Table of Contents

TABLE OF FIGURES

INTRODUCTION OF IoT

Figure 1 What is IoT ((Pedamkar, 2022)

One of the most technical definition of Internet of Things (IoT) according to (Gillis, 2022), is a system of interconnected computing devices, mechanical and digital machinery, items, animals, and people with unique IDs and the capacity to send data over a network without the need for human-to-human or human-to-computer contact. The purpose of IoT is, in a nutshell, to expand internet connectivity beyond ordinary devices such as PCs, tablets, and mobile phones to relatively dumb items such as doorknobs, toasters, humidifiers, etc. Almost every gadget may be made "smart" through the Internet of Things, therefore enhancing aspects of our lives. These smart IoT devices are designed with embedded CPUs, sensors, and connectivity to gather and transmit data from diverse surroundings. The process of IoT is typically done in 3 steps: Collect data, Collate, and transfer, data and Analyze data and the action taken. The devices linked to an IoT hub or gateway exchange the data they collect and evaluate locally. These gadgets may link to other devices and act based on the data they receive from another device. These technologies mostly function without human intervention or interaction (Pedamkar, 2022).

INTRODUCTION OF THE PROJECT

Figure 2 Wet and dry waste ((Fonseca, 2017)

According to the World Bank, humans create an astounding quantity of waste: about 2 billion tons every year, or nearly 4.5 trillion pounds. And this number will continue to rise. By 2050, it is estimated that global waste would reach 3.4 billion tons. According to statistics from the World Bank's What a Garbage 2.0 study, dry recyclables such as plastic, paper and cardboard, metal and glass make up 38% of municipal waste. (Clifford, 2021). Only 13.5 % of those dry waste are recycled. Accommodation of these immense volumes of waste without being recycled produces harmful greenhouse gases. Dry waste consists of polymers, wood, metals, glass, and other similar materials which are non-biodegradable (which don't decompose). Wet waste refers to the biological waste produced by foods, which is heavier than dry wastes. This comprises both cooked and uncooked foods, as well as fruits, floral debris, and vegetable peels. Wet wastes are biodegradable (which decomposes).

Recycling is an important answer. It will minimize MSW, since waste separation is a vital component to a successful recycling management in terms of boosting the quality of recyclables, and will optimize incineration (Chen & Lee, 2020 as cited in Zhuanget al., 2008). When trash is separated into important streams, such as wet and dry, it has a greater capacity for recuperation and may thus be reused continually. The portion of wet waste is typically converted into fertilizer, methane gas, or both.

Dry trash may be recycled or repurposed. In spite of the vast variety of mechanical waste isolation techniques available, it is always preferable to isolate the loss at its source. The advantages of doing so are that a greater quality of the material is retained for reuse, meaning that more value may be recovered from the trash.

The purpose of this project is to build a minimum, simple, and easy-to-use separation device for urban households and businesses in order to facilitate trash management.

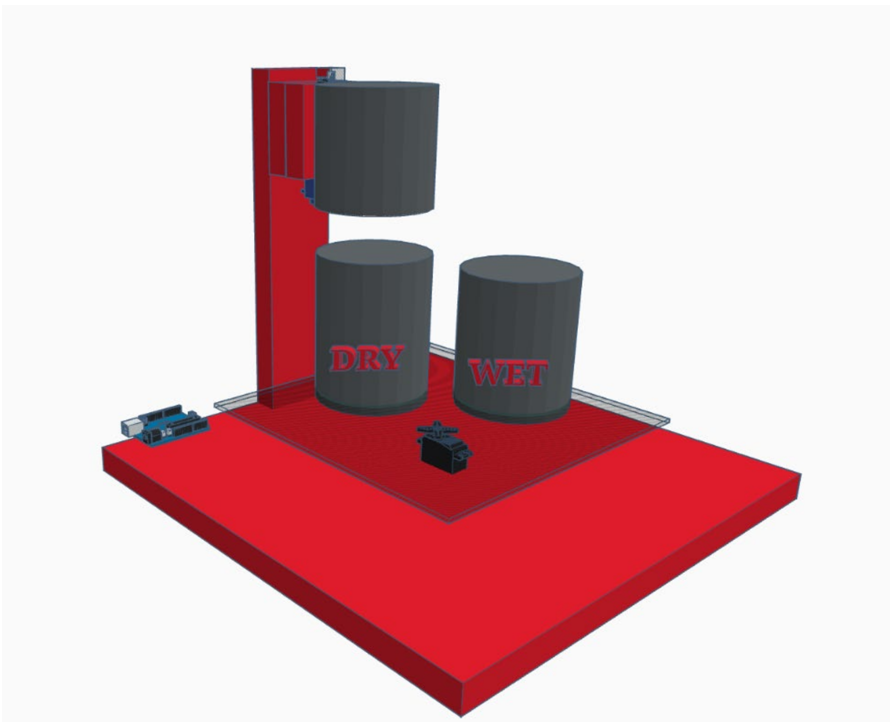

Figure 3 Waste segregator

Objectives:

1. Distinguish wet waste materials and dry waste materials
2. Separate wet waste materials and dry waste materials

LIST OF COMPONENTS

1. Arduino Uno board x 1
2. Bread Board x1
3. MG946R Metal Gear Servo x1
4. SG90 Micro Servo x1
5. Rain Sensor Module x1
6. Infrared Sensor Module x1
7. LCD display 16x2 x1
8. Potentiometer
9. Resistors x 1
10. USB 2.0 Cable Type A/B x1
11. Wood 5mm sheet (50cmx50cm) x1
12. Wood plank (37cm x 8cm x 2.7 cm) x1
13. Wood plank (9cm x 8cm x 2.7 cm) x2

Arduino Uno

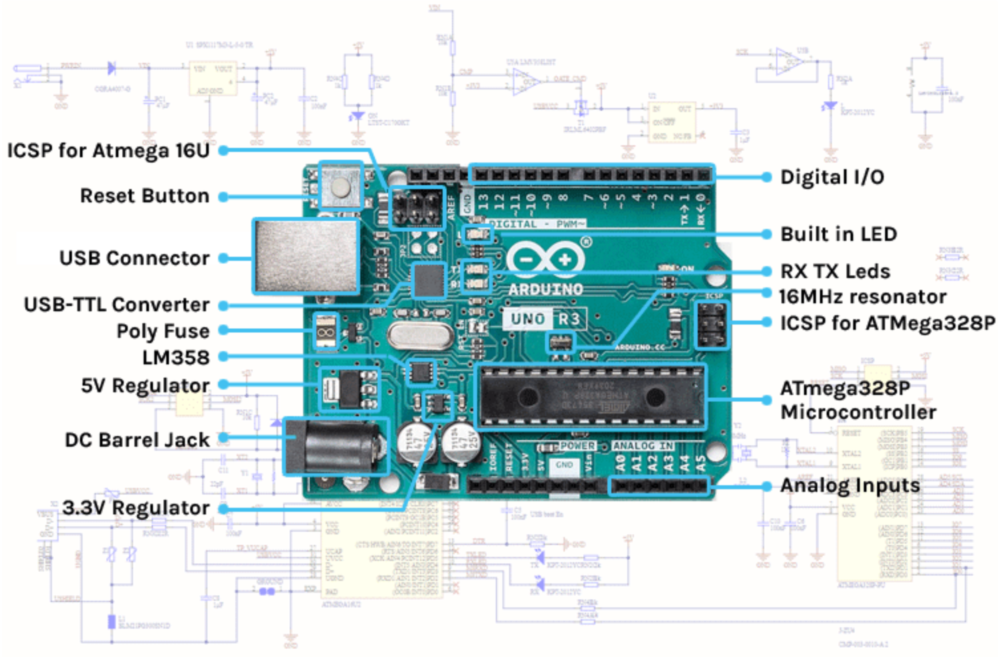

Figure 4 Arduino Uno (Joseph, 2022)

Arduino Uno is a microcontroller installed on a PCB that may be programmed and used for simple everyday chores, mathematical computations, prototyping, and testing. An Arduino development board comprises of the core microcontroller, its auxiliary components, and the appropriate circuitry for communicating with the PC that will be used for both microcontroller communications and programming (Joseph, 2022). Arduino Uno is the main component of this project which is used to program and control the actions of all the following devices connected to this board.

Bread Board

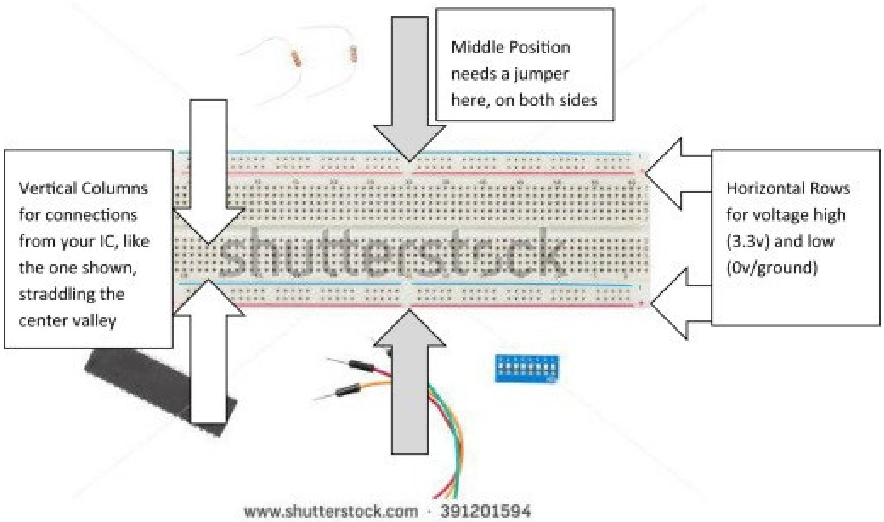

Figure 5 Bread board (Schousek, 2018)

Breadboards are temporary work boards that is used for electronic circuit designs and implementations. This project utilizes of a full-sized breadboard which can be seen from Figure 5 Bread board (Schousek, 2018). In a full-size breadboard, there are (630+2100) = 830 holes. Vertical columns are for connection from IC, and horizontal rows are for voltage high and low.

MG946R Metal Gear Servo

Figure 6 MG946R Metal Gear Servo (www.towerpro.com.tw)

Servo motors are excellent gadgets that may be turned to a certain position. Typically, servo arms may rotate 180 degrees. Using Arduino, servos can be programmed to turn specific angles. A servo motor incorporates a motor, a feedback circuit, and, most significantly, a motor driver. Only one power line, one ground, and one control pin are required. MG946R Metal Gear Servo is used in this project specifically for some of its specification that differs from the mini servo that comes with the Arduino UNO starter kit. This project requires to two bins to be placed on sheet which the MG946R Metal Gear Servo must support. The torque specification of the servo supports 10.3KG/cm with 4.8v supplied. Secondly the, the metal gear type of this servo guaranteed smooth operation of the servo under the load.

SG90 Micro Servo

Figure 7 SG90 Micro Servo (www.diyelectronics.co.za)

The SG90 Micro Servo is utilized as a supplementary servo to open and close the lid with minimal effort. According to the specification, the torque of this servo is 2.5kg/cm at 4.8v. This servo has a maximum rotational range of 180 degrees, 90 degrees to each side.

Rain Sensor Module

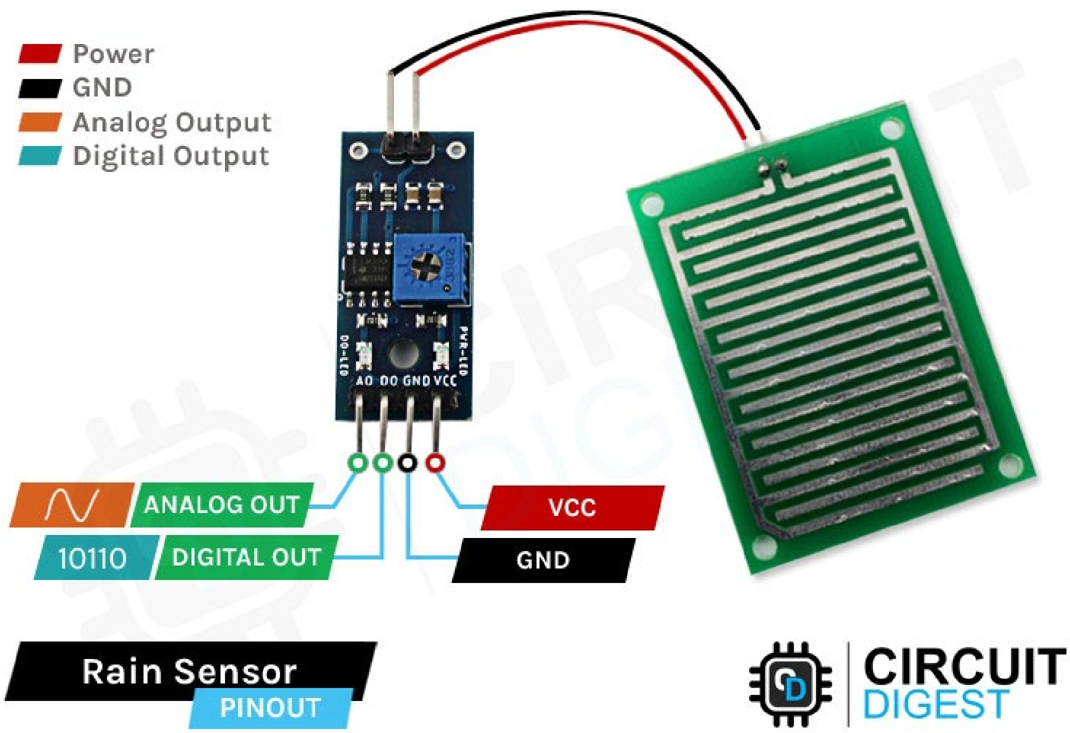

Figure 8 Rain Sensor Module ((Das, 2022)

The raindrop sensor, often known as the rain detector sensor, is a simple device for detecting rainfall. It functions as a switch when raindrops fall on the sensor, and with little code modifications, it can also measure the rainfall's strength. This sensor additionally features a separate indicator light and an integrated potentiometer for adjusting the sensitivity of the sensor's digital output signal. Multiple exposed log conductive plates are placed in a grid pattern to make the PCB. When rain falls on top of the sensor, the resistivity of the conductive plates varies, and by measuring the resistance changes, the intensity of the rainfall may be determined. The greater the rainfall intensity, the lesser the resistance. The rain sensor module was selected for this project due to its uniform PCB shape and larger surface area. The sensor may generate a LOW digital signal when water is detected on the PCB and a HIGH signal when no water is detected. In addition to digital, it may generate an analogue signal that measures the resistance based on the quantity of water on the PCB board. This gives more option to control the resulting output of the system.

Infrared Sensor Module

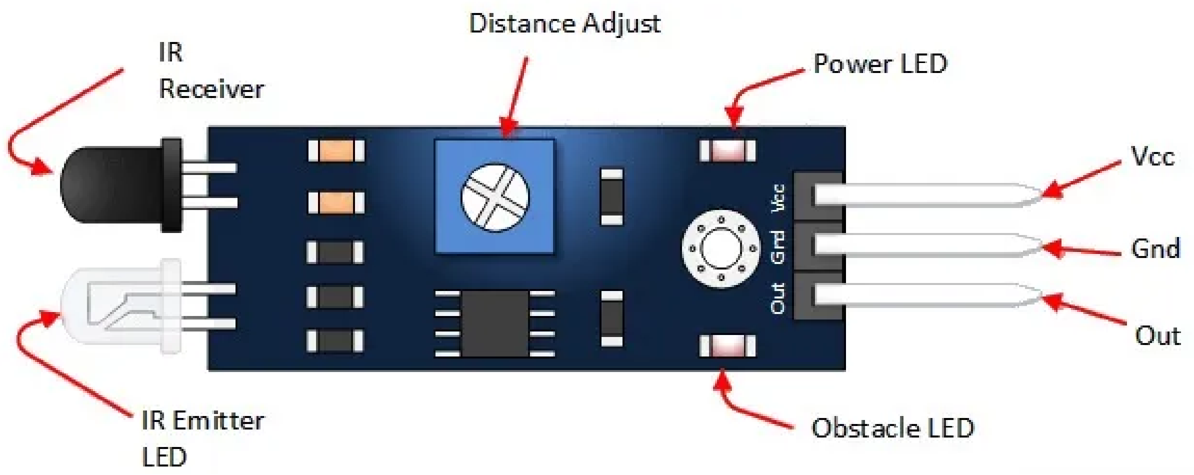

Figure 9 Infrared Sensor Module (IR sensor and Arduino interfacing " Pija education, 2020)

An infrared sensor is an electronic module that can produce and/or detect infrared radiation in order to determine the physical characteristics of its surroundings. IR sensors can also determine the amount of heat radiated by an item and detect motion. In this project, the IR sensor is used to detect whether waste is put to segregate. IR transmitter transmits IR signal, which, upon detecting an obstruction in its route, is reflected by the obstruction, and received by the receiver. The board incorporates a potentiometer for adjusting the distance from which an impediment is sensed. The obstacle LED illuminates when the IR receiver gets the signal, which is a helpful built-in function that eliminates the need for extra equipment or IDE's serial monitor.

LCD display 16x2

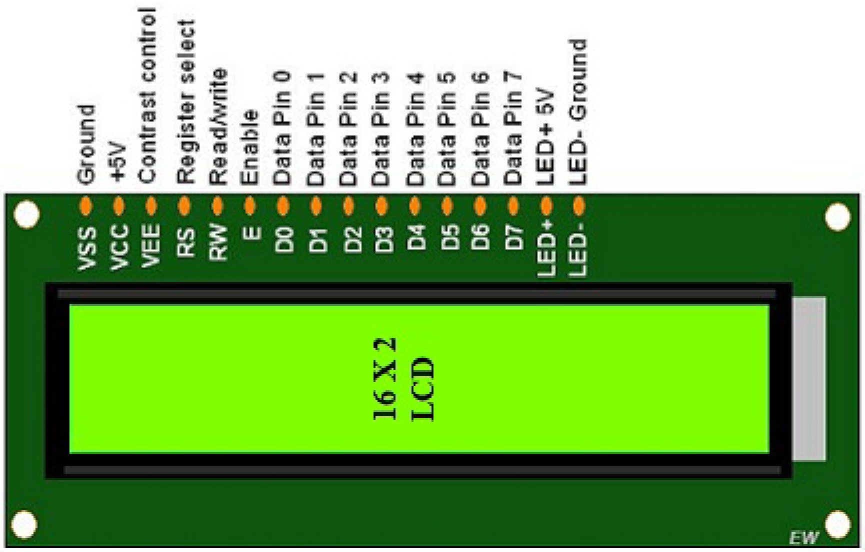

Figure 10 LCD display (LCD 16x2: Pin Configuration, features and its working, 2019)

LCD is an acronym for liquid crystal display. It is one type of electronic display module used in a wide variety of circuits and devices, such as mobile phones, calculators, computers, televisions, etc. These displays are favored for light-emitting diodes with multiple segments and seven segments. The primary advantages of adopting this module are its low cost, ease of programming, animations, and lack of restrictions on displaying special characters. The LCD display used in this project is a 16x2 display which consists of 16 columns and 2 rows.

Resistors

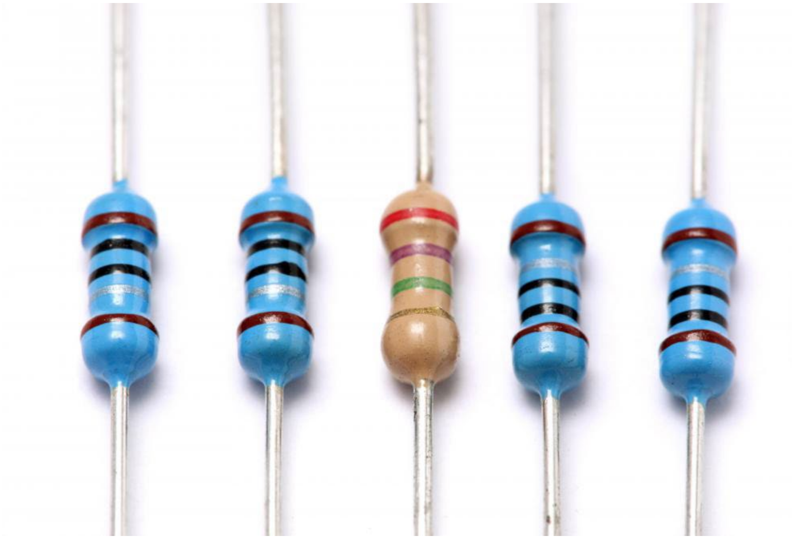

Figure 11 Resistors (www.aboutmechanics.com)

Resistor is a passive electrical component that resists the passage of electric current. They may be found in most electrical networks and electronic circuits. Ohms () are used to quantify resistance. An ohm is the resistance encountered when a current of one ampere (A) flows through a resistor with a voltage drop of one volt (V) between its terminals. Current is proportional to the voltage between the terminals. Ohm's law depicts this proportion as $R = \frac{\mathrm{V}}{\mathrm{I}}$ (What is a resistor?: Resistor fundamentals: Resistor guide, n.d.). Resistors serve several applications. Several examples include restricting electric current, dividing voltage, producing heat, matching, and loading circuits, controlling gain, and setting time constants. They are available commercially with resistance ratings spanning more than nine orders of magnitude. They can be employed as electric brakes to diffuse the kinetic energy of trains or as electrical components smaller than one millimeter square. Resistors can be divided by the functional type and the resistance material. Typically, there are two types of resistors: Variable resistors and fixed resistors. Variable resistors are utilized in potentiometers which is built into the two sensors that the project utilizes (raindrop sensor module and IR sensor module). Resistors with a fixed value have a preset ohmic resistance and are not adjustable. These resistors are the commonly used in electric circuits and electronic components. This project utilizes one resistor which is used in the connection of the display to the power HIGH voltage.

USB 2.0 Cable Type A/B

Figure 12 USB 2.0 cable Type A/B (https://www.datapro.net/products/usb-2-0-a-to-b-device-cable.html)

The 1594 series cable is an A-to-B (host to device) USB cable. This is the most common and widespread USB cable, and it is used for practically every USB printer, scanner, and external enclosure available on the market. Unless a device is meant to have a low profile or a compact form factor, its interface will normally be a USB connection of type B. The same USB A/B cable is compatible with all scanners, printers, and other auxiliary devices. The type 'A' connection is a rectangular plug that connects to downstream port sockets on a USB hub or USB host. The USB 2.0 cable Type A/B is used to connect the Arduino board to the computer in order upload the code from the Arduino IDE.

SYSTEM DESIGN

1. Prepare the breadboard

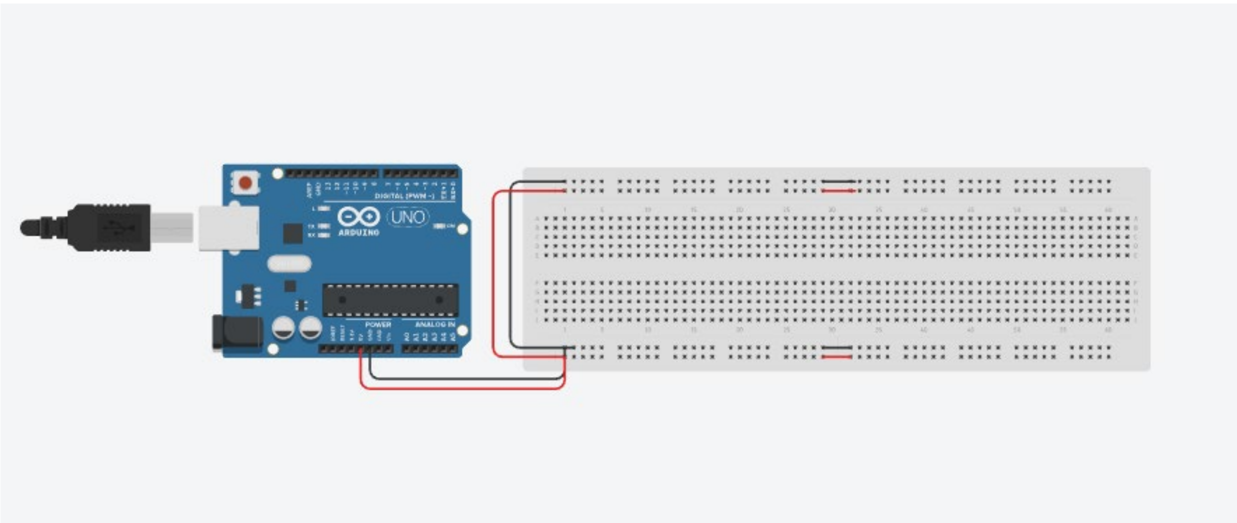

Figure 13 Prepare the breadboard

Breadboard preparation is the initial phase in the circuit design process. As demonstrated, this is accomplished by connecting the Arduino board's 5v HIGH and GND (ground) power supplies across two rows of the breadboard (Figure 13 Prepare the breadboard). This will enable the project's subsequent components to be powered.

2. Rain sensor module/ Soil moisture sensor circuit connection

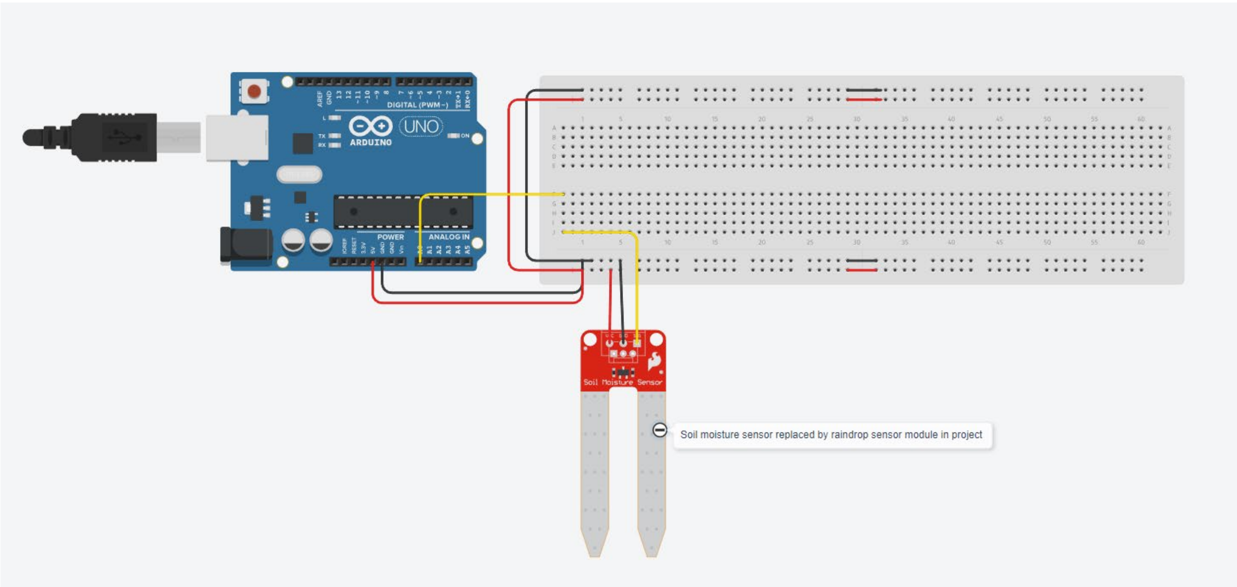

Figure 14 Rain sensor module/ Soil moisture sensor circuit connection

In the second step of the circuit design, the Soil Moisture Sensor is connected to the Arduino board. The VCC terminal on the sensor is connected to the 5v power output pin on the breadboard. GND terminal on the sensor is connected to the 0v GND power output pin on the breadboard. The signal output terminal on the sensor is connected to the Analog pin (A0) on the Arduino board using the breadboard to demonstrate clean wiring management shown in (Figure 14 Rain sensor module/ Soil moisture sensor circuit connection). Instead of a Soil Moisture Sensor, the projects utilizes a Rain Sensor Module which has the same Analog, VCC, and GND connection as the Soil Moisture Sensor shown in (Figure 14 Rain sensor module/ Soil moisture sensor circuit connection).

3. IR sensor/PIR sensor circuit connection

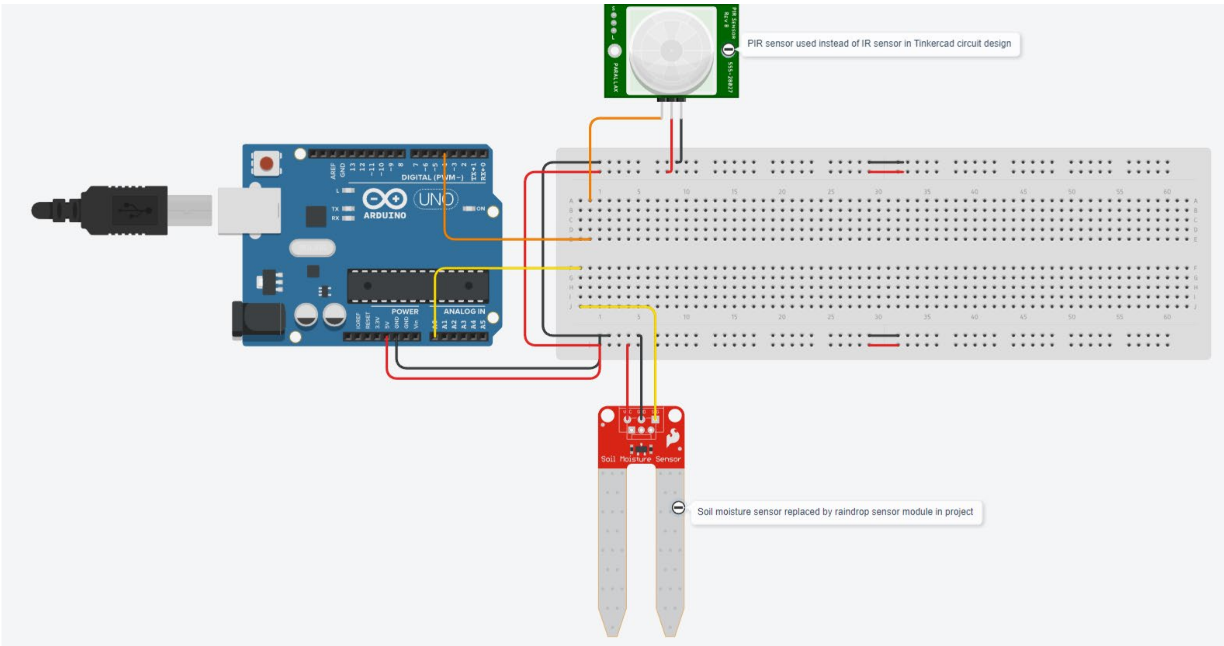

Figure 15 IR sensor/PIR sensor circuit connection

Due to the lack of an IR sensor module in Tinkercad, the passive infrared (PIR) sensor is employed in the circuit design example. Infrared (IR) and passive infrared (PIR) sensors will utilize the same connections. The VCC connection of the sensor is linked to the breadboard's 5v power pin. The sensor's GND connection is linked to the breadboard pin that receives GND power from the Arduino board. The Signal terminal of the sensor is linked to Digital pin 4 on the Arduino board using an orange jumper wire routed from the breadboard to illustrate tidy cable management (Figure 15 IR sensor/PIR sensor circuit connection).

4. MG946R Metal Gear Servo connection

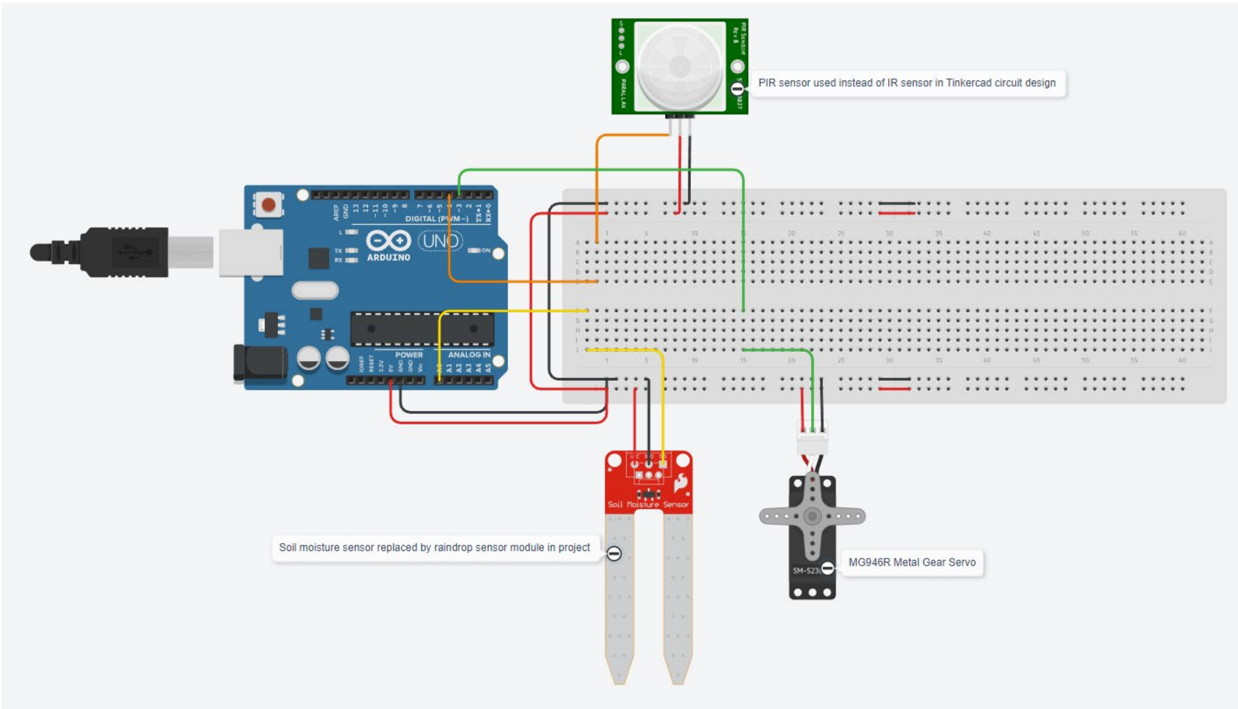

Figure 16 MG946R Metal Gear Servo connection

After the PIR sensor connection, the next component to be connected is the MG946R Metal Gear Servo to the Arduino board. The Power terminal on the servo is directly connected to the horizontal row pin on the breadboard which is supplied with 5v power lead from the Arduino board using a red jumper cable. The ground (GND) terminal of the servo is connected to the horizontal row pin which is supplied with the GND power from the Arduino board. The Signal terminal on the servo is connected to the Digital pin (3) on the Arduino board routed from the breadboard using a green jumper cable demonstrated in the (Figure 16 MG946R Metal Gear Servo connection).

5. SG90 Micro Servo connection

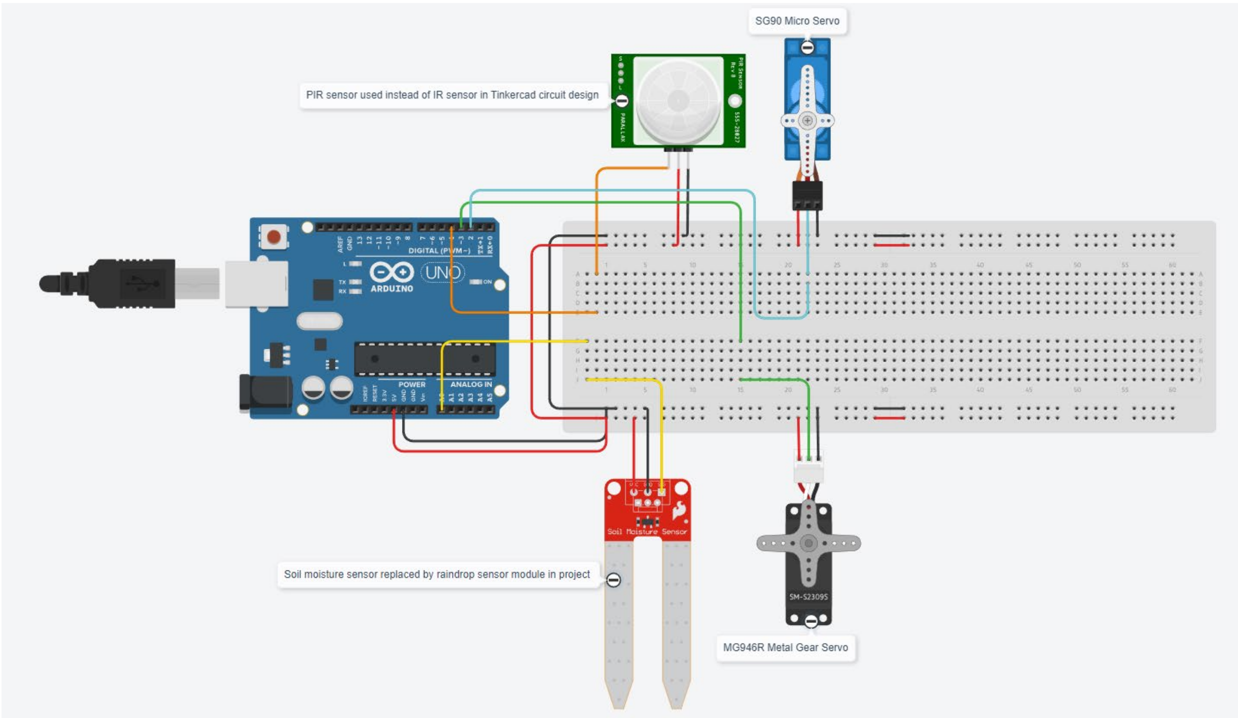

Figure 17 SG90 Micro Servo connection

The SG90 Micro Servo is to be connected as the second servo. The Power connection on the servo is directly linked to the pin on the breadboard's horizontal row that receives 5v from the Arduino board. The ground (GND) connection of the servo is wired to the pin on the breadboard's horizontal row that receives GND power from the Arduino board. The Signal terminal on the servo is connected to the Digital pin (2) on the Arduino board using a turquoise-colored jumper cable routed from the breadboard as seen in (Figure 17 SG90 Micro Servo connection).

6. Potentiometer connection

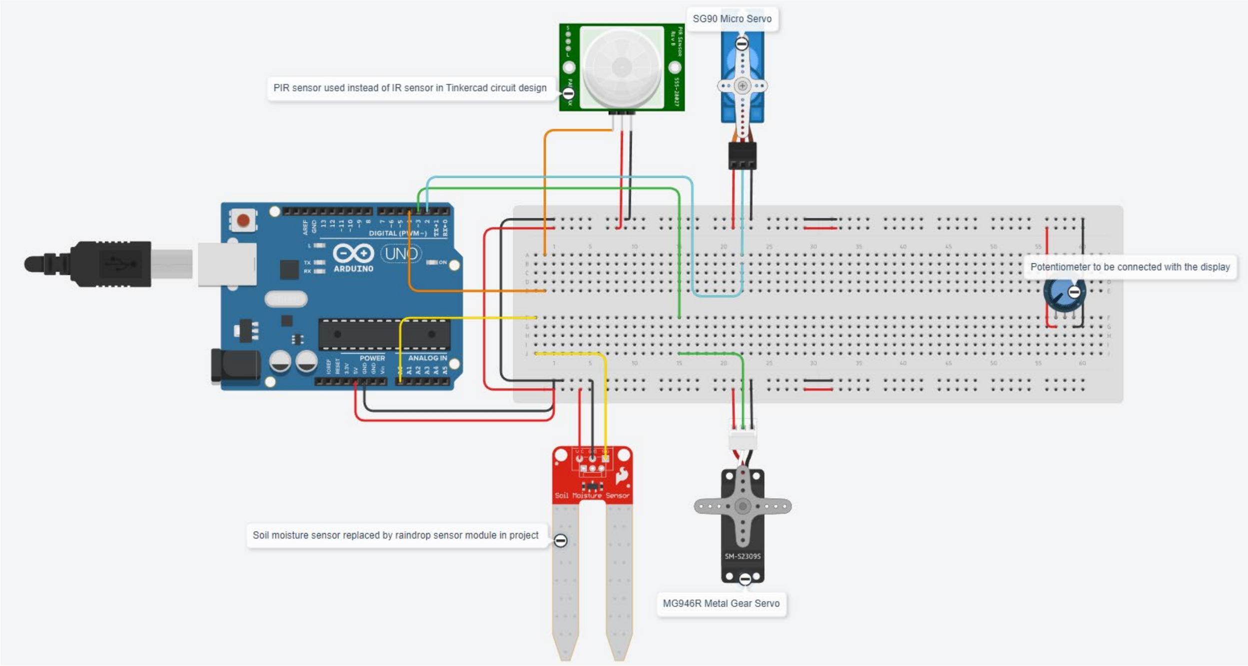

Figure 18 Potentiometer connection

The following step in the circuit is to connect the display, however first the potentiometer must be connected. Terminal 1 of the potentiometer is connected to the pin on the horizontal row of the breadboard using a red jumper cable which is supplied with the 5v power from the Arduino board. Terminal 2 of he potentiometer is connected to the pin on the horizontal row of the breadboard using a black jumper cable which is supplied with GND power from the Arduino. The wiper terminal on the potentiometer is to be connected to the Contrast (VO) terminal on the LCD display.

7. LCD Display connection

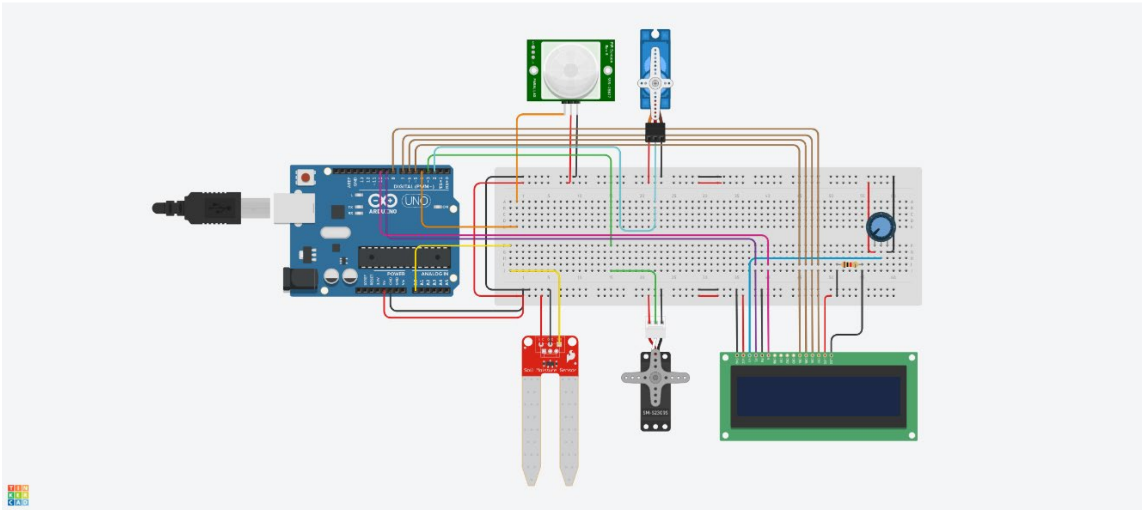

Figure 19 LCD Display connection

After the connection of the potentiometer, the LCD Display is connected to the circuit shown in (Figure 19 LCD Display connection). Starting from most left of the display board, GND terminal is connected to the pin on the horizontal row of the breadboard supplied with ground (GND) power from the Arduino board. The VCC terminal on the display is connected to the pin on the horizontal row of the breadboard supplied with 5v positive power from the Arduino board. Contrast (VO) terminal is connected to the Wiper terminal on the potentiometer with a blue jumper cable. Register select (RS) terminal on the display board is connected to Digital pin (10) on the Arduino board routed from the breadboard using a purple jumper cable. Read/Write (RW) terminal on the display is connected to the pin on the horizontal row of the breadboard supplied with GND power from the Arduino board. Enable (E) terminal on the display board is connected to the Digital pin (9) on the Arduino board routed from the breadboard using a purple jumper cable. DB4 terminal is connected to the Digital pin (5) on the Arduino board routed from the breadboard using a brown jumper cable. DB5 terminal is connected to the Digital pin (6) on the Arduino board routed from the breadboard using a brown jumper cable. DB6 terminal is connected to the Digital pin (7) on the Arduino board routed from the breadboard using a brown jumper cable. DB7 terminal is connected to the Digital pin (8) on the Arduino board routed from the breadboard using a brown jumper cable. LED Anode terminal on the display board is connected directly to the 5v power pin on the breadboard. LED Cathode terminal on the display board is connected to a resistor and the resistor is then connected to the ground (GND) pin on the breadboard.

PROTOTYPE SCREENSHOTS

Raindrop sensor module

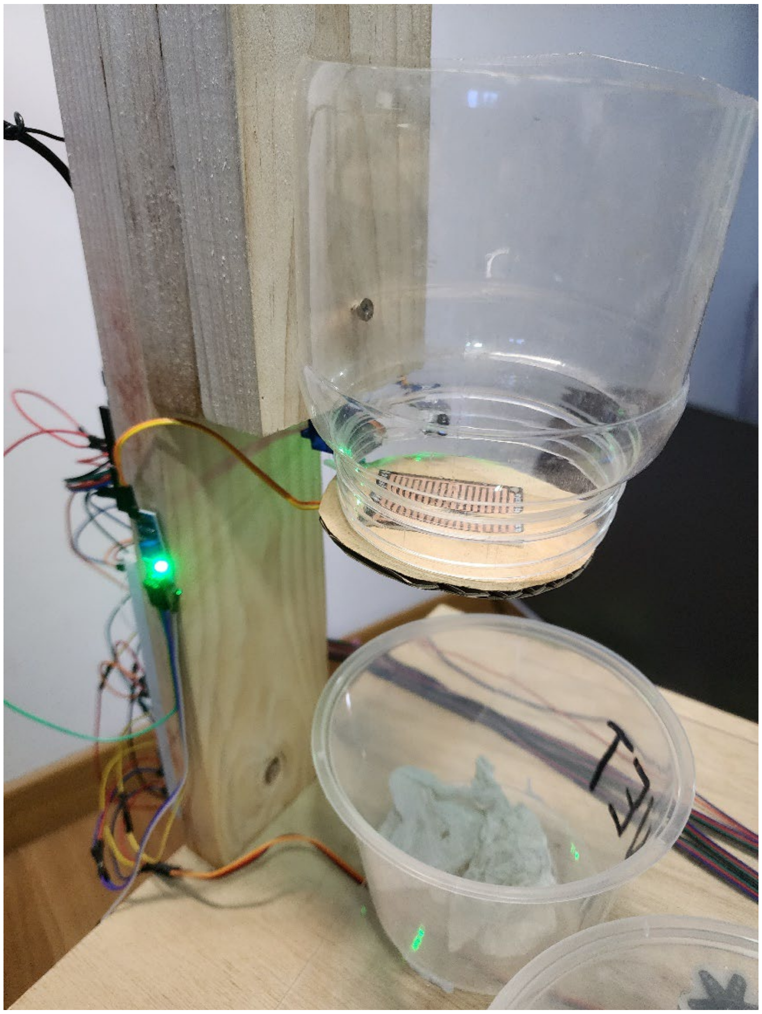

Figure 20 Placement of the Raindrop Sensor Module

Figure 20 (Placement of the Raindrop Sensor Module) demonstrates the placement of the Raindrop sensor which is inside the top container on the lid where the trash is first put to sense whether it belongs to the wet waste category or dry waste category.

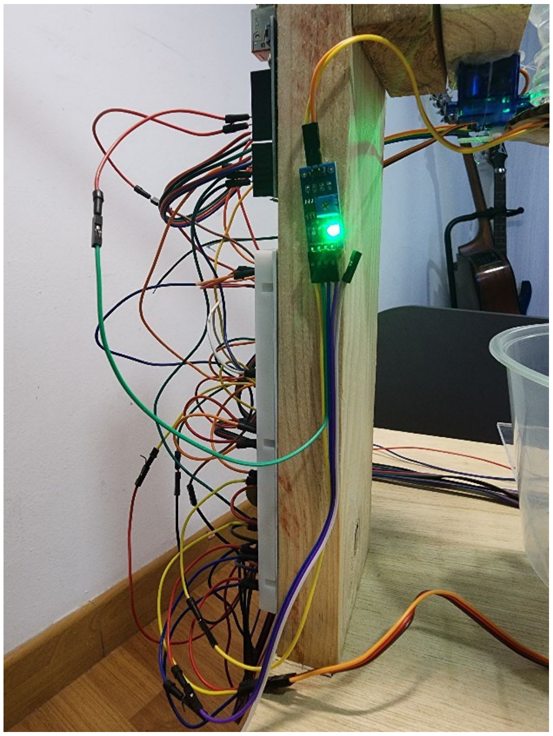

Figure 21 LN393 comparator and potentiometer module

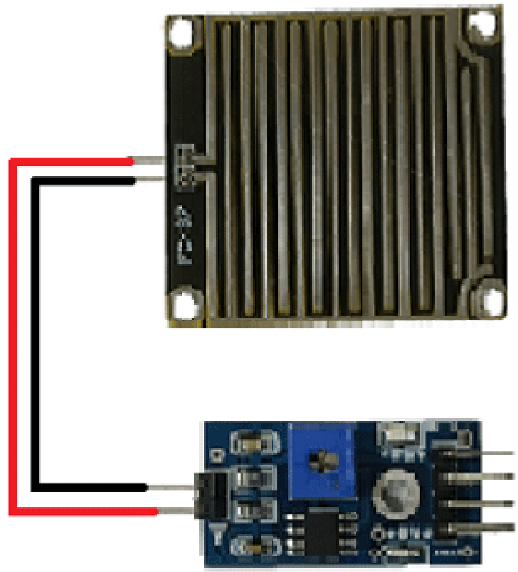

Figure 22 Raindrop sensor to LN393 and potentiometer connection

The Raindrop sensor module comes with the built in LN393 comparator and potentiometer module which is connected to the Raindrop sensor module shown in Figure 22 (Raindrop sensor to LN393 and potentiometer connection).

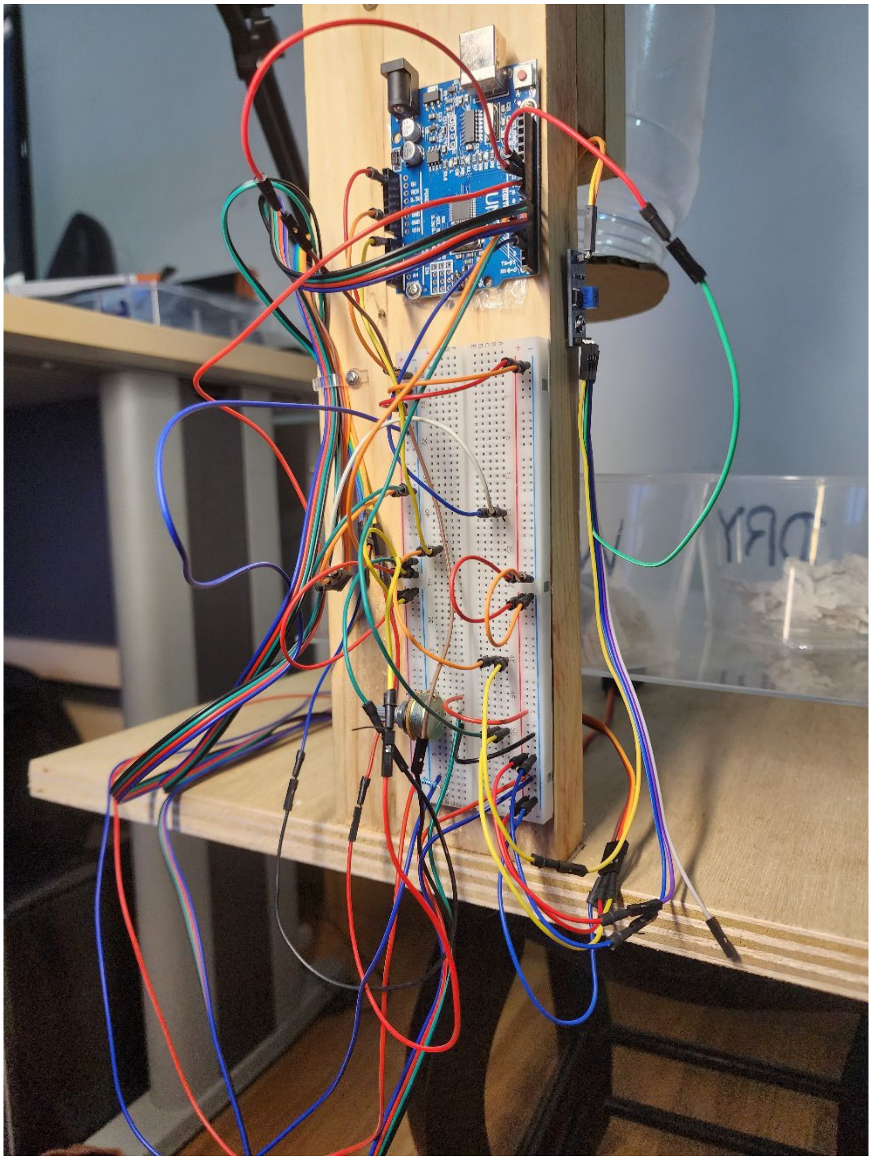

Figure 23 Circuit wiring

Figure 23 (Circuit wiring) shows the implementation of the wiring for the project. On the most right side of the support board, placed is the Raindrop sensor control module (LN393 and potentiometer). As you can see the green wire which is the signal is connected to the Digital Pin (11) on the Arduino board. Unlike the soil moisture sensor demonstrated in Figure 14 (Rain sensor module/ Soil moisture sensor circuit connection), the Raindrop sensor is configured to output Digital signal instead of analog signal. Tracing the purple wire from the VCC terminal of the control module is connected to the 5v power output pin on the breadboard. The blue wire from the GND terminal of the control module is connected to the GND pin of the breadboard.

IR sensor

Figure 24 IR sensor placement

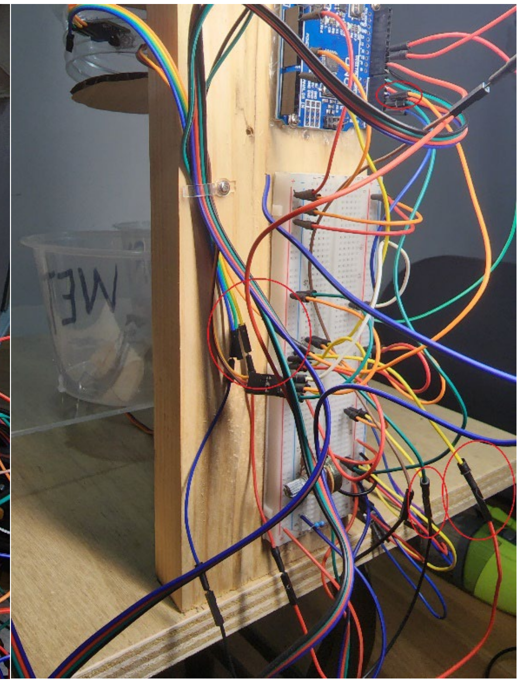

Figure 25 IR sensor wiring

Figure 24 (IR sensor placement) shows the placement of the IR sensor inside the top container to detect waste dropped inside the container to sense. Figure 25 (IR sensor wiring) shows a screenshot of the circuit wiring of the IR sensor. **Within the red circles are the wires that are connected to the IR module**. The IR sensor module is connected with a green-, yellow-, and orange-colored wires. The green is wire connected to the GND terminal of the IR sensor module. A blue male cable is then connected to the green wire with the female end. The blue cable is then connected to a female black jumper cable to extend. A dark green male cable is then connected to the female black extended cable. Dark green cable is then connected to the GND pin of the breadboard. As for the positive terminal of the sensor module, a yellow wire connected to the postive terminal of the IR sensor module. A red male cable is then connected to the yellow wire with the female end. The red cable is then connected to a female red jumper cable to extend. A yellow male cable is then connected to the female red extended cable. Yellow cable is then connected to the positive pin of the breadboard. As for the signal cable, an orange cable is connected to the DO terminal of the sensor. The orange cable is then paired with male orange cable which is then connected to the Digital pin (4) of the Arduino board.

MG946R Metal Gear Servo

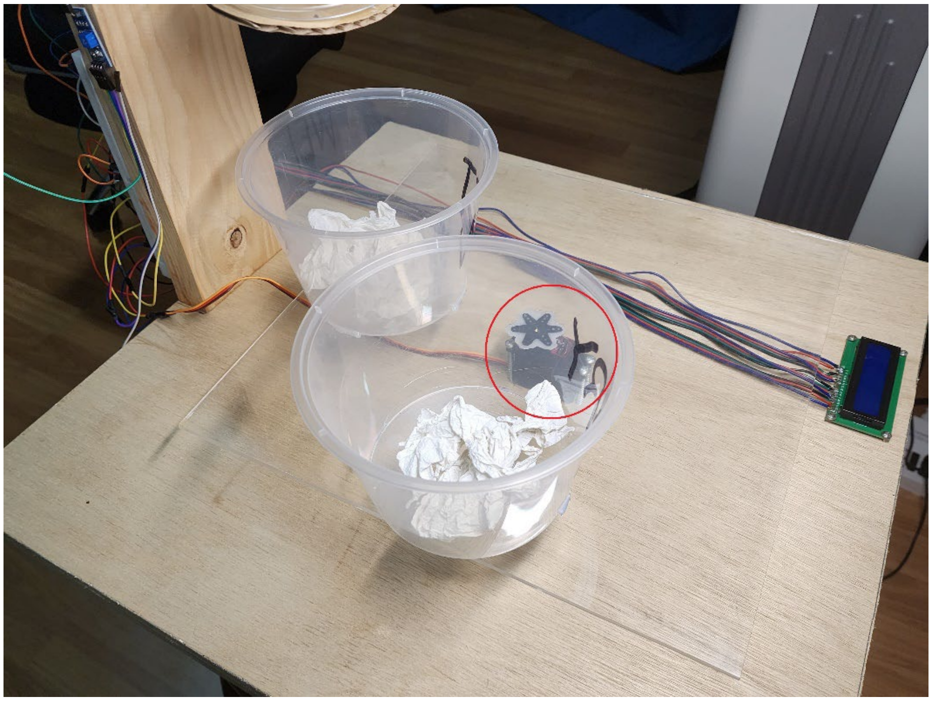

Figure 26 MG946R Metal Gear Servo placement

Figure 26 (MG946R Metal Gear Servo placement) depicts the positioning of the MG946R Metal Gear servo indicated by the red circle in Figure 25. The servo is responsible for rotating to the correct position based on the sensor reading to position the wet waste bin and dry waste bin directly beneath the top container.

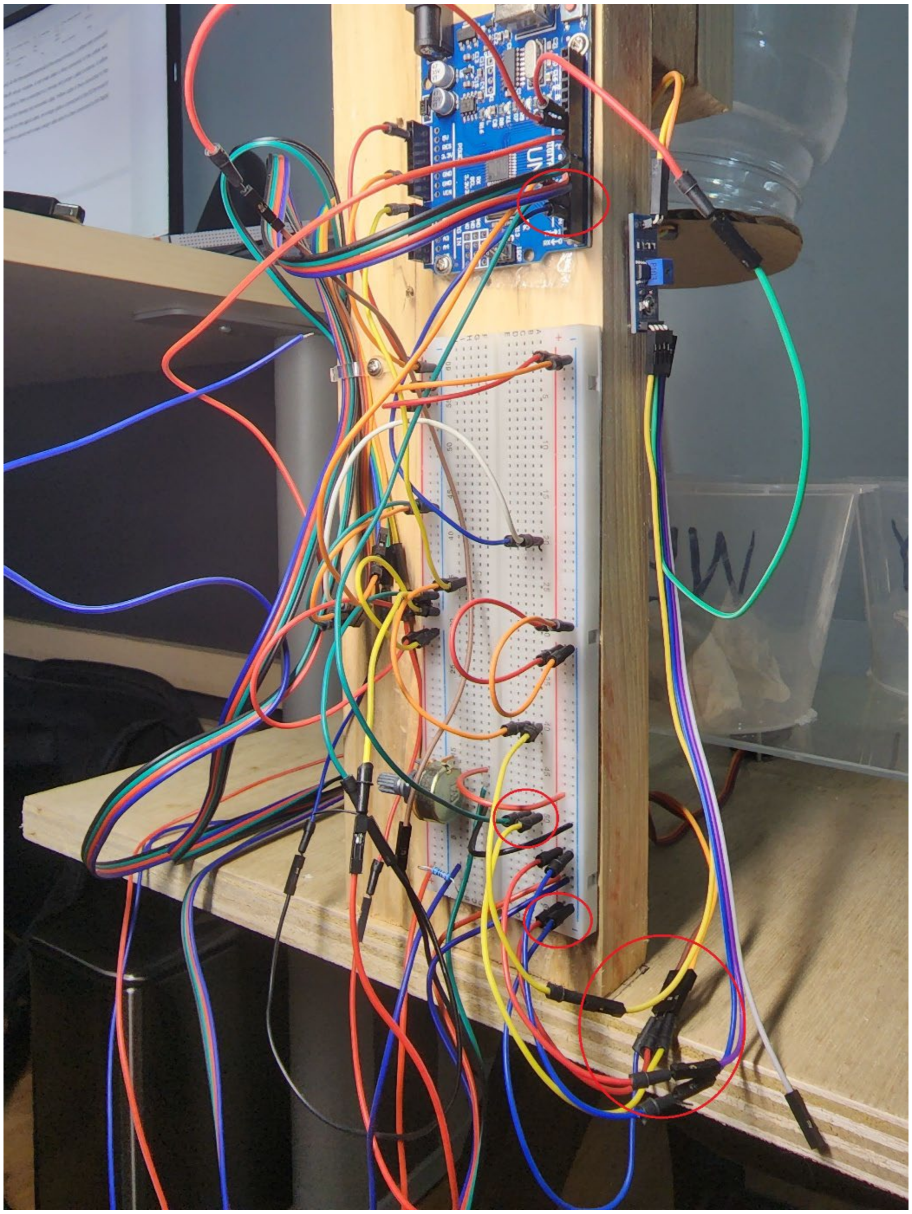

Figure 27 servo connections

Figure (27 servo connections) demonstrates the connection points for the servo which is circled in red. The servo uses three female connectors to connect to the terminals. Orange = Signal terminal, Red = Positive terminal, Brown = negative terminal. A yellow male jumper cable is linked with the orange female cable from the servo. The orange jumper cable is then connected to the breadboard. A dark green male cable is connected to the vertical pin of the breadboard which connects to the yellow jumper cable. Then the dark green cable is connected to the Digital pin (3) of the Arduino board.

As for the positive lead of the servo, the red female cable is linked with a male red jumper cable which is then connected to the 5v power pin of the breadboard.

As for the negative lead of the servo, the brown female cable is linked with a male blue jumper cable which is connected to the GND pin of the breadboard.

SG90 Micro Servo

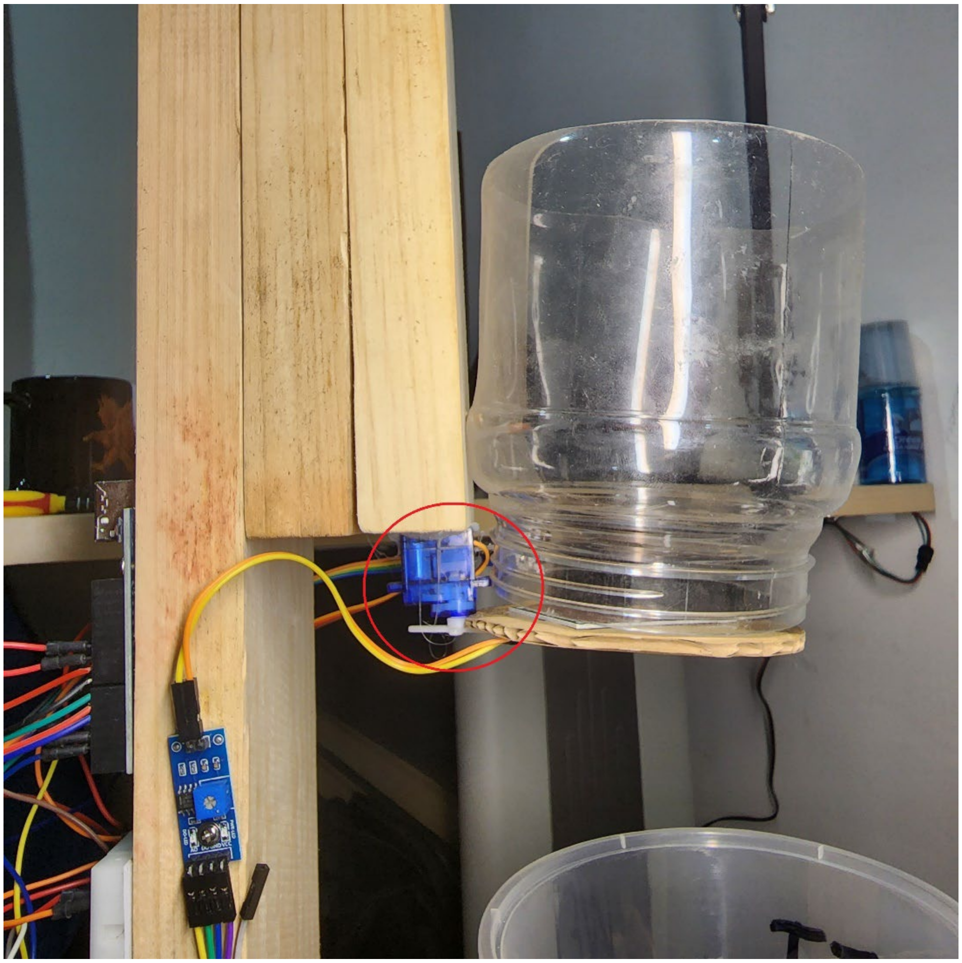

Figure 28 SG90 Micro Servo placement

Figure 28 (SG90 Micro Servo installation) depicts the location of the highlighted micro server within the red circle, which is located beneath the support plank that houses the top container. The servo responsible for opening the container's lid after the sensors have finished reading.

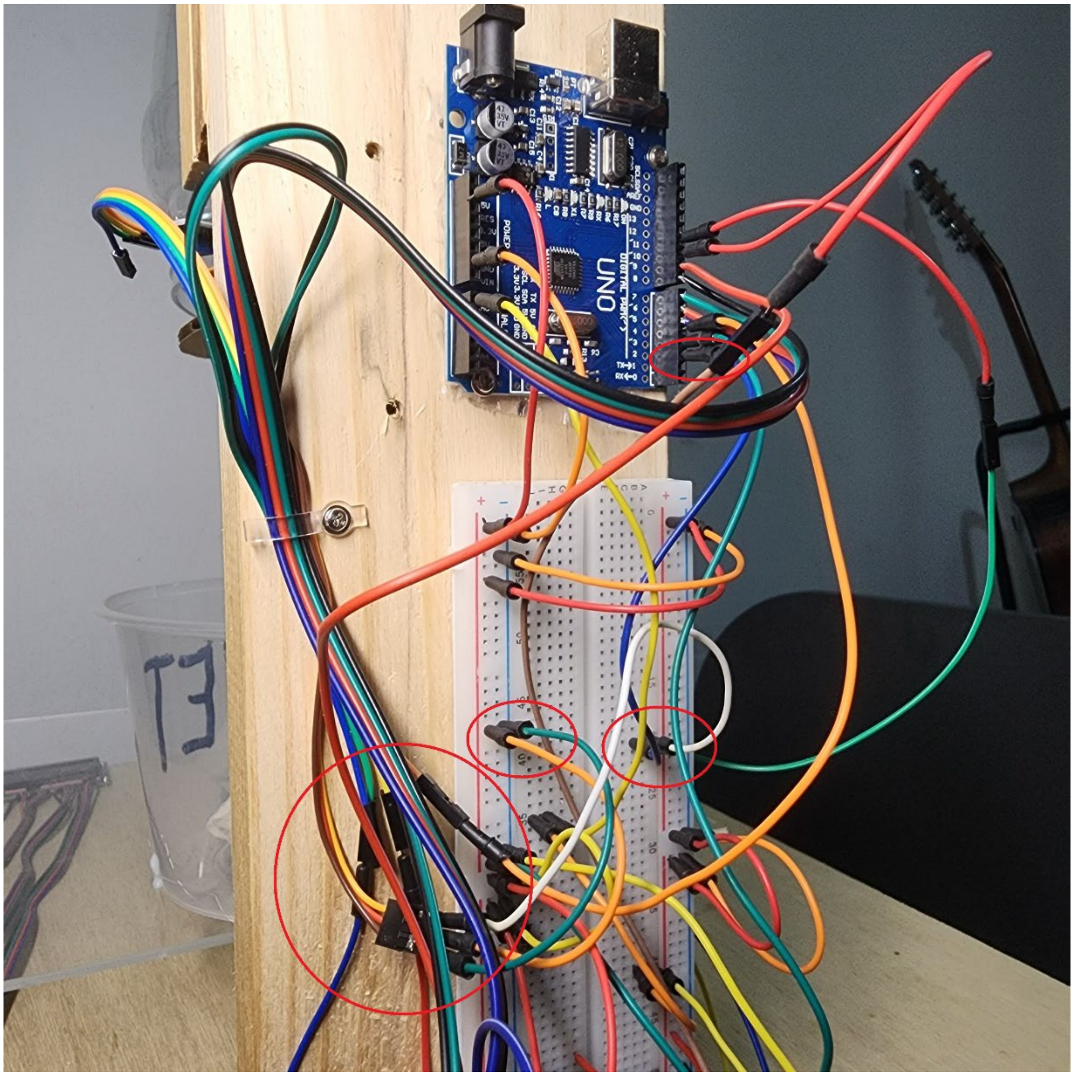

Figure 29 SG90 Micro Servo circuit connection

Figure 29 (SG90 Micro Servo circuit connection) demonstrates the connection points for the servo which is circled in red. SG90 Micro Servo uses three female connectors to connect to the terminals. Orange = Signal terminal, Red = Positive terminal, Brown = negative terminal. The brown female wire from the servo is linked with a male dark green jumper cable which is then connected to the GND pin of the breadboard. The orange female wire from the servo is linked with a male orange jumper cable which is then connected to the 5v positive pin of the breadboard. The female yellow cable from the servo which carries the signal is linked with the male white jumper cable and connected to a pin on the breadboard. A blue male jumper cable is linked with the white male jumper cable on the breadboard which the other end leads to the Digital pin (2) of the Arduino board.

LCD Display

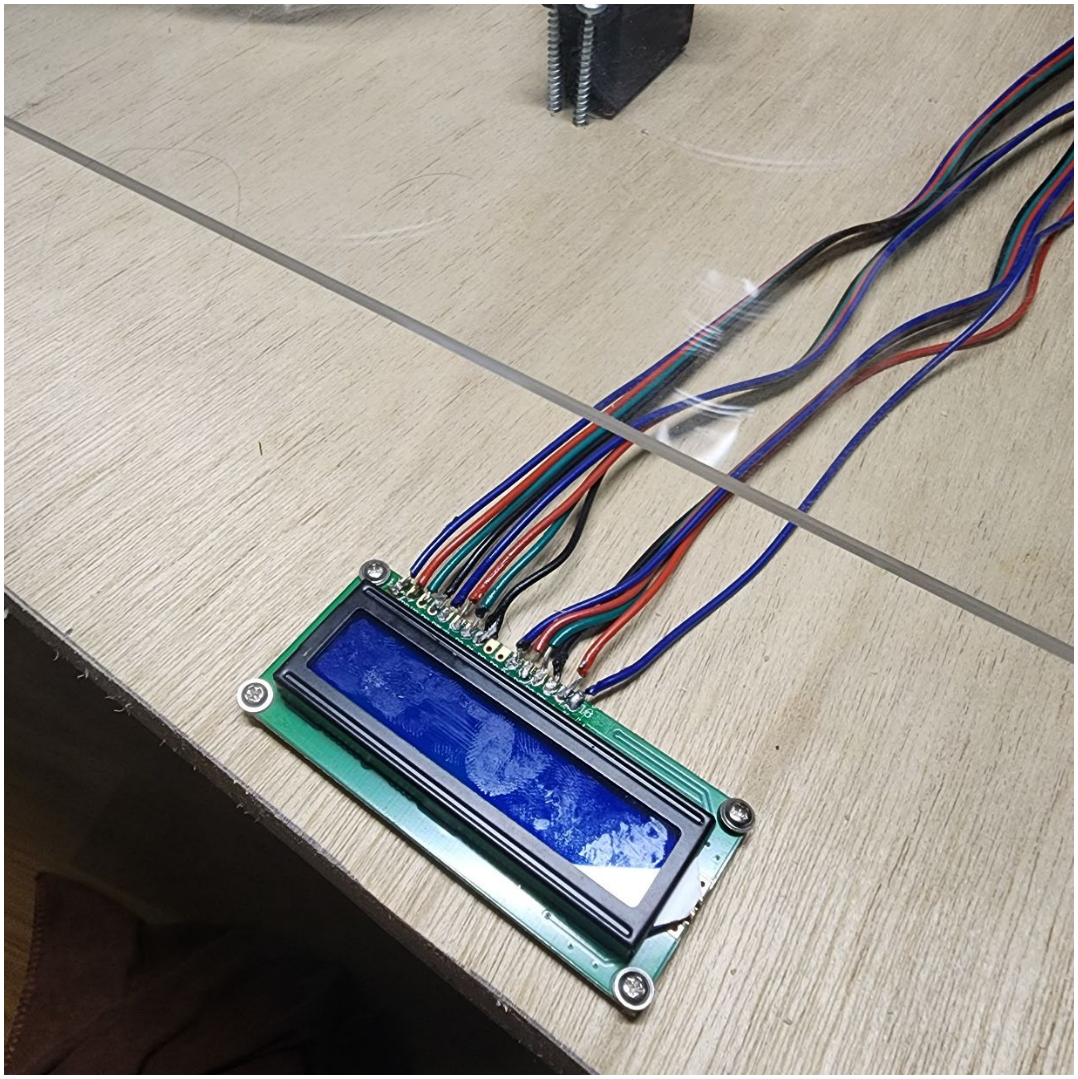

Figure 30 LCD Display placement

Figure 30 (LCD Display placement) shows the placement of the LCD display which is on the main board. It is responsible of informing the user when wet waste has been detected and when the dry waste has been detected. It also shows the status of the system for example when the system is booting and when the system is ready to be used. The wires were soldered to the display board as there was a limitation of jumper wires available.

SOURCE CODE

```
#include <Servo.h>
#include <LiquidCrystal.h>
/* LCD pin configuration */
LiquidCrystal lcd(10,9,5,6,7,8); // LCD pins

/* IR sensor  pin configuration */
int ir_pin = 4 ;
/* Raindrop sensor  pin configuration */
int digital_wetSensor_pin = 11;
/* Servo varaibles for the two servos used */
Servo servo,mini_servo;
/* Variable (pos1) = servo */
int pos1 = 90;
/* Variable (pos1) = mini_servo */
int pos2 = 75;
/* Time interval of IR sensor read INPUT signal */
const unsigned long eventInterval = 5000;
/* Time interval of Raindrop sensor read INPUT signal */
const unsigned long eventInterval_1 = 600;

unsigned long previousTime = 0 ;
unsigned long previousTime_1 = 0 ;
int ir_sense = 2;
int wet_sense = 2;


void setup() {
  pinMode (ir_pin, INPUT); // sensor pin INPUT
  pinMode(digital_wetSensor_pin, INPUT); // Raindrop sesnor pin INPUT

  Serial.begin(9600); // Serial monitor begin
  lcd.begin (16,2);  // LCD initialisation
  lcd.setCursor(0,0); // lCD set cursor to 0 column 0 row
  lcd.print("Getting ready"); // Print "Getting ready"
  servo.write(90);  // postion the servo to 90 degrees
  mini_servo.write(75);// postion mini_server to the postiion 75 degrees
  delay(1000);
  lcd.clear(); // clear lcd
  lcd.setCursor(0,0);
  lcd.print("Ready to use now");// print ready to use
  delay(3000); // dealy 3 seconds


}
```

Adam Sayyaf Abdulla

```
void loop() {

  unsigned long currentMillis = millis();

  int statusSensor = digitalRead (ir_pin);
  int wet_sensor = analogRead(A0);
  int d_wet_sensor = digitalRead(digital_wetSensor_pin);
  servo.attach(3, 500, 2500);
  mini_servo.attach(2, 500, 2500);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Scanning for ");
  lcd.setCursor(0,1);
  lcd.print("waste...");


/* TIMED event = IR sensor */
if(currentMillis - previousTime >= eventInterval){
  //Serial.print("statusSensor");
  //Serial.println(statusSensor);
  if (statusSensor == 0 ){
    ir_sense = 1;
  }else{
    ir_sense = 2;
  }
  previousTime = currentMillis;
}
```

```
/* TIMED event = Raindrop sensor */
if(currentMillis - previousTime >= eventInterval_1){
  //Serial.print("\nWetSensor\n");
  //Serial.println(d_wet_sensor);
  if (d_wet_sensor == LOW){
      Serial.print("Wet detected");
      wet_sense = 1;
      //ir_sense = 1;
  }else {
    //if (d_wet_sensor == HIGH){
      //Serial.print("DRY detected");
      wet_sense = 0;
      //ir_sense = 1;
  }
  previousTime_1 = currentMillis;
}

/* Check for IR sensor and Raindrop sensor INPUT for Wet wate detection
and perfom the action */
if (ir_sense == 1 && wet_sense == 1 ){
        Serial.print ("WET DETECTED");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Detected:");
        lcd.setCursor(1,8);
        lcd.print("Wet waste:");
        servo.write(175);
        delay(1000);
        mini_servo.write(165);
        delay(1000);
        mini_servo.write(75);
        ir_sense = 2;
        wet_sense = 2;
        delay(1000);
}
```

```
/* Check for IR sensor and Raindrop sensor INPUT for Dry wate detection
andperfom the action */
if (ir_sense == 1 && wet_sense == 0 ){
        Serial.print ("DRY DETECTED");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Detected:");
        lcd.setCursor(1,8);
        lcd.print("Dry waste:");
        servo.write(90);
        delay(1000);
        mini_servo.write(165);
        delay(1000);
        mini_servo.write(75);
        ir_sense = 2;
        wet_sense = 2;
        delay(1000);
}
}
```

Figure 31 Source code

```
#include <Servo.h>
#include <LiquidCrystal.h>
/* LCD pin configuration */
LiquidCrystal lcd(10,9,5,6,7,8); // LCD pins

/* IR sensor  pin configuration */
int ir_pin = 4 ;
/* Raindrop sensor  pin configuration */
int digital_wetSensor_pin = 11;
/* Servo varaibles for the two servos used */
Servo servo,mini_servo;
/* Variable (pos1) = servo */
int pos1 = 90;
/* Variable (pos1) = mini_servo */
int pos2 = 75;
/* Time interval of IR sensor read INPUT signal */
const unsigned long eventInterval = 5000;
/* Time interval of Raindrop sensor read INPUT signal */
const unsigned long eventInterval_1 = 600;

unsigned long previousTime = 0 ;
unsigned long previousTime_1 = 0 ;
int ir_sense = 2;
int wet_sense = 2;
```

Figure 32 Headers and variables

Figure 32 demonstrates the header files and the variables used in the source code. <servo.h> header file is used to call the function related to controlling the servos in the project. <LiquidCrystal.h> header file is included to call the function related to controlling the LCD display. Variable ir_pin is initialized to store the value of the pin that the IR sensor connects to (pin 4). Variable digital_wetSensor_pin is initialized to store the value of the pin that the Raindrop sensor connects to (pin 11). Servo function is called to initialize the variables that corelates to the name of the servos. Variable name: servo = MG9464 Metal gear servo. Variable name: mini_servo = SG90 Micro servo. Variable pos1 is initialized to 90 where 90 is the degree of angle that the servo will be at the start of the program. Variable pos1 is initialized to 75 where 75 is the degree of angle that the min_servo will be at the start of the program. Variable: eventInterval is intialisized to 5000 milliseconds (5 seconds). Variable: eventInterval_1 is intialisized to 600 milliseconds (0.6 seconds). These two variables are used to time the intervals for when the sensor be read. Variable ir_sense is initialized to 2 which is the neutral position when the program is expected to do nothing. wet_sense is initialized to 2 which is the neutral position when the program is expected to do nothing.

```
void setup() {
  pinMode (ir_pin, INPUT); // sensor pin INPUT
  pinMode(digital_wetSensor_pin, INPUT); // Raindrop sesnor pin INPUT

  Serial.begin(9600); // Serial monitor begin
  lcd.begin (16,2);  // LCD initialisation
  lcd.setCursor(0,0); // lCD set cursor to 0 column 0 row
  lcd.print("Getting ready"); // Print "Getting ready"
  servo.write(90);  // postion the servo to 90 degrees
  mini_servo.write(75);// postion mini_server to the postiion 75 degrees
  delay(1000);
  lcd.clear(); // clear lcd
  lcd.setCursor(0,0);
  lcd.print("Ready to use now");// print ready to use
  delay(3000); // dealy 3 seconds


}
```

Figure 33 Setup codes

The codes used in the void setup function are depicted in Figure 33. This function is executed only once at initialization of the Arduino board. Here, the pin Mode for the IR and Raindrop sensors is stated. In addition, serial monitor is configured to begin at this moment. At this moment, LCD display is also initialized. During the setup run, the starting position of the servos is determined. After positioning the servos, the LCD will display Ready to use.

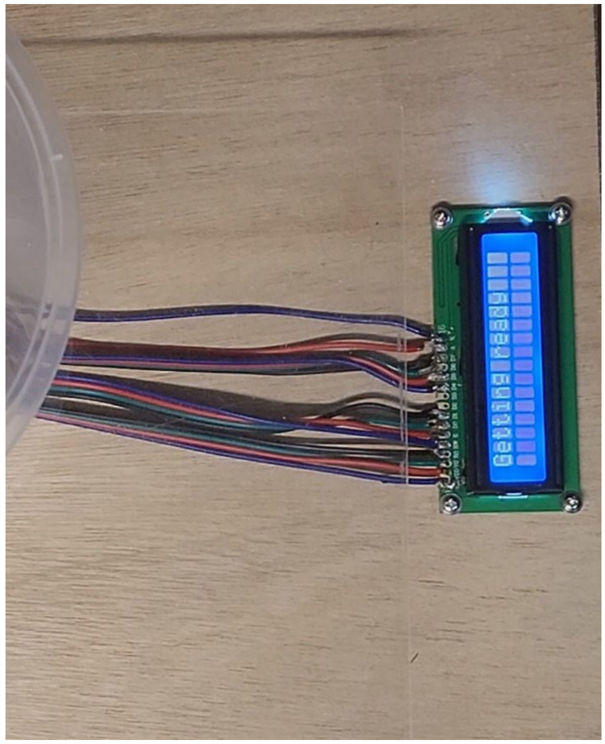

Figure 34 Output of setup

Figure 34 demonstrated the LCD display output of the setup function in the code.

```
void loop() {

  unsigned long currentMillis = millis();

  int statusSensor = digitalRead (ir_pin);
  int wet_sensor = analogRead(A0);
  int d_wet_sensor = digitalRead(digital_wetSensor_pin);
  servo.attach(3, 500, 2500);
  mini_servo.attach(2, 500, 2500);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Scanning for ");
  lcd.setCursor(0,1);
  lcd.print("waste...");


/* TIMED event = IR sensor */
if(currentMillis - previousTime >= eventInterval){
  //Serial.print("statusSensor");
  //Serial.println(statusSensor);
  if (statusSensor == 0 ){
    ir_sense = 1;
  }else{
    ir_sense = 2;
  }
  previousTime = currentMillis;
}
```

Figure 35 Loop function codes

The majority of the Arduino sketch is executed here. The programme begins immediately after the opening curly bracket (), runs until it encounters the closing curly bracket (), and then returns to the first line of loop() and begins again. The Raindrop sensor is configured to the use the digitalRead signal rather than the analogRead signal. The analogRead signal is to monitor the level of the resistance across the Raindrop sensor PCB board. One of the issues faced during the implementation of the program was the timing of the two sensors. Where IR sensor inputs the reading faster than the Rain drop sensor can sense water on its PCB. Since two sensors were required to evaluate an output, the output wouldn't perform as expected most of the times. Due to that, the millis() function was used to time the events of the reading of the two sensors independently. The function millis() is a built in function in Arduino that records the time in milliseconds from the point the Arduino is powered. Using this function the two sensors were able to act independently.

```
21:44:56.522 -> statusSensor: 1
21:45:01.510 -> statusSensor: 1
21:45:06.521 -> statusSensor: 1
21:45:11.524 -> statusSensor: 1
21:45:16.522 -> statusSensor: 1
21:45:21.529 -> statusSensor: 1
21:45:26.527 -> statusSensor: 1
```

Figure 36 Serial Output of IR sensor

Figure 36 shows the timed reading of the IR sensor which is set 5000 milliseconds (5 seconds) in order to wait for the Wet sensor to input the reading. If IR sensor sense an object, the program then initializes the variable ir_sense = 1, else ir_sense = 2.

```
/* TIMED event = Raindrop sensor */
if(currentMillis - previousTime >= eventInterval_1){
  //Serial.print("\nWetSensor: ");
  //Serial.println(d_wet_sensor);
  if (d_wet_sensor == LOW){
      Serial.print("Wet detected");
      wet_sense = 1;
      //ir_sense = 1;
  }else {
    //if (d_wet_sensor == HIGH){
      //Serial.print("DRY detected");
      wet_sense = 0;
      //ir_sense = 1;
  }
  previousTime_1 = currentMillis;
}
```

Figure 37 Time event for Raindrop sensor

Figure 37 shows the code used to time the sensor reading of the Raindrop sensor module. This piece of code allows the program to read the input signal of the Raindrop sensor every 600 milliseconds (0.6 seconds). This was done because the Raindrop sensor takes a bit of timing to determine whether the object is wet or dry. If the digitalRead signal from the sensor is LOW, then the wet_sense variable is initialized to 1, else the wet_sense variable is initialized to 0.

```
21:59:16.150 -> WetSensor: 1
21:59:16.150 ->
21:59:16.150 -> WetSensor: 1
21:59:16.215 ->
21:59:16.215 -> WetSensor: 1
21:59:16.215 ->
21:59:16.215 -> WetSensor: 1
21:59:16.215 -> statusSensor: 1
21:59:16.759 ->
21:59:16.759 -> WetSensor: 1
21:59:16.759 ->
21:59:16.759 -> WetSensor: 1
21:59:16.759 ->
21:59:16.759 -> WetSensor: 1
21:59:16.808 ->
21:59:16.808 -> WetSensor: 1
21:59:16.808 ->
21:59:16.808 -> WetSensor: 1
21:59:16.852 ->
21:59:16.852 -> WetSensor: 1
21:59:16.852 ->
21:59:16.852 -> WetSensor: 1
21:59:16.852 ->
21:59:16.852 -> WetSensor: 1
21:59:16.852 ->
21:59:16.852 -> WetSensor: 1
```

Figure 38 Output of timed event of Raindrop sensor

Figure 38 shows that the timed interval between the Raindrop sensor reading is approximately 600 milliseconds.

```
/* Check for IR sensor and Raindrop sensor INPUT for Wet wate detection
and perfom the action */
if (ir_sense == 1 && wet_sense == 1 ){
        Serial.print ("WET DETECTED");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Detected:");
        lcd.setCursor(1,8);
        lcd.print("Wet waste:");
        servo.write(175);
        delay(1000);
        mini_servo.write(165);
        delay(1000);
        mini_servo.write(75);
        ir_sense = 2;
        wet_sense = 2;
        delay(1000);
}
/* Check for IR sensor and Raindrop sensor INPUT for Dry wate detection
andperfom the action */
if (ir_sense == 1 && wet_sense == 0 ){
        Serial.print ("DRY DETECTED");
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Detected:");
        lcd.setCursor(1,8);
        lcd.print("Dry waste:");
        servo.write(90);
        delay(1000);
        mini_servo.write(165);
        delay(1000);
        mini_servo.write(75);
        ir_sense = 2;
        wet_sense = 2;
        delay(1000);
}
}
```

Figure 39 Actions for when object detected

Figure 39 depicts the two if statements that are utilized to output the action when the sensor values are assessed. The if statement that defines the action if the object is identified as a wet waste comes first. If ir_sense = 1 and wet_sense = 1, the serial monitor will output Wet Detected and the LCD will display the same message. The servo will spin 175 degrees in order to position the wet waste bin beneath the container on top. The mini servo will then open the lid and dump the thing into the bin. Then the value 2 will be assigned to the variables ir_sense and wet_sense.

Regarding dry waste, the if statement that describes the action if the object is classified as dry waste occurs before any other statements. If ir_sense = 1 and wet_sense = 0, the serial monitor will output Dry Detected and the LCD will show the same message. The servo will rotate ninety degrees to place the dry trash bin beneath the container on top. The small servo will then open the lid and dispose of the object in the trash. Then the value 2 is set to the ir_sense and wet_sense variables.

LIMITATION

During the design of the circuit using Tinkercad, the limitations of several of the simulator's modules became known. Absent were the raindrop sensor module, the IR sensor module, and their control module. This resulted in the replacement of the IR sensor module with the PIR sensor module and the Raindrop sensor module with the Soil moisture sensor module. The IR sensor module and Raindrop sensor module feature an external control module with an LN393 comparator and a potentiometer for outputting digital signals to the Arduino board. This was employed for the project, although this module lacked in Tinkercad.

The absence of timed event teaching, which was required for this project, presented a second restriction. Two sensors have an enormous sensing gap. This was resolved by conducting more study on the millis() method and incorporating it into the project.

The last constraint was a misinterpretation of the number of jumper wires required for the project, which prevented the LCD display from being configured. This issue was resolved by soldering additional wires that were lying around to the display board. Consequently, it is challenging to connect the wires to the breadboard.

FUTURE ENHANCEMENTS

This project has a lot of potential for upgrades. In mind, there is a plan to introduce two more features to the project.

1. Trash level indicator: Using ultrasonic sensor to detect the level of garbage filled in each bin and indicating it in the system, which restricts users from pouring additional waste into the bin after a particular level has been achieved.
2. GSM module integration: Using a GSM module, the system is capable of transmitting its real-time status to a server. This will allow the waste management services to be notified when the bin needs to be emptied.

CONCLUSION

In conclusion, the implemented prototype has the potential to address the global waste management problem. The prototype can differentiate between wet and dry waste. In addition, it contains a mechanism for sorting the garbage deposited into the system. The prototype can become more compact and portable with future enhancements and customized PCB design. The primary objective of this project is to develop a prototype that may be utilized in urban households and businesses, where waste segregation is regarded to be the fundamental problem. Introduction of such systems would reduce the amount of human labor necessary to solve the problem. One of the primary aims of IoT is to enhance life and minimize the amount of human effort necessary to complete a task. This initiative has the potential to meet those expectations and make a lasting effect in the environment.

References

Chen, B., & Lee, J. (2020). Household waste separation intention and the importance of public policy. *International Trade, Politics and Development*.

Clifford, C. (2021, May 30). *Trillions of pounds of trash: New Technology tries to solve an old garbage problem*. Retrieved from CNBC: https://www.cnbc.com/2021/05/29/can-new-technology-solve-a-trillion-pound-garbage-problem.html

Das, D. (2022, March 28). *Arduino Rain Sensor Tutorial - How Rain Sensor Works and How to use it with Arduino*. Retrieved from circuitdigest.com: https://circuitdigest.com/microcontroller-projects/interfacing-rain-sensor-with-arduino

Environmental Protection Agency. (2022, September 6). *Basic Information about Landfill Gas*. Retrieved from Environmental Protection Agency: https://www.epa.gov/lmop/basic-information-about-landfill-gas

Fonseca, K. (2017, October 16). *Youth Incorporated Magazine*. Retrieved from https://youthincmag.com/wet-waste-dry-waste

Gillis, A. S. (2022, march 04). *What is IOT (internet of things) and how does it work? - definition from techtarget.com*. Retrieved from IoT Agenda: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

IR sensor and Arduino interfacing " Pija education. (2020, July 22). Retrieved from PIJA Education: https://pijaeducation.com/arduino/sensor/ir-sensor-and-arduino/

Joseph, J. (2022, March 21). *Everything you need to know about the Arduino hardware*. Retrieved from What is Arduino Uno Hardware Board?: https://circuitdigest.com/article/everything-you-need-to-know-about-arduino-uno-board-hardware

LCD 16x2: Pin Configuration, features and its working. (2019, August 26). Retrieved from ElProCus: https://www.elprocus.com/lcd-16x2-pin-configuration-and-its-working/

Pedamkar, P. (2022, Januray 19). *What is Iot?: Understanding and importance of IOT in future*. Retrieved from EDUCBA: https://www.educba.com/what-is-iot/

Schousek, T. (2018). *The Art of Assembly Language Programming Using PIC© Technology.*

What is a resistor?: Resistor fundamentals: Resistor guide. (n.d.). Retrieved from EEPower: https://eepower.com/resistor-guide/resistor-fundamentals/what-is-a-resistor/#