SYSTEM ARCHITECTURE

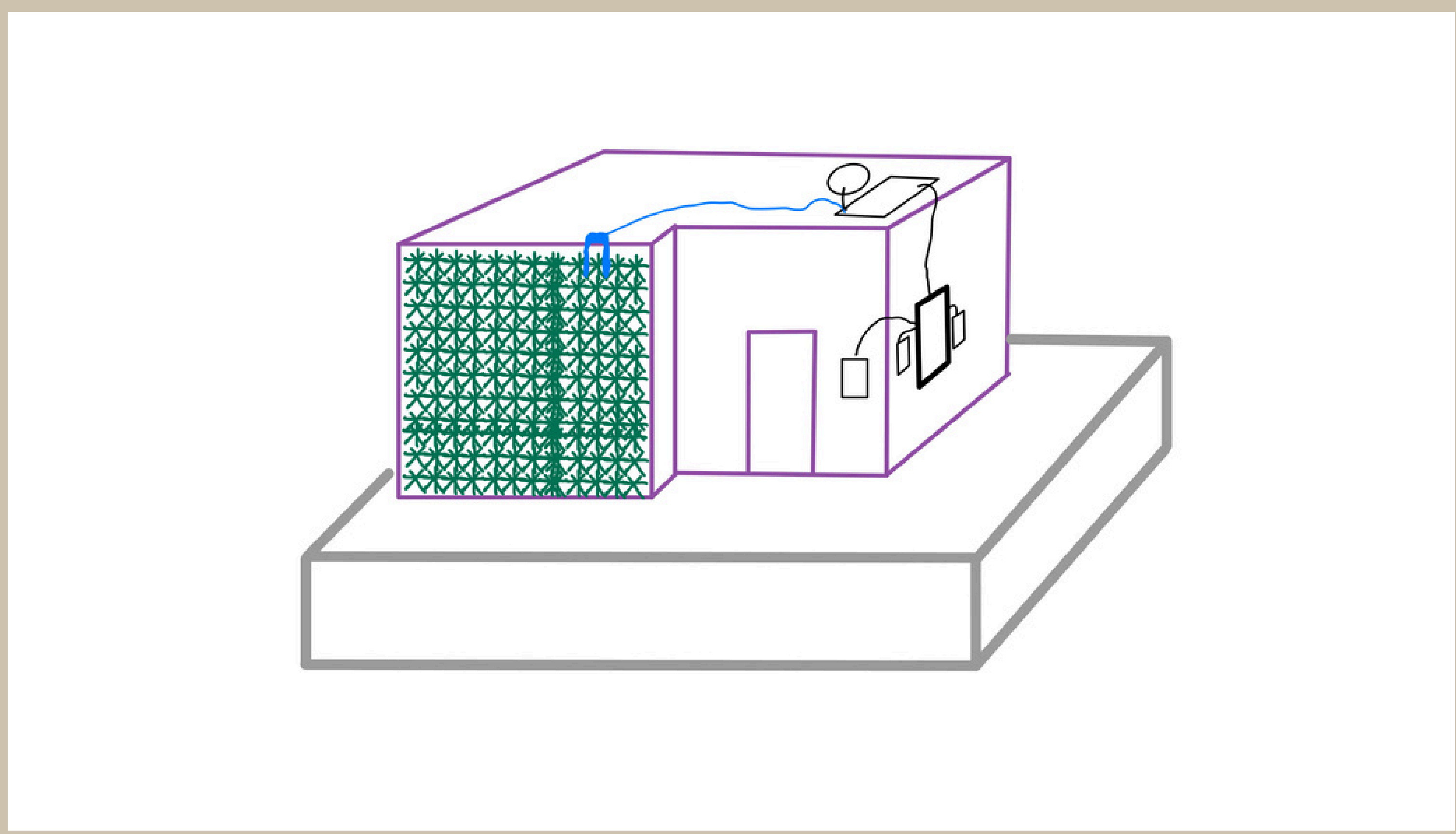
## DESCRIPTION OF SYSTEM FLOW

1. System Initialization: The process starts with the system being activated.
2. Button Press Check:
If the button is pressed, Door D1 opens.
If not, a delay of 3 seconds occurs, and Door D1 is closed.
3. Proximity Sensor Check (Ultrasound ≤ 1m):
If an object is detected within 1 meter, Door D1 remains open.
If no object is detected, a delay of 3 seconds occurs, and Door D1 is closed.
4. Moisture Level Check (Moisture < 2000):
If moisture is below the threshold, the OLED display shows real-time moisture data.
If moisture exceeds the threshold, a buzzer alert is triggered.
5. Blynk Connection: The system connects to the Blynk platform for remote interaction.
6. Virtual LED Button Press Check:
If the virtual LED button is pressed through Blynk, LED1 turns on, and the OLED displays the LED status.
This process integrates sensor data, user inputs, and remote monitoring to manage doors, display data, and control LEDs.

## SYSTEM ARCHITECTURE DIAGRAM




# METHADOLOGY



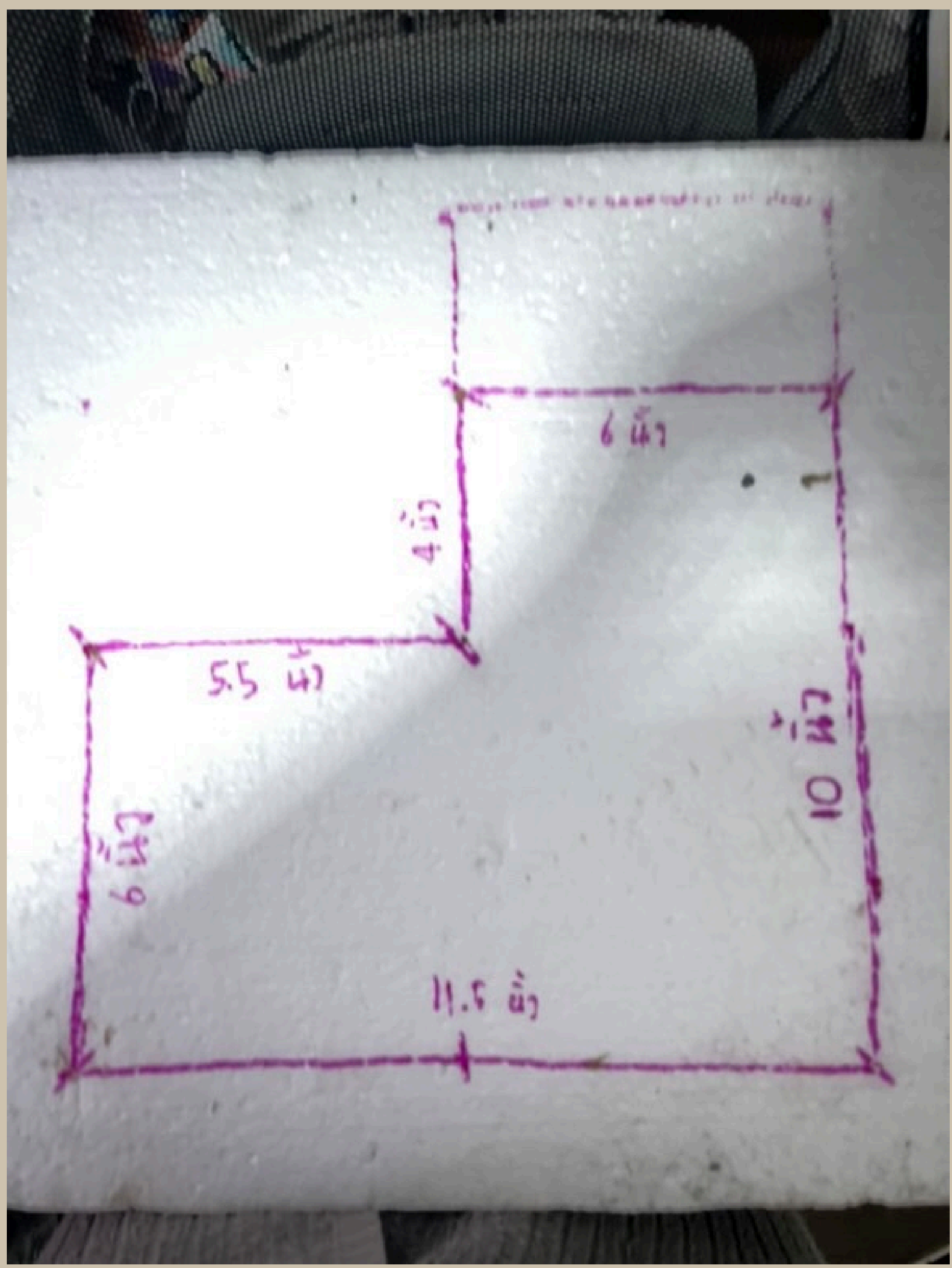
## Collecting the Components

The team has collected various sensors in the irover box.There are boards that control the system: Ikb1 and Ipst boards. There are various sensors such as a buzzer that sends out an alarm, a soil measure that measures humidity in the soil, a servo motor that opens and closes the door, and a neopixel that emits light . All of this is displayed on an OLED display that shows both normal

## Planning the layout

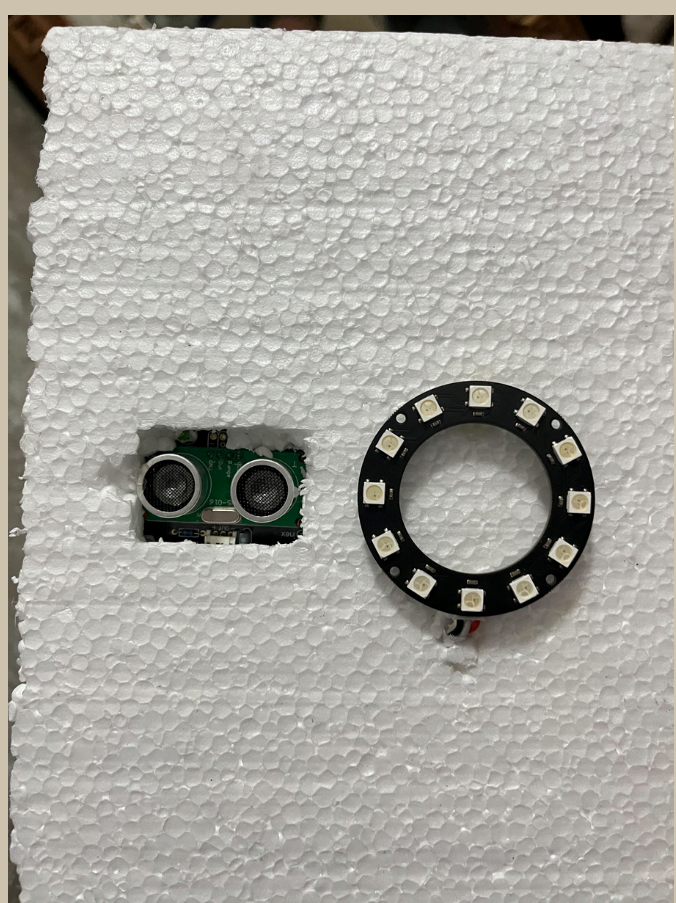
The team planned to build a smart homethe top is a board that controls all the sensors. There will be sensors around the interior of the house and there will be an OLED display to show various information of the house.

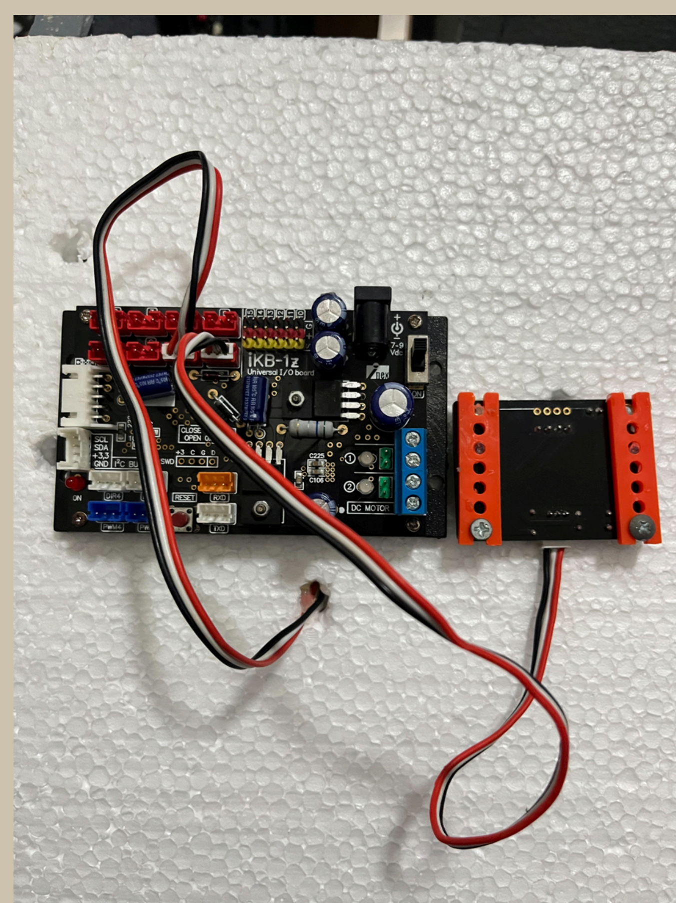
# METHADOLOGY



Start by sketching the design and measuring the dimensions of the components. Mark the cutting points on the foam sheet and cut the foam into various parts to construct the structure. Assemble the pieces to build a modern-style house, designed to meet the lifestyle demands of the current era.

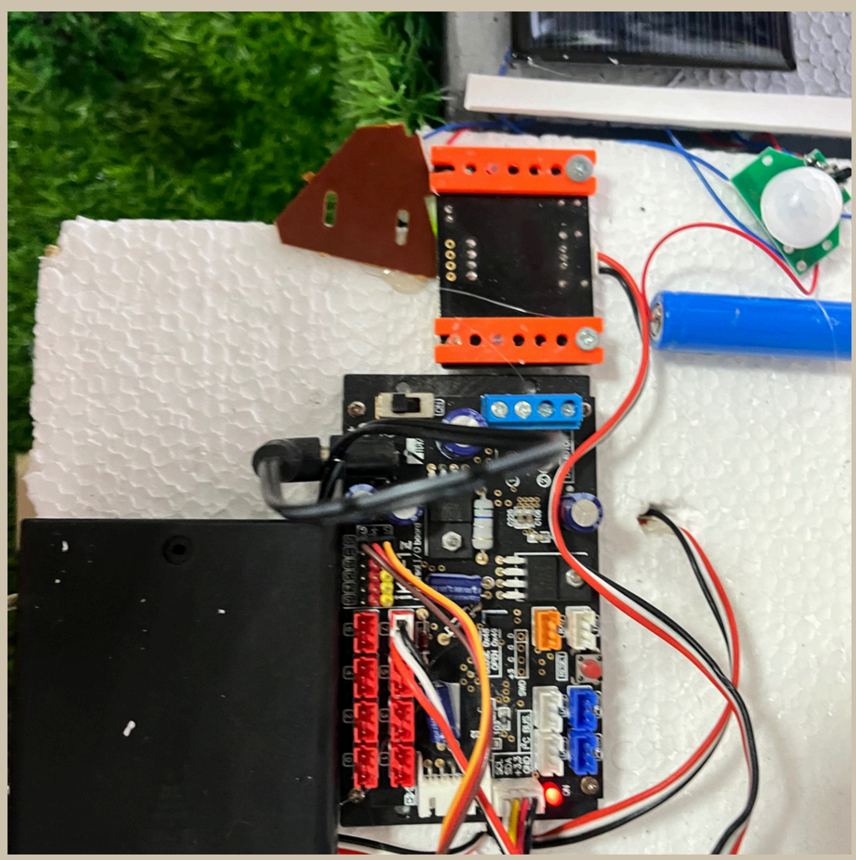
For the installation of equipment, begin by attaching the door to the servo motor. Place the IPST-board and IKB1 in position. Next, install the switch sensor, ultrasonic sensor, soil moisture sensor, and Neopixel. Finally, connect the wires to the control board for the complete setup.

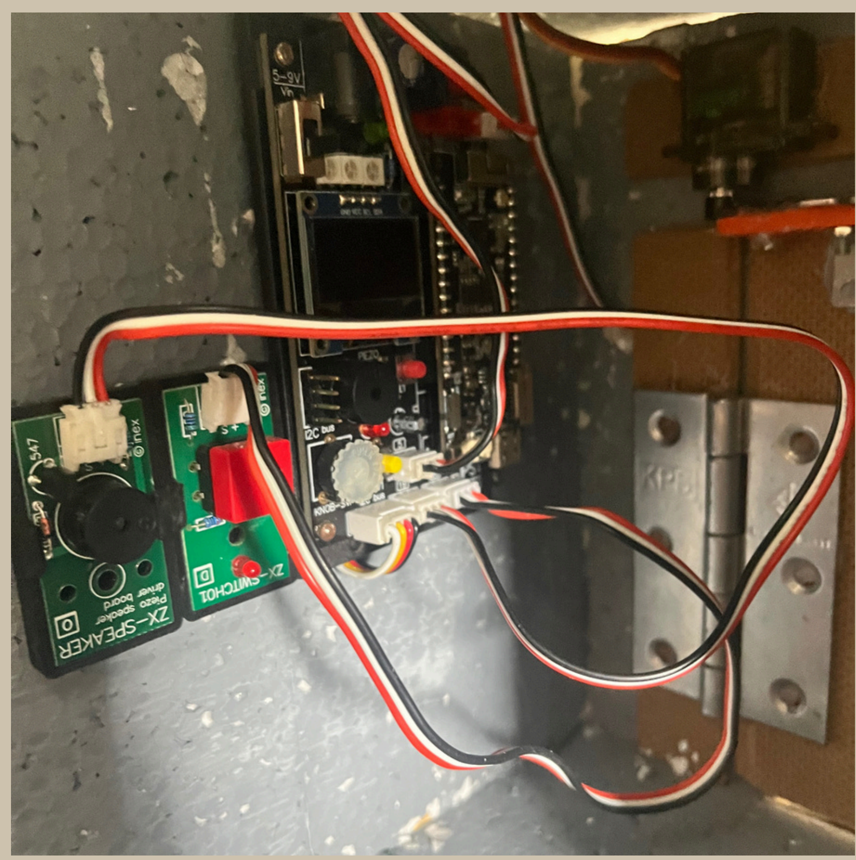




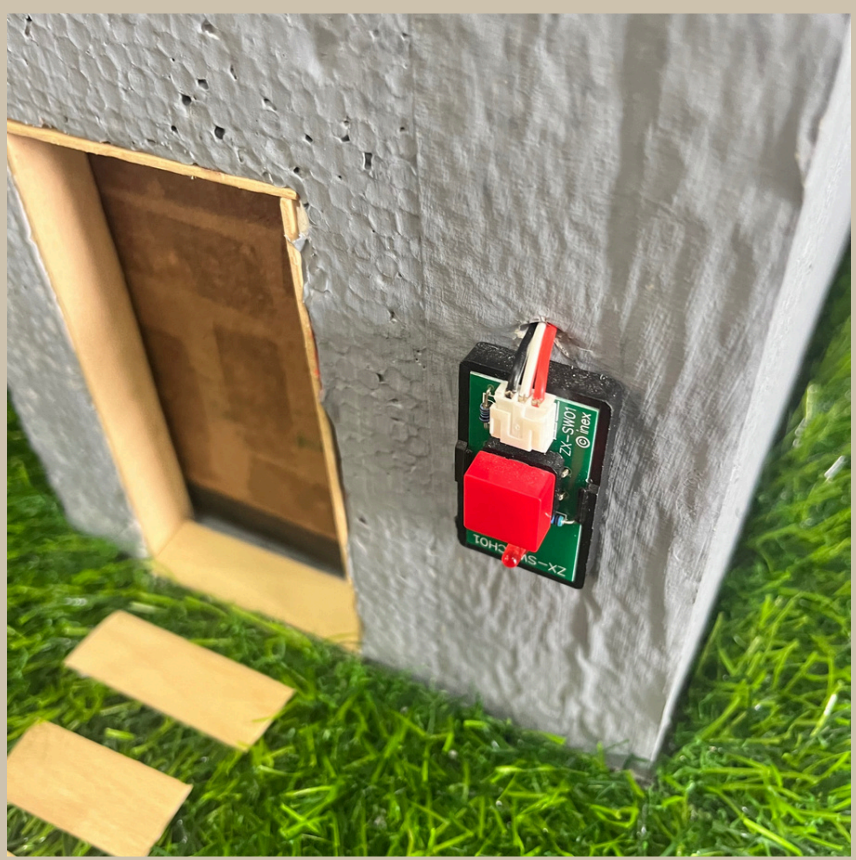
# METHADOLOGY

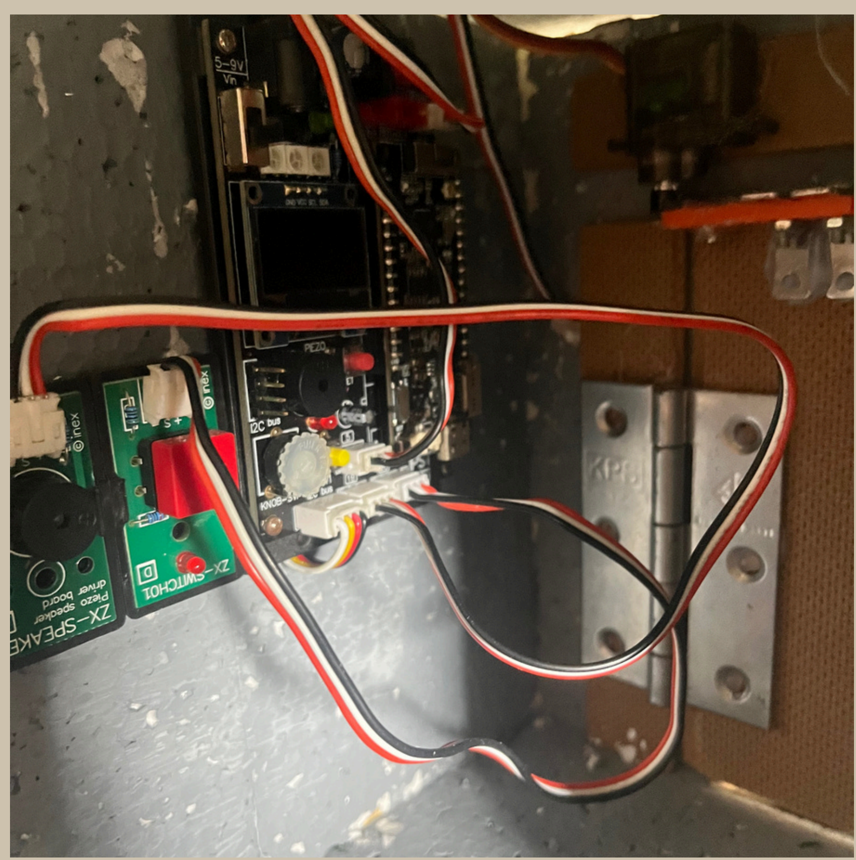
• IPST-board and IKB1: Used to control the house



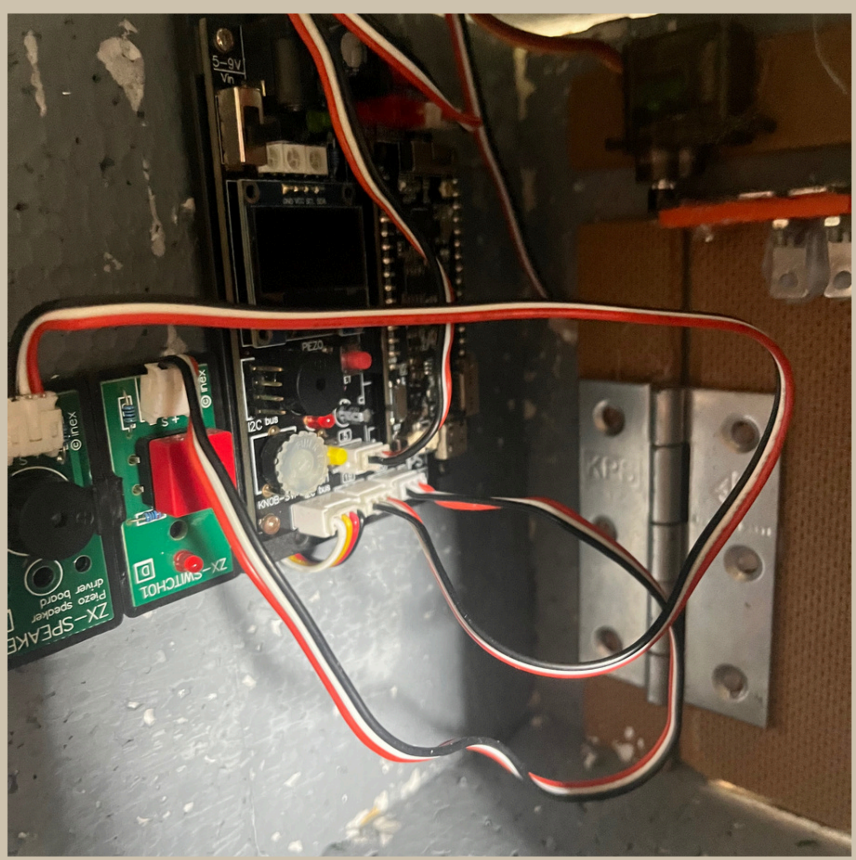


• Switch Sensor: Used to open the house door.
Working Principle: When the switch is pressed, the door will open



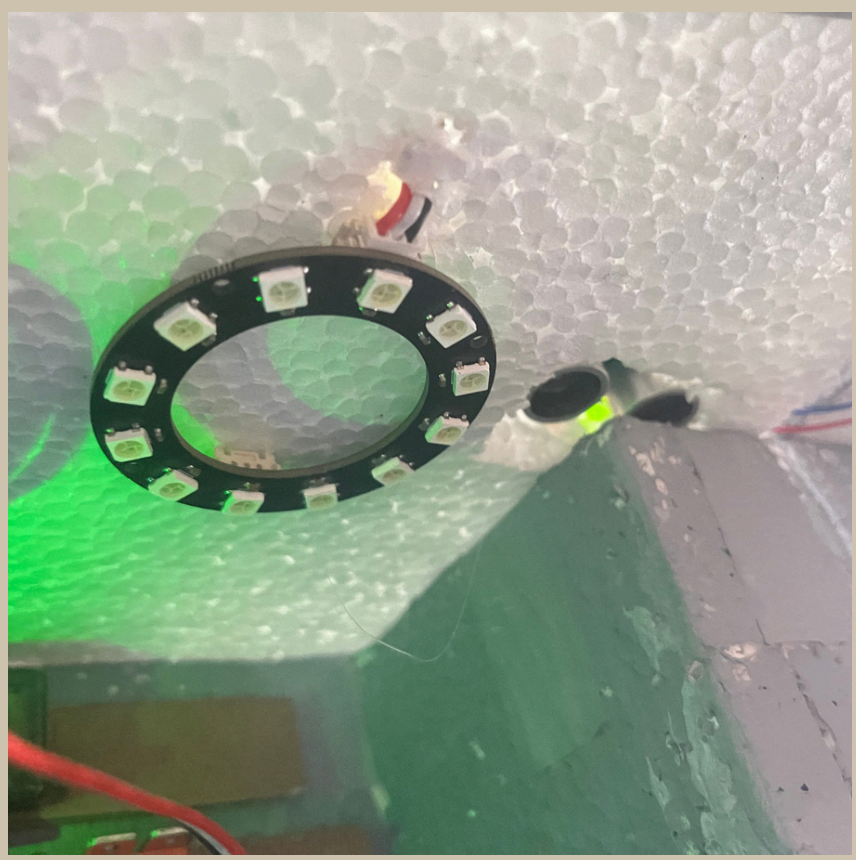


• Servo Motor: Positioned at the front door to control its opening and closing. The servo motor rotates to open and close the door, working together with the switch.



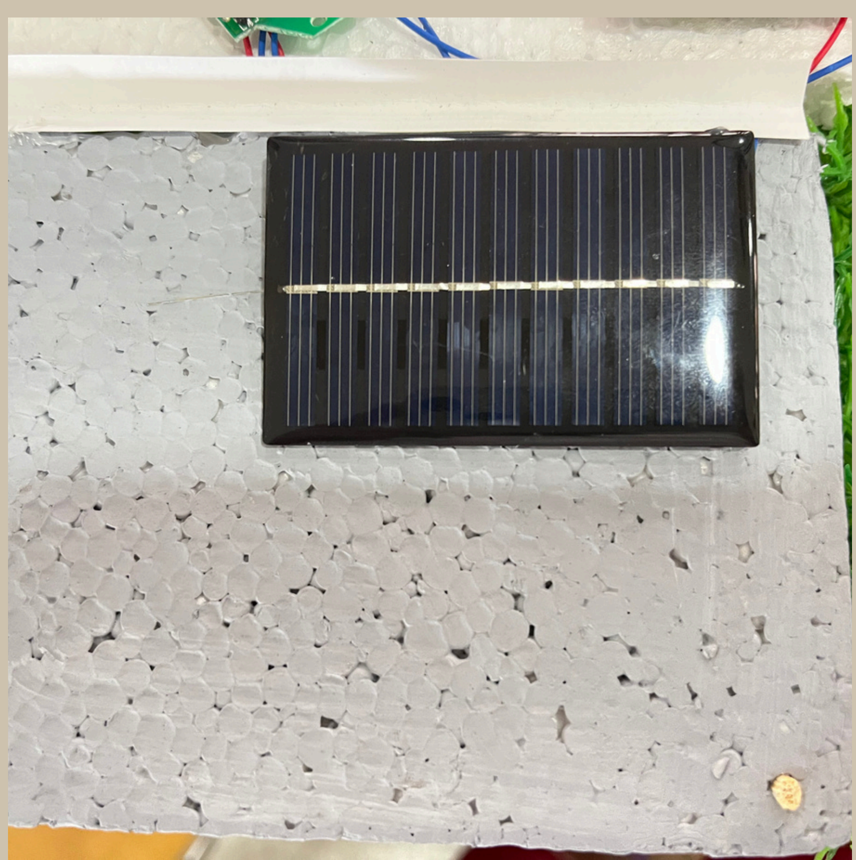
# METHADOLOGY

• Neopixel: Used to display lights. When the door opens, the lights will turn on as well.



Next is to make the roof of the house by using bagasse paper to cut into the shape of the house and use the roof to cover the wires.

instal solar cell to use light auto.



# Code

```
//LIBARIES
#include <IROVER.h>
#include <WiFi.h>
#include <Adafruit_NeoPixel.h>

// Wi-Fi
const char* ssid = "Prem_2.4G";
const char* password = "88888888";

//VARIABLES
#define PIN i5
#define NUMPIXELS 12
#define button1 i4
#define button2 i6
#define buttons (button1,button2)
#define ultra i7
#define servo_motor 10
#define moist_sensor i2
#define buzzer i3
#define led1 i1
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

int centerX = SCREEN_WIDTH / 2;
int centerY = SCREEN_HEIGHT / 2;

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
WiFiServer server(80);

bool Open;
bool obs;

void drawSmileyFace() {
  // Clear the display first
  oled.clearDisplay();
  oled.write(2,0,"SMILE!!");
  oled.show();

}

void setup() {
  init(0x48);
  Serial.begin(115200);
  pixels.begin();
  pixels.setBrightness(10);
  for (int a = 0; a < NUMPIXELS; a++) {
    pixels.setPixelColor(a, pixels.Color(10, 50, 130));
  }
  delay(3);
  pixels.clear();
  pixels.show();
```

```
  }
  delay(3);
  pixels.clear();
  pixels.show();
  delay(100);
  oled.clearDisplay();
  pinMode(led1,OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(button1,INPUT);
  pinMode(button2,INPUT);
  pinMode(buttons,INPUT);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected!");

  // Start the server
  server.begin();
  Serial.println("Server started");
  Serial.println(WiFi.localIP()); // Print IP address
}

void loop() {

float moist = analogRead(moist_sensor);

delay(500);
//DOOR VAIRIABLES//
bool DoorC = false;
int i;
if (i == 0){
  DoorC;
}
if (i > 0){
  !DoorC;
}
//MAIN CODE//
  //OUTER
  if(!Open && button1){//OPEN
    for (i=0;i<=60;i++){
      servo(i,servo_motor);
      delay(5);
    }
    if (!obs && Open){//CLOSE
      for (i=60;i<=0;i--){
        servo(i,servo_motor);
        delay(5);
```

```
        servo(i,servo_motor);
        delay(5);
      }
    }

  }//INNER DOOR
  if(!Open && obs){//OPEN
    for(i=0;i<60;i++){
      servo(i,servo_motor);
      delay(5);
    }
  }
    if(Open && !obs){//CLOSE
    for(i=60;i<0;i--){
      servo(i,servo_motor);
      delay(5);
    }
  }
    oled.clearDisplay();
    oled.textSize(1);
    oled.textColor(WHITE);
    oled.print(moist_amount);
  if(moist < 2000){
    tone(buzzer,1000,5);
  }

  //Webserver
   WiFiClient client = server.available(); // Check for incoming clients
  if (client) {
    Serial.println("New Client.");
    String request = client.readStringUntil('\r');
    Serial.println(request);
    client.flush();

    // Parse the color from the request
    if (request.indexOf("/color?hex=") != -1) {
      int idx = request.indexOf("hex=") + 4;
      String hexColor = request.substring(idx, idx + 6);

      // Convert HEX to RGB
      long number = strtol(&hexColor[0], NULL, 16);
      int r = (number >> 16) & 0xFF;
      int g = (number >> 8) & 0xFF;
      int b = number & 0xFF;

      // Set NeoPixel color
      for (int i = 0; i < NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(r, g, b));
      }
      pixels.show();
    }
```

```
    // Send HTTP response
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println();
    client.println("<!DOCTYPE html>");
    client.println("<html>");
    client.println("<body>");
    client.println("<h1>Control NeoPixel Ring</h1>");
    client.println("<input type='color' id='colorPicker' value='#ff0000'>");
    client.println("<script>");
    client.println("document.getElementById('colorPicker').addEventListener('input', function(e) {");
    client.println("  fetch('/color?hex=' + e.target.value.substring(1));");
    client.println("});");
    client.println("</script>");
    client.println("</body>");
    client.println("</html>");

    client.stop();
    Serial.println("Client disconnected.");
  }
}
```

# Conclusion

The integration of smart technology in modern homes significantly enhances convenience, efficiency, and security. By implementing features such as soil moisture sensors, automatic door systems, smartphone-controlled lighting, and OLED status displays, this project successfully demonstrates how automation can improve daily living. Despite challenges like system integration, power management, and security concerns, strategic solutions have ensured a seamless and user-friendly experience. Moving forward, additional improvements such as AI-driven automation, voice control, and backup power solutions will further optimize the system. This smart home project serves as a foundation for future advancements in home automation, making living spaces more intelligent and sustainable.