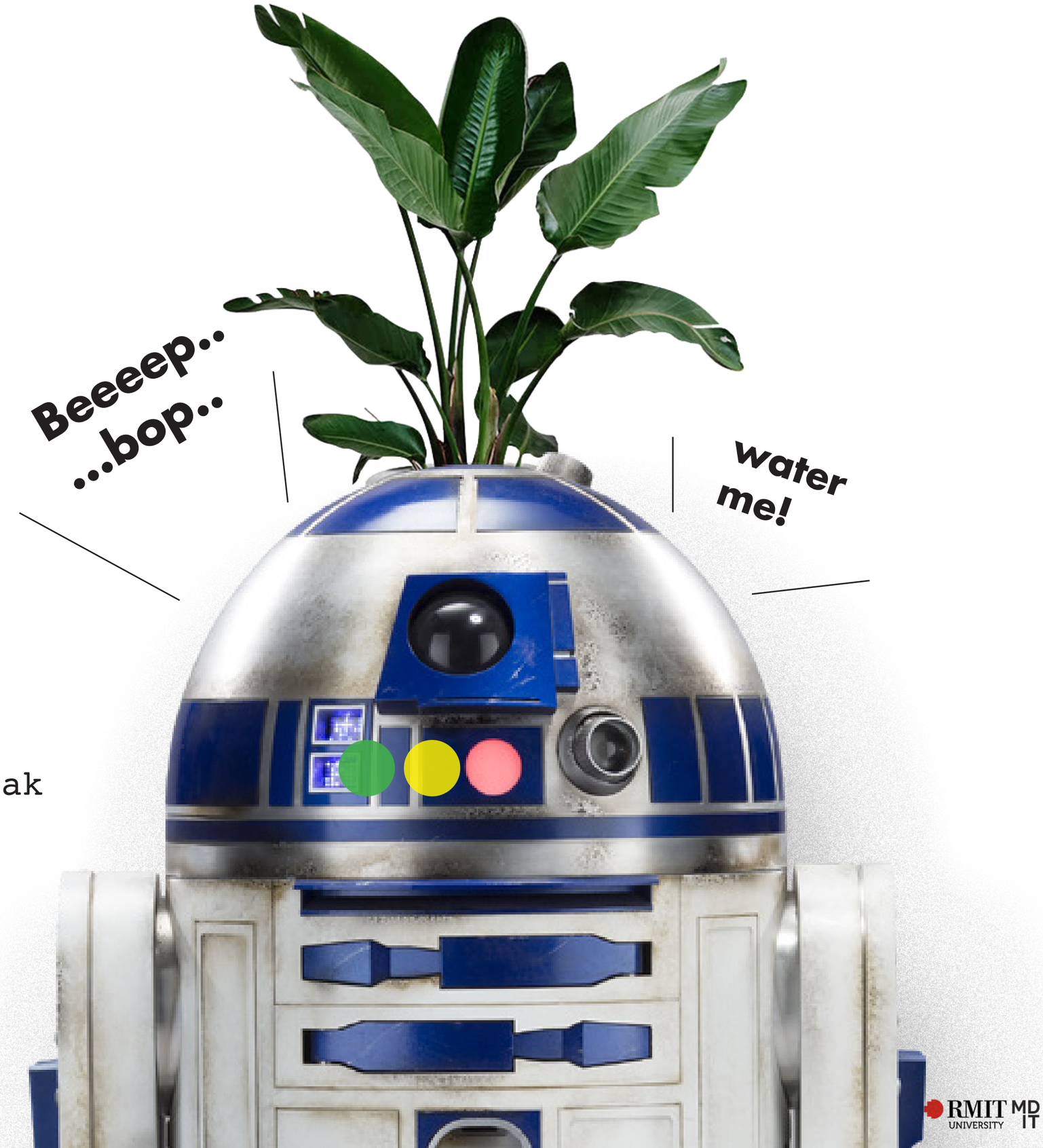
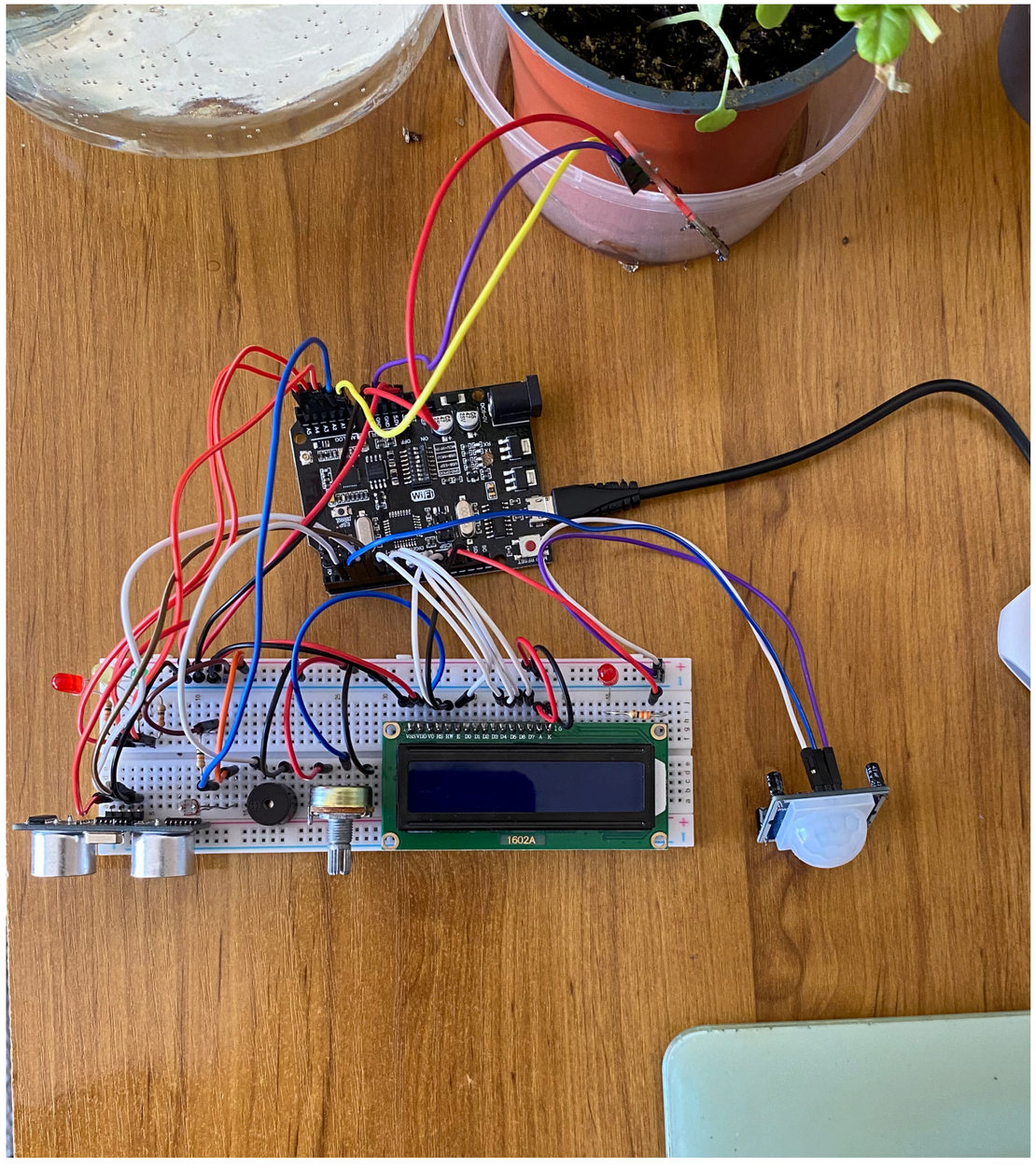
Zulfikar Sepyan - 3922236

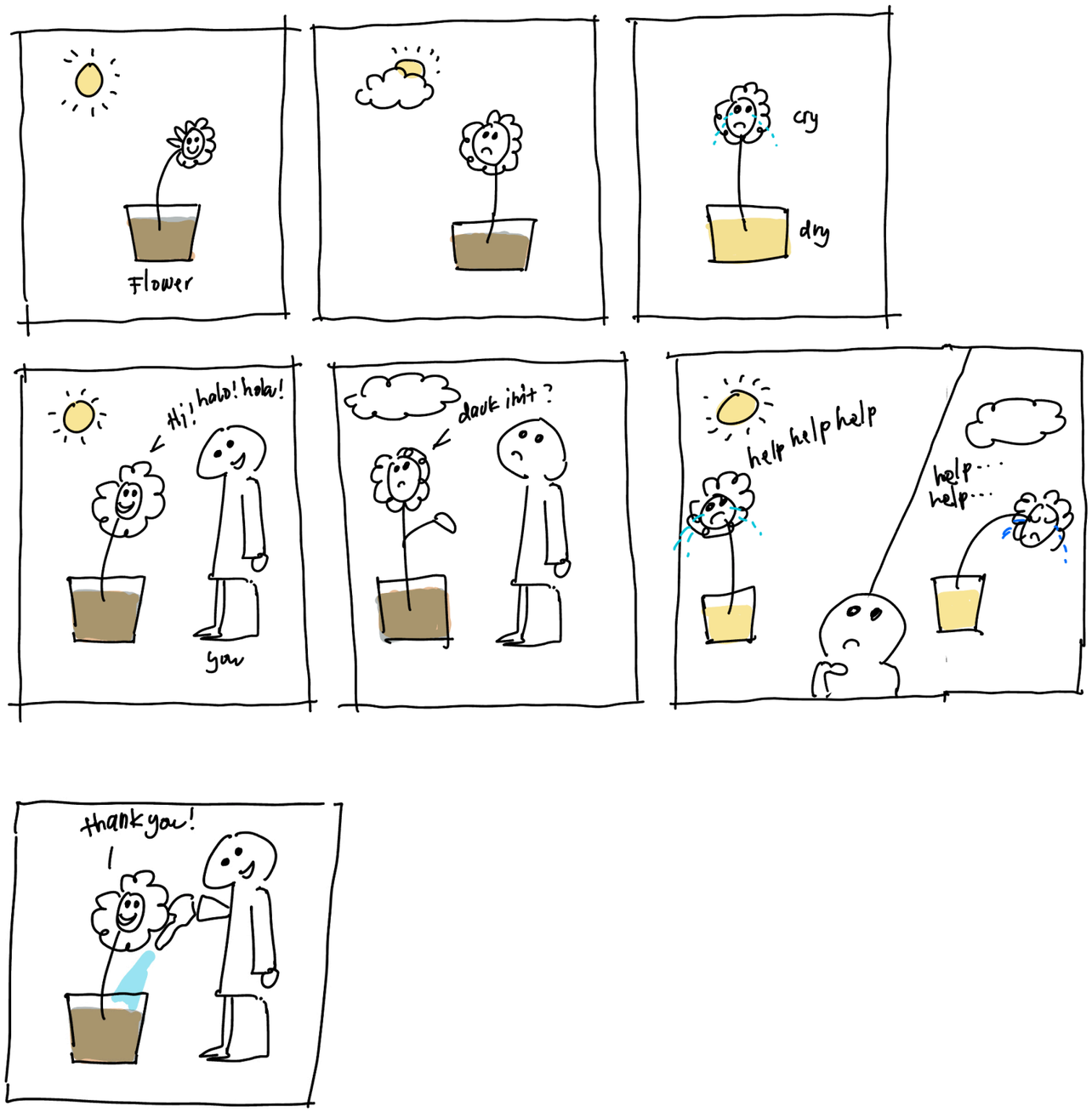
ARTURITO

if your plant can speak




WiFi
1602A

Introduction



Say you have a plant around, that is under your responsibility to feed it as if it is your pet. This plant may stand in a pot on the top of your garden table. Different with cats or dogs, a plant can not speak to you in whatever means. Or at least you will never hear it speaks a sound. Also, this plant would not be able to move away from where it is standing right now. However, what if your plant want to say something to you, in case of emergency? Or it just want to have a bit of chit-chat with you nearby?

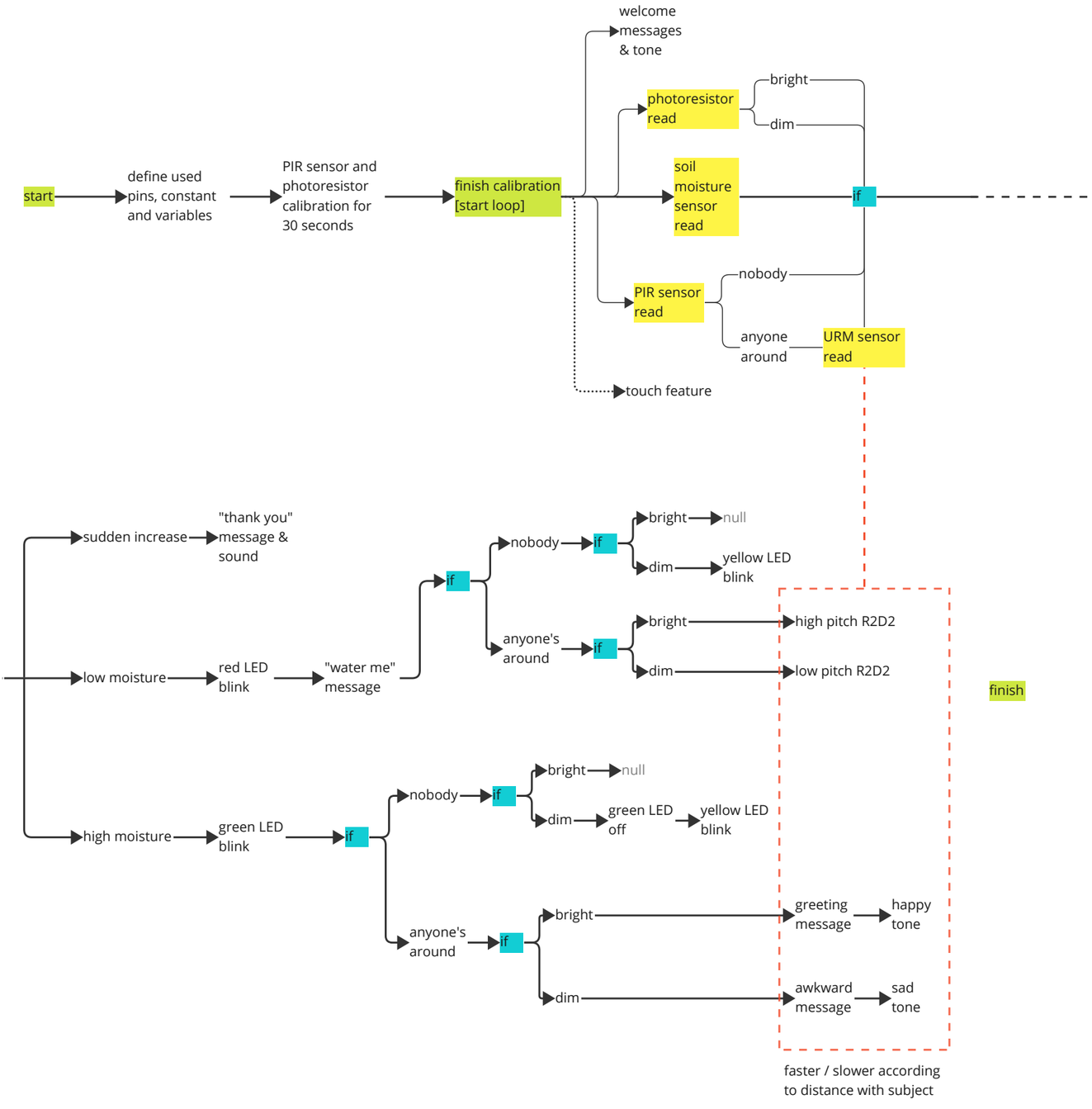
Using Arduino, we created a tool that can be pluged into your plant (to the soil). So that it can speak to you if it needs of sunlight. With you are around, it can speak more about random things. And it can cry in its own way when it needs more water, or when it needs you to move it somewhere sunny.

Moreover, this plant will appreciate everytime you give it some water. You will love your plant, and your plant will love you even more.

PSEUDOCODE & WORK FLOW

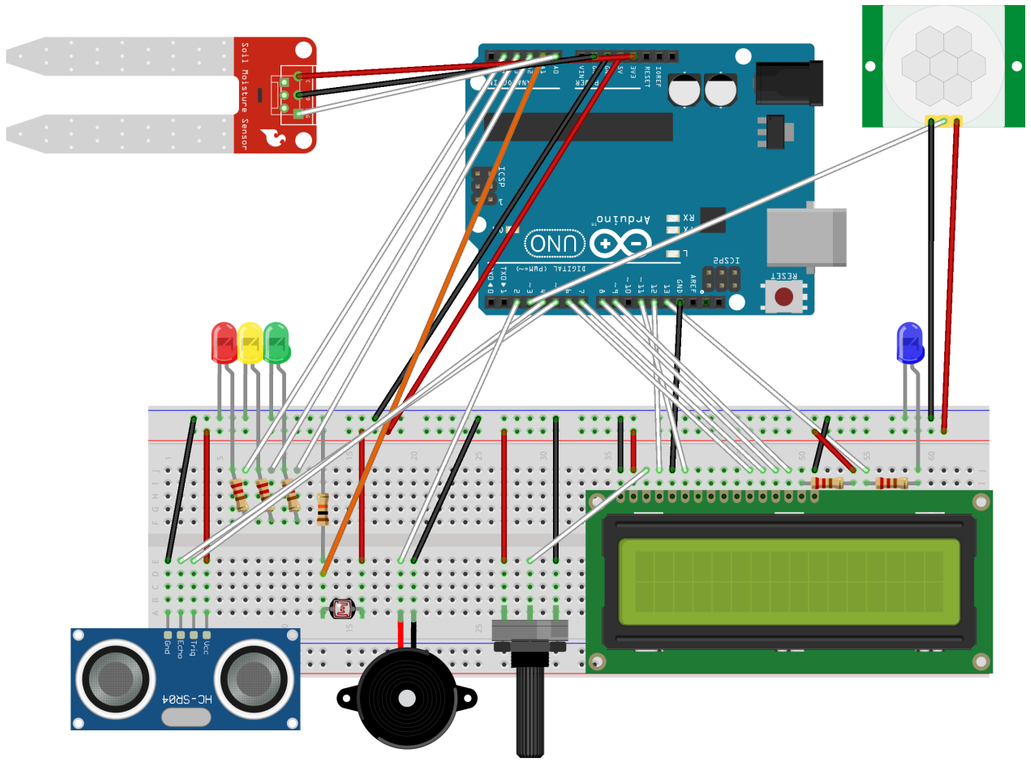
Arturito Pseudocode_Zulfikar Sepyan.ino

```
// this is pseudocode for Arturito
// author: Zulfikar Sepyan

// pins 12, 11, 6, 7, 8, 9 for LCD
// pin A0 for moisture sensor
// pin 2 for buzzer
// pin 3 for PIR
// pin 13 for blue led
// pin A2 for green led
// pin A4 for red led
// pin 4 and 5 for URM
// pin A1 for photoresistor

// create library for various greeting messages

void setup() {
  // initialize the machine
  // define input & output
  // set randomSeed
  // calibrate sensor
}

void loop() {
  // map moisture, ambience light, and distance reads
  // set all led off
  // if surrounding is dim, turned on yellow led. else, do nothing.

  // if moisture is low:
    // turn on red led
    // show water me message
    // only if there is anyone around
      // play R2D2 sound
      // vary the sound according to ambience light:
        // bright --> high pitch
        // dim --> low pitch
      // vary the delay according to distance with anyone around
        // near --> low delay / chattier
        // far --> long delay / less chatty

  // if moisture is high:
    // turn on green led
    // only if there is anyone around
      // if ambience light is bright
        // show various greeting message
        // play happy tone
      // if ambience light is dim
        // show awkward message
        // play sad tone

  // if there is sudden peak in moisture:
    // show thank you message
}
```

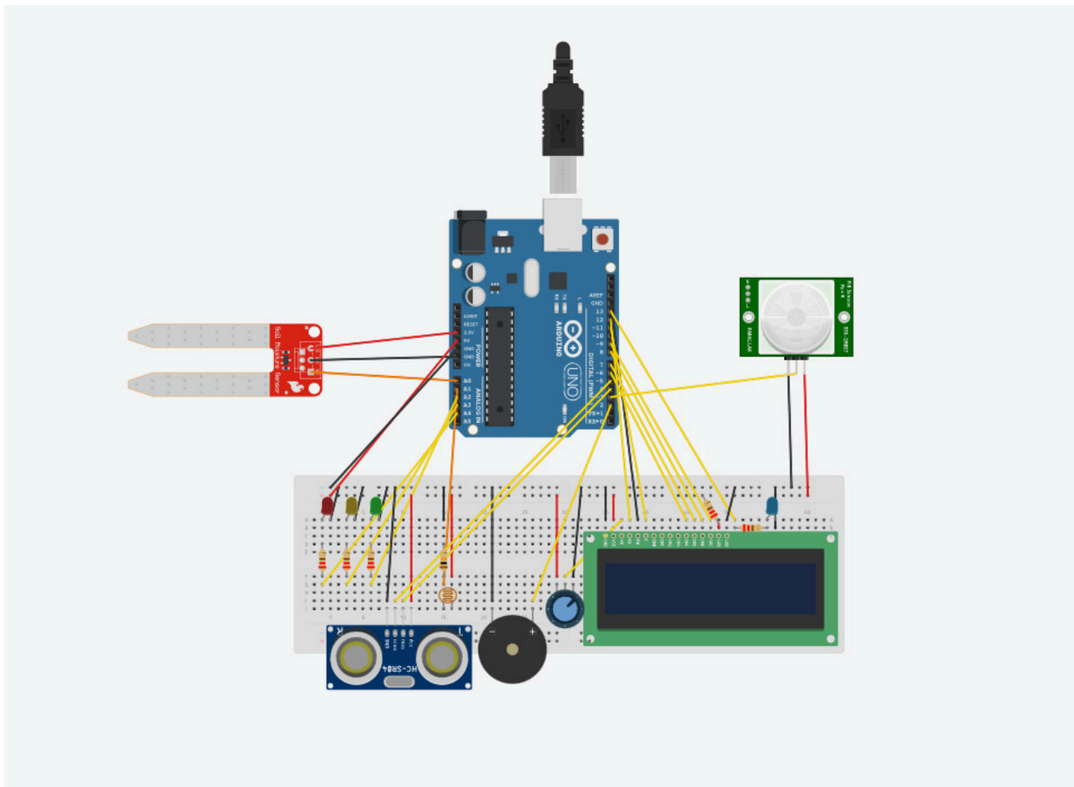


Circuit Board Configuration

To ensure that the board works well (the circuit is closed, correct currents and voltages are set to each component, necessary connection is made and the components are working), simulation on software application is conducted. The simulation was taken place only in TinkerCAD (www.tinkercad.com) although circuit board configuration were done on both TinkerCAD and Fritzing as well. Link to the board on TinkerCAD is: https://www.tinkercad.com/things/4Bric4f2zdT-arturitozulfikar-sepyan/editel?sharecode=C7pGVBI0aVoO3drpbGcjR-WeWqLMva66GQIUtVi4JYGk



Circuit board in Fritzing

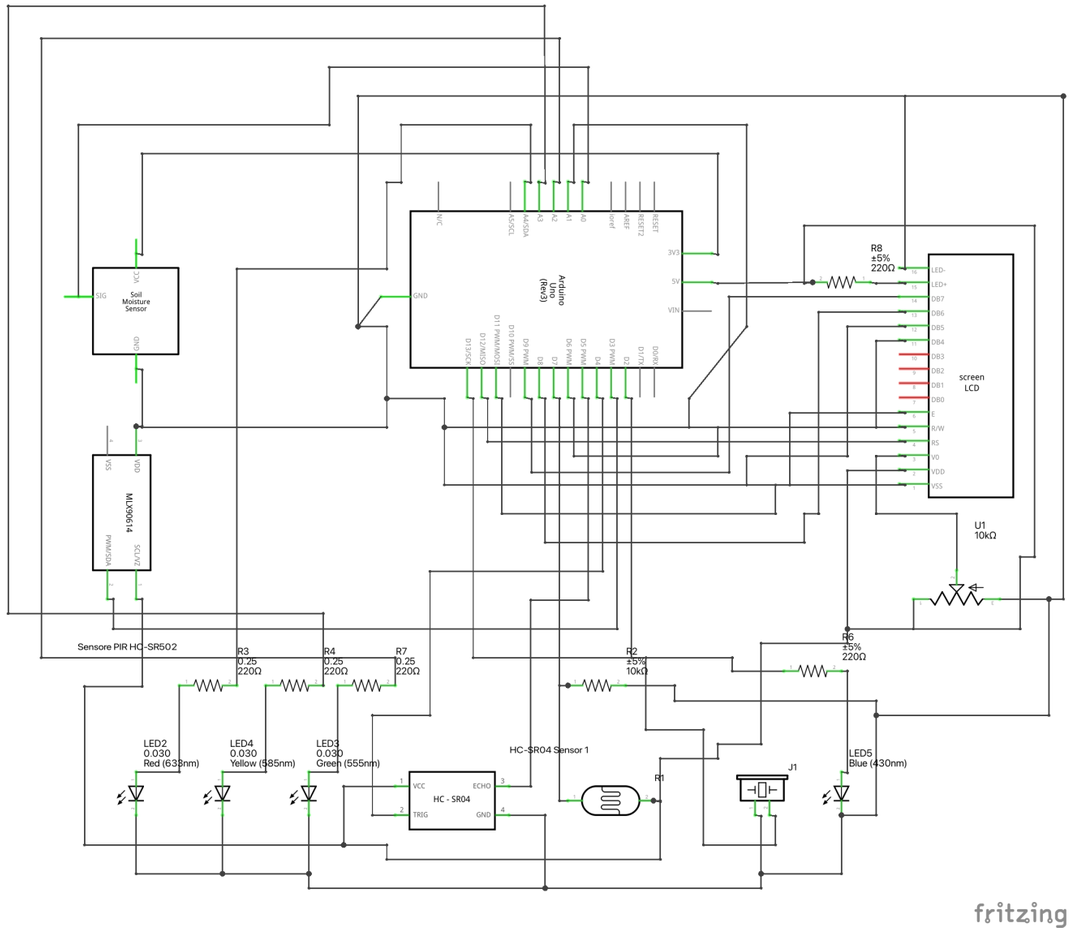


Circuit board in TinkerCAD

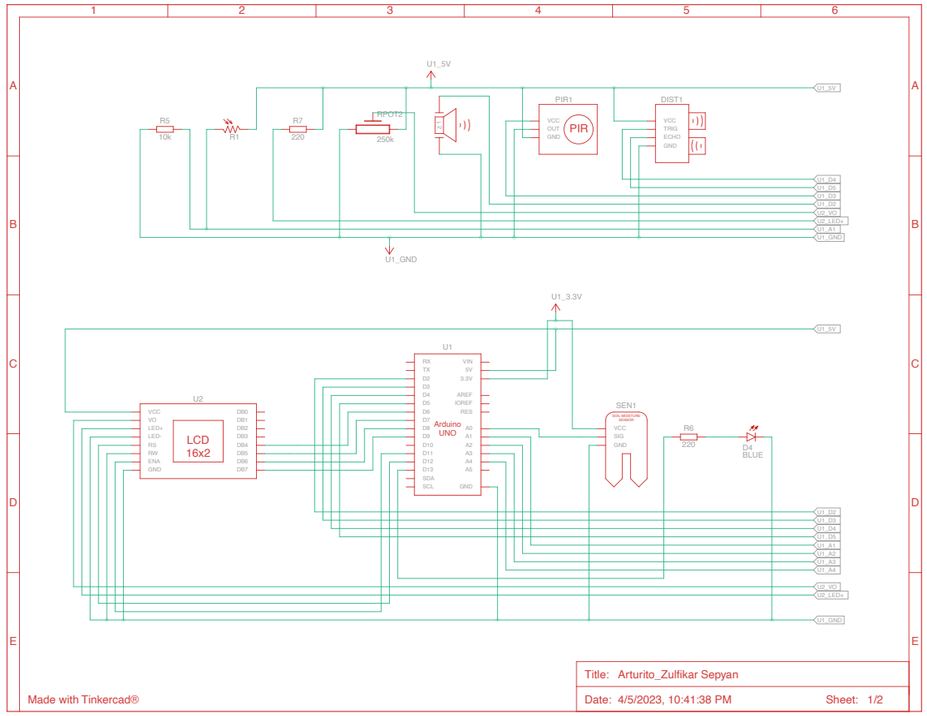
*notes after conducting the simulation: Although all parts are proved to work, the simulation in TinkerCAD took too long time (1s in TinkerCAD not equal to 1s in real life) and the sound of piezzo used was different with the passive buzzer that we use in the real design.

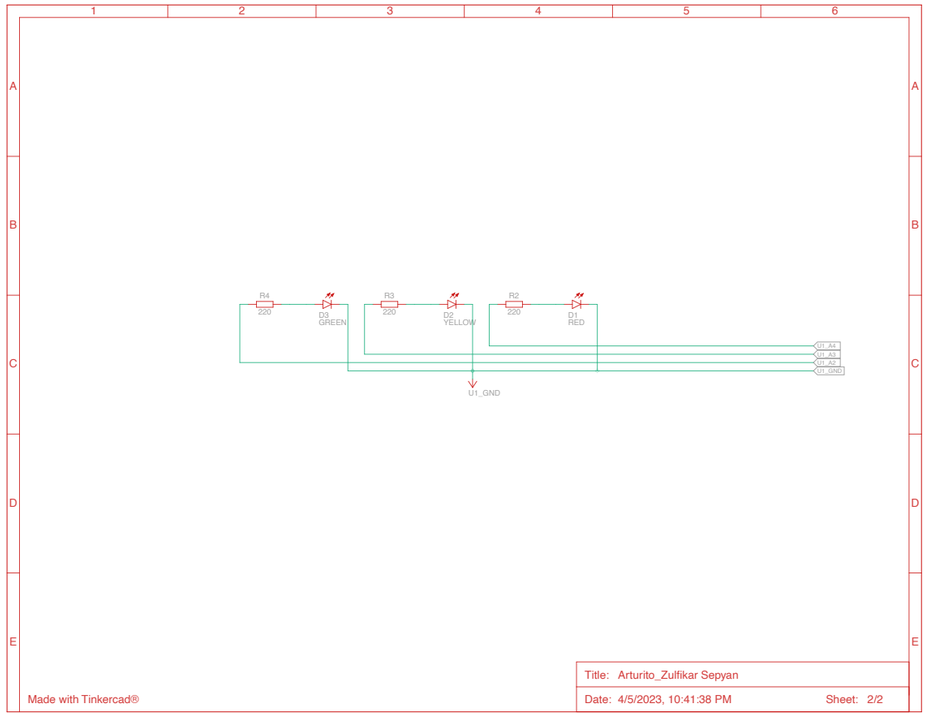
Schematic Diagram

With the following circuit board configuration, a schematic diagram is generated automatically using both Fritzing and TinkerCAD.



Schematic Diagram in Fritzing





Schematic Diagram in TinkerCAD

The Arduino Code 1/4

```
// Resistive Soil Moisture Sensor + LCD Output + buzzer + PIR
Sensor + LEDs + URM Sensor + Photoresistor for plant
communication device
// author: Zulfikar Sepyan
// link to TinkerCAD circuit board:
// https://www.tinkercad.com/things/4Bric4f2zdT-copy-of-
arturitozulfikar-sepyan/editel?
sharecode=C7pGVBI0aVoO3drpbGcjRWeWqLMva66GQIUtVi4JYGk

#include <LiquidCrystal.h>                              //
make LCD output available
LiquidCrystal LCD(12, 11, 6, 7, 8, 9);                  //
digital pin on the Arduino for LCD

// greeting messages program memory
const byte greetingMessages = 15;
char* names[greetingMessages] = {
  "Hello, you!",
  "Bipbipbip bepbop",
  "No doubt",
  "You look good",
  "Enough water, Pa",
  "Enough water, Ma",
  "All too well",
  "Coffee at 9?",
  "Hahahaha funny",
  "Yes",
  "Ask again later",
  "Good day init?",
  "Better call Saul",
  "Ask GPT!",
  "I am good"
};

// pins
const int moisturePin = A0;
const int buzzerPin = 2;
const int PIRPin = 3;
const int blueLedPin = 13;
const int greenLedPin = A2;
const int yellowLedPin = A3;
const int redLedPin = A4;
const int URMTrigPin = 4;
const int URMEchoPin = 5;
const int photoresistorPin = A1;

// additional variables for PIR Sensor
int percentageCurr = 100;                               //
initial value
const int calibrationTime = 30;

// additional variables for photoresistor
int sensorValue;                                        // to
hold sensor value
int sensorLow = 1023;                                   // to
calibrate low value
int sensorHigh = 0;                                     // to
calibrate high value

// additional variables for soil moisture sensor
int moistureValue;                                      // to
hold sensor value
int moistureLow = 0;                                    //
minimum level of soil moisture
int moistureHigh = 740;                                 //
maximum level of soil moisture as what the sensor read

// additional variables for URM sensor
long duration;
int distance;
int distanceLow = 400;                                  //
maximum range of URM sensor is 400cm
int distanceHigh = 2;                                   //
minimum range of URM sensor is 2cm

void setup() {

  Serial.begin(9600);                                   //
start communication with Serial Monitor
  LCD.begin(16, 2);                                     //
start communication with LCD 16 by 2 characters

  // digital pins setup
  pinMode(PIRPin, INPUT);                               // set
PIR sensor as the input
  pinMode(buzzerPin, OUTPUT);                           // set
buzzer as the output
  pinMode(blueLedPin, OUTPUT);                          // set
the led as the output
  pinMode(greenLedPin, OUTPUT);                         // set
the led as the output
  pinMode(yellowLedPin, OUTPUT);                        // set
the led as the output
  pinMode(redLedPin, OUTPUT);                           // set
the led as the output
  pinMode(URMTrigPin, OUTPUT);                          // set
the URM sensor's trig as the output
  pinMode(URMEchoPin, INPUT);                           // set
the URM sensor's echo as the input
```

The Arduino Code 2/4

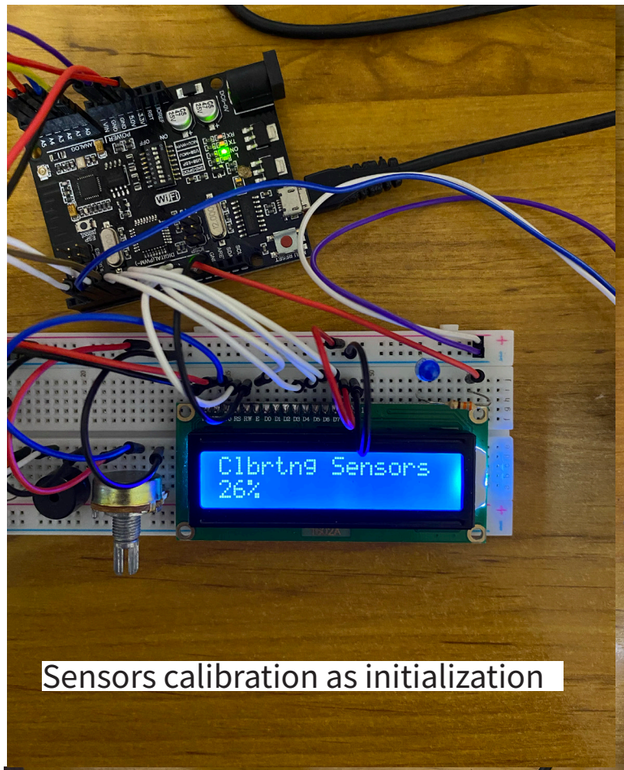
```
  randomSeed(analogRead(0));                                  // to
generate random number

  // sensors calibration (for 30 seconds)
  digitalWrite(PIRPin, LOW);                                  //
initial condition of PIR sensor
  LCD.setCursor(0, 0);                                        //
show calibration status on LCD
  LCD.print("Clbrtng Sensors");
  while (millis() < 30000) {                                  //
millis 30000 = 30s. do this loop for sensor calibration in 30s
duration.
    for (int i = 0; i < calibrationTime; i++) {

      sensorValue = analogRead(photoresistorPin);
      if (sensorValue > sensorHigh) {

        sensorHigh = sensorValue;                             // the
highest reading during calibration become maximum number of
photoresistor
      }
      if (sensorValue < sensorLow) {

        sensorLow = sensorValue;                              // the
lowest reading during calibration become minimum number of
photoresistor
      }

      int process = map(i, 0, 30, 0, 100);                    //
display percentage of calibration process on the LCD
      LCD.setCursor(0, 1);
      LCD.print(process);
      LCD.print("%");
      delay(1000);
    }
  }
  LCD.clear();
  LCD.print("Calibration Done");                              //
show that calibration has done on the LCD
  delay(1000);
  LCD.clear();
  delay(50);
}

// tone randomizer mode 1: low freq --> high freq == happy
tone. random function will choose one number between the
range.
void phrase1() {

  int k = random(1000,2000);
  digitalWrite(blueLedPin, HIGH);
  for (int i = 0; i <= random(100,2000); i++){

    tone(buzzerPin, k + (-i * 2));
    delay(random(.9,2));
  }
  digitalWrite(blueLedPin, LOW);
  for (int i = 0; i <= random(100,1000); i++){

    tone(buzzerPin, k + (i * 10));
    delay(random(.92));
  }
}

// tone randomizer mode 2: high freq --> low freq == sad tone
void phrase2() {

    int k = random(1000,2000);
    digitalWrite(blueLedPin, HIGH);
    for (int i = 0; i <= random(100,2000); i++){

        tone(buzzerPin, k + (i * 2));
        delay(random(.9,2));
    }
    digitalWrite(blueLedPin, LOW);
    for (int i = 0; i <= random(100,1000); i++){

        tone(buzzerPin, k + (-i * 10));
        delay(random(.9,2));
    }
}

void measure_it() {                                           //
function for URM sensor to read distance
  digitalWrite(URMTrigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(URMTrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(URMTrigPin, LOW);

  duration = pulseIn(URMEchoPin, HIGH);
  distance = duration * 0.034/2;                              //
divided by speed of sound in microsecond
}

void loop() {

  // map the highest value and the lowest value of sensor
readings into the specified ranges
```

The Arduino Code 3/4

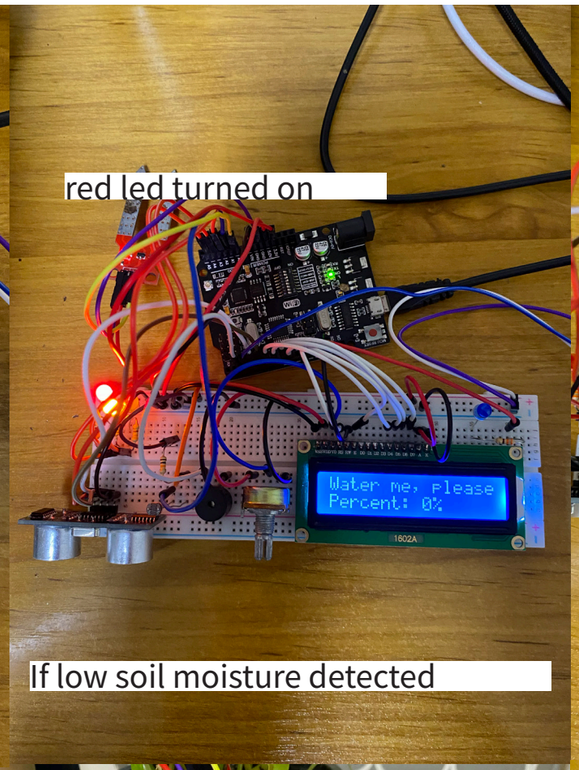
```
  moistureValue = analogRead(moisturePin);
  int percentage = map(moistureValue, moistureLow,
moistureHigh, 0, 100);
  sensorValue = analogRead(photoresistorPin);
  int pitchDivider = map(sensorValue, sensorLow, sensorHigh,
10, 1); // it will become "divider", and bright result will
divide smaller than dim result. thus it is mapped this way.

  digitalWrite(yellowLedPin, LOW);                    //
initial state of yellow led is off for all moisture condition,
unless it is dim or cloudy
  if (pitchDivider > 5) {                             // if
it is dim or cloudy, yellow led turned on

    digitalWrite(yellowLedPin, HIGH);
  }
  LCD.setCursor(0, 1);                                //
show moisture sensor reading in percentage on the LCD
  LCD.print("Percent: ");
  LCD.print(percentage);
  LCD.print("%");

  if (percentage <= 40) {                              // if
soil moisture level lower than 20%

    digitalWrite(greenLedPin, LOW);
    digitalWrite(redLedPin, HIGH);                    // red
led turned on as the indicator when soil moisture level is low
    LCD.setCursor(0, 0);
    LCD.print("Water me, please");                    // the
"water me" message when soil moisture level is low

    if (digitalRead(PIRPin) == HIGH) {                //
(additionally) if someone around when the soil moisture level
is low. else, it won't do nothing more.
      measure_it();                                   //
measure the distance with the person approaching
      // map the measured distance to 1 - 10 range to be used
as delay multiplier for sound variation. addtnly, distance >
100 will be regarded as 10.
      int distancePercent = map(distance, 2, 100, 1, 10);

      // produce the sound --> sound of R2D2
      // 2 parts of R2D2 sound
      // 1st part: tone randomizer function called (phrase 1
and phrase 2). switch function will choose 1 of 7 variations
provided.
      switch (random(1,7)) {

        case 1:phrase1(); break;
        case 2:phrase2(); break;
        case 3:phrase1(); delay(distancePercent); phrase2();
break;
        case 4:phrase1(); delay(distancePercent); phrase2();
delay(distancePercent); phrase1(); break;
        case 5:phrase1(); delay(distancePercent); phrase2();
delay(distancePercent); phrase1(); delay(distancePercent);
phrase2(); delay(distancePercent); phrase1(); break;
        case 6:phrase2(); delay(distancePercent); phrase1();
delay(distancePercent); phrase2(); break;
      }

      // 2nd part:
      for (int i = 0; i <= random(3, 9); i++){

        digitalWrite(blueLedPin, HIGH);
        tone(buzzerPin, random(300, 4000) / pitchDivider);
        delay(random(70, 100) * pitchDivider / 5);
        digitalWrite(blueLedPin, LOW);
        noTone(buzzerPin);
        delay(random(1, 30) * distancePercent);
      }
      noTone(buzzerPin);
      delay(random(100, 200) * distancePercent);
    }
  }

  else if (percentage - percentageCurr > 30) {          // if
there is a sudden rise in soil moisture level --> assumably,
someone water it

    digitalWrite(greenLedPin, HIGH);                    //
green led turn on when soil moisture level is good
    digitalWrite(redLedPin, LOW);
    LCD.setCursor(0, 0);
    LCD.print("Thank you");                             //
show "thank you" message on the LCD when the plant is watered
    LCD.setCursor(0, 1);
    LCD.print("so much, mate!");

    for (int i = 0; i < 1; i++){                        //
play the sound (do re mi fa sol la si do) when the plant is
watered

      tone(buzzerPin, 251.626); delay(200); noTone(buzzerPin);
delay(100);
      tone(buzzerPin, 294.33); delay(200); noTone(buzzerPin);
delay(100);
      tone(buzzerPin, 327.03); delay(200); noTone(buzzerPin);
delay(100);
```

The Arduino Code 4/4

```
      tone(buzzerPin, 348.83); delay(200); noTone(buzzerPin);
delay(100);
      tone(buzzerPin, 392.44); delay(200); noTone(buzzerPin);
delay(100);
      tone(buzzerPin, 436.04); delay(200); noTone(buzzerPin);
delay(100);
      tone(buzzerPin, 490.55); delay(200); noTone(buzzerPin);
delay(100);
      tone(buzzerPin, 523.25); delay(200); noTone(buzzerPin);
delay(100);
    }
    delay(2000);
  }

  else {                                                    //
(percentage >= 40) if soil moisture is in the good level

    digitalWrite(redLedPin, LOW);
    digitalWrite(greenLedPin, HIGH);                        //
green led turned on when soil moisture level is good
    LCD.setCursor(0, 0);
    LCD.print("Soil Moisture");                             //
show this character on the first line of the LCD

    if (digitalRead(PIRPin) == HIGH) {                      //
(additionally) when someone is around when the soil moisture
level is good. else, do nothing more.

      if (pitchDivider <= 5) {                              //
when it is BRIGHT, someone is around, and soil moisture is
good

        LCD.setCursor(0,0);
        LCD.print(names[random(0,greetingMessages)]);       //
show "greeting message" on the LCD that chose randomly from
the program memory
        phrase1();                                          //
happy tone played = phrase 1
        delay (200);
      }

      else {                                                //
(pitchDivider > 5) but when the surrounding is DIM instead

        digitalWrite(greenLedPin, LOW);                     // set
the green led off so that the yellow led as light indicator
can catch the attention
        LCD.setCursor(0,0);
        LCD.print("A bit dark init?");                      //
show "worry message" on the LCD
        phrase2();                                          // sad
tone played = phrase 2
        delay (200);
      }

      noTone (buzzerPin);
      delay(2000);
    }
  }
  percentageCurr = percentage;                              //
store percentage value in another variable
  delay(1000);
  LCD.clear();                                              //
clear the LCD from any output
}
```

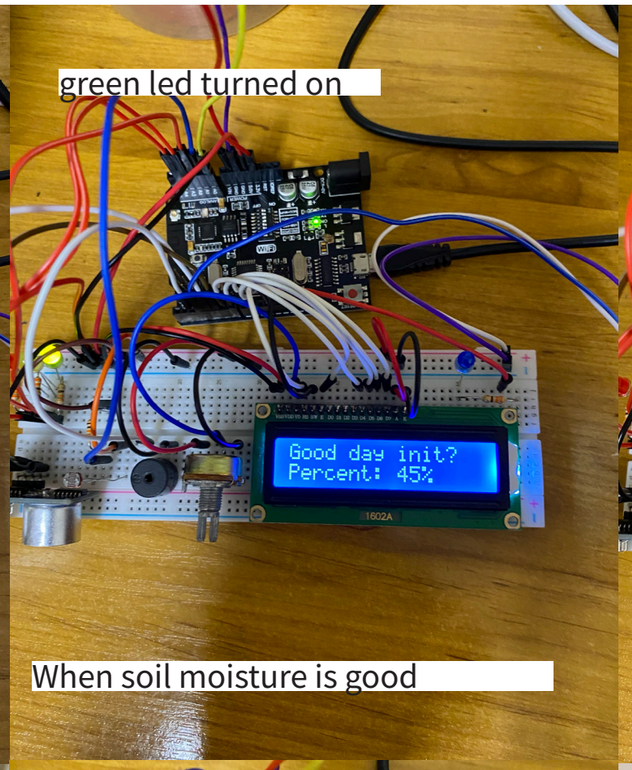


Sensors calibration as initialization



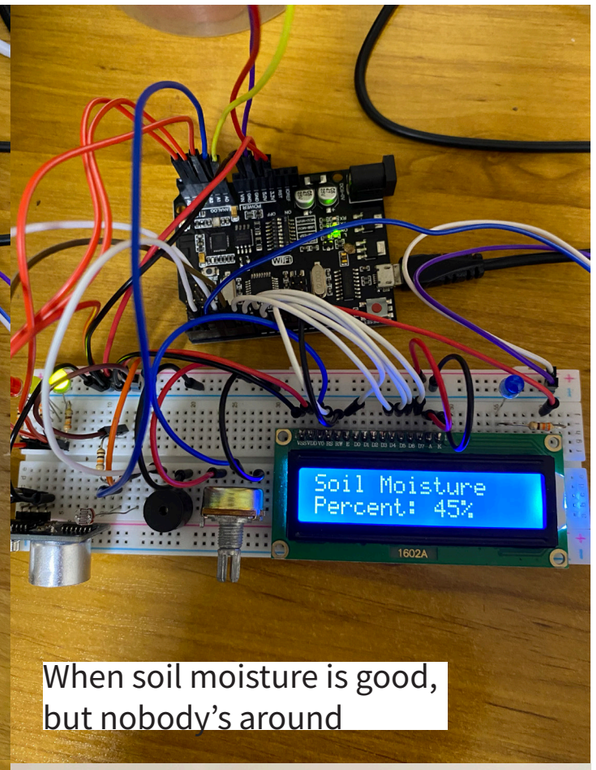
red led turned on

If low soil moisture detected

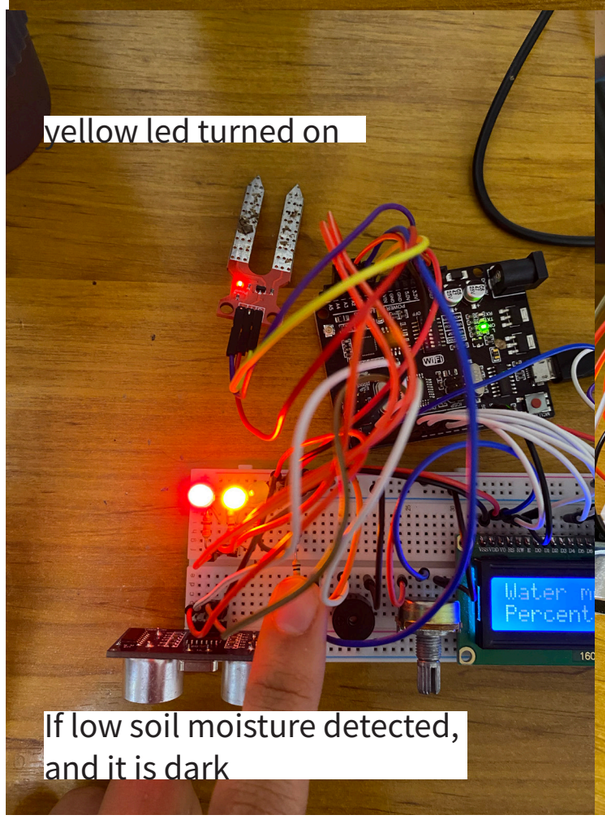


green led turned on

When soil moisture is good

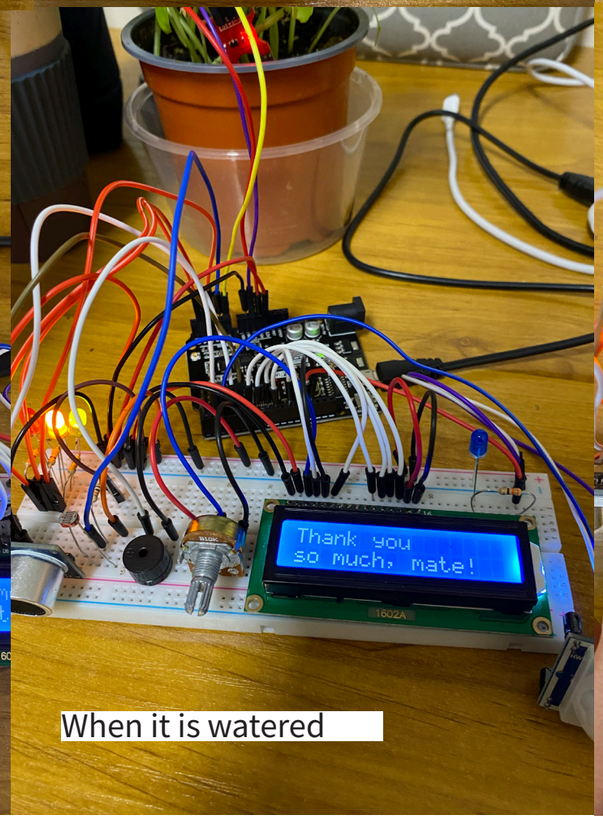


When soil moisture is good, but nobody's around

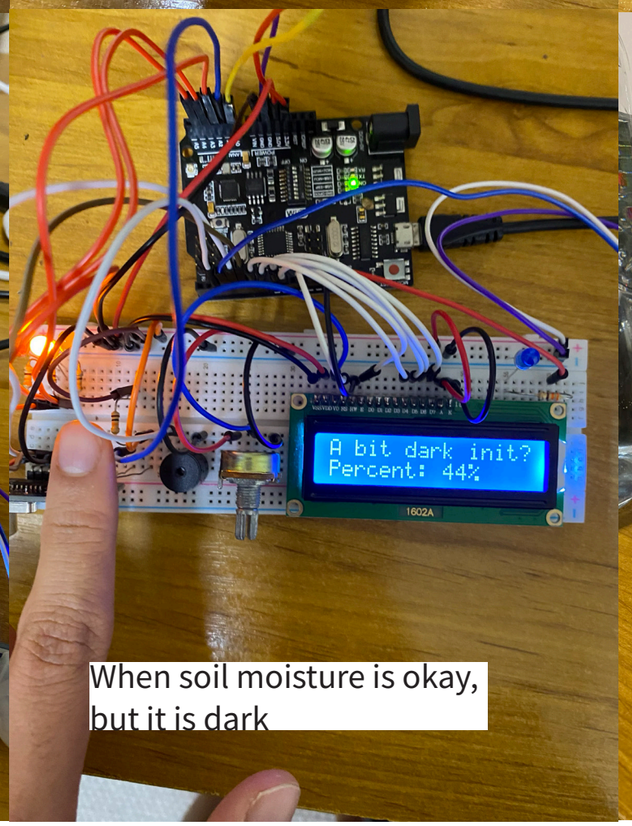


yellow led turned on

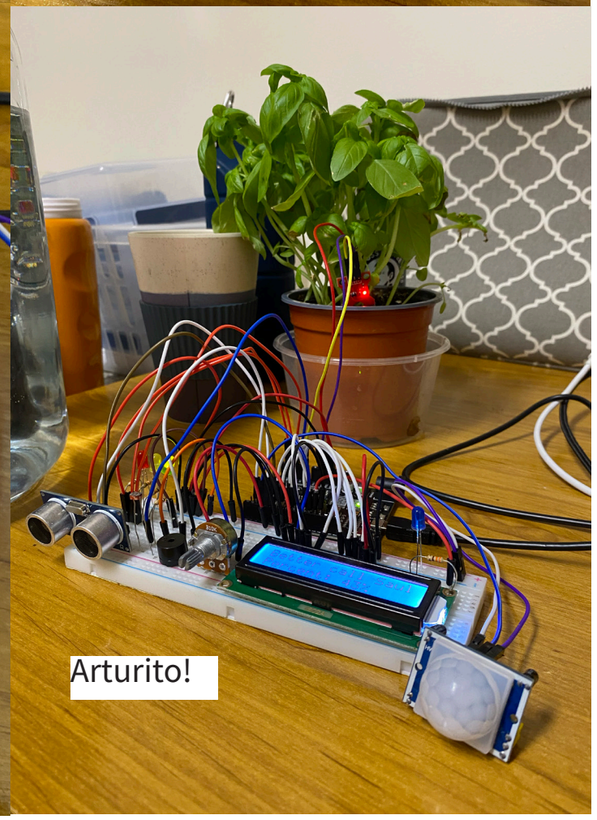
If low soil moisture detected, and it is dark



When it is watered



When soil moisture is okay, but it is dark



Arturito!

References

Arduino (21 May 2020) '06 Starter Kit: Light Theremin' [video], *Arduino*, YouTube website, accessed 2023/04/05/10:47:49. https://www.youtube.com/watch?v=DnD92Q_Kpac

Arduino Forum (2009) 'How to print random text', Arduino Forum, accessed 2023/04/05/11:09:47. https://forum.arduino.cc/t/how-to-print-random-text/42163

Artist unknown (n.d.) 'R2-D2 | Wookieepedia | Fandom', accessed 2023/04/05/13:08:45. https://starwars.fandom.com/wiki/R2-D2

Circuit Magic (6 July 2015) 'PIR Motion Sensor with Arduino' [video], *Circuit Magic,* YouTube website, accessed 2023/04/05/11:01:44. https://www.youtube.com/watch?v=FxaTDvs34mM

How To Mechatronics (27 July 2015) 'Ultrasonic Sensor HC-SR04 and Arduino Tutorial' [video], *How To Mechatronics*, YouTube website, accessed 2023/04/05/11:03:38. https://www.youtube.com/watch?v=ZejQOX69K5M

Larios M 2023, R2D2-Sound-Generator, 2023/02/27/T02:03:25Z, https://github.com/marcelolarios/R2D2-Sound-Generator.

Mr. Arduino (29 February 2020) 'How to use soil moisture sensor using LCD display with arduino' [video], *How To Mechatronics*, YouTube website, accessed 2023/04/05/11:08:39. https://www.youtube.com/watch?v=xMeZTZqjugc