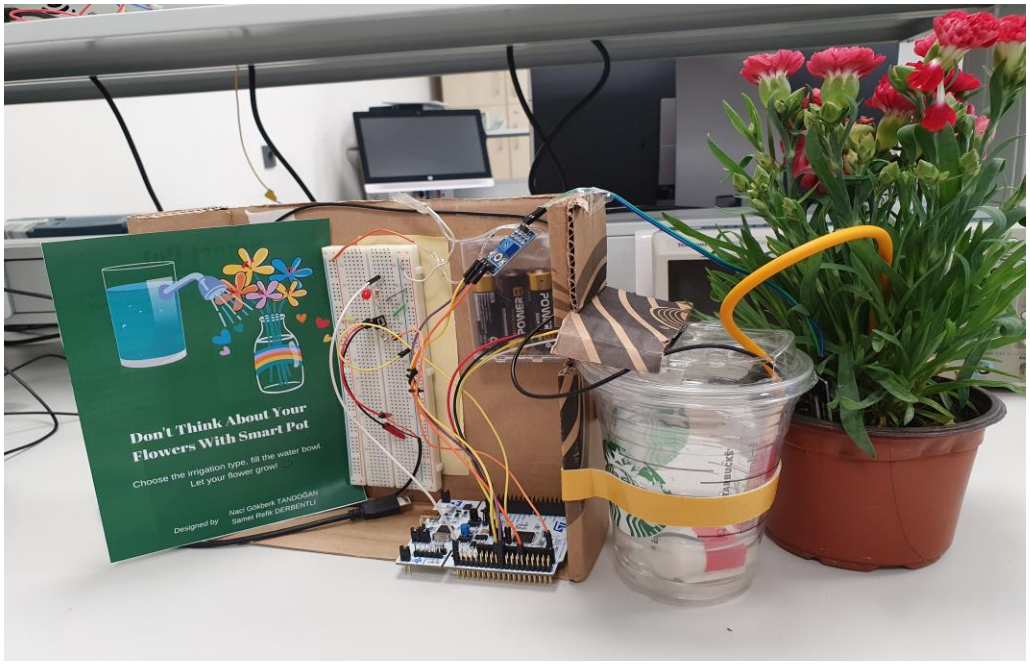
MEF UNIVERSITY

EE308 EE ENGINEERING DESIGN STUDIO

SMART POT


Don't Think About Your Flowers With Smart Pot
Choose the irrigation type, fill the water bowl. Let your flower grow!
Designed by Naci Gökberk TANDOĞAN Samet Refik DERBENTLI

GROUP 4B

041802049 NACI GÖKBERK TANDOĞAN

041902001 SAMET REFIK DERBENTLI

1. Introduction

Nowadays, there is a large group of people who like to grow flowers in pots at home. But often people do not know or forget when to water these flowers and how much. Thanks to the irrigation system we will make, we will both save water and ensure the healthy growth of our flowers by trying to achieve maximum efficiency with the minimum water we use. First, we will place a humidity sensor in our flowerpot, and we will get the information from the sensor instantly with the help of our microcontroller. With the help of the ultrasonic sensor of the water amount in the water tank, the amount of water in the water tank is regularly informed and the water pump is informed. When the water level reaches a level that will damage the water pump (about 20% remaining), the red led will turn on and stop the pump until you add water again. In this way, damage to the pump will be prevented and the user will understand that user needs to add water when user sees the red led on. This information will be sent to the microcontroller regularly with the help of the ultrasonic sensor placed on the water bottle. When the soil moisture level drops below the optimum level, the microcontroller receives the signal from the soil moisture sensor, controls the ultrasonic sensor (checks if there is enough water in the water tank), and triggers the DC water pump, which supplies water to the area. We use batteries for circuit feeding because the biggest advantage of our smart pot module is that it is portable and can be used anywhere.

1.1 Functional Description

The Smart Pot is designed for small and medium pots, which are found in homes and workplaces. These flowers are violet, begonia, poinsettia, carnation, etc. are species. Our smart pot has two different modes, the first mode is the growing mode, where we aim to fix the soil moisture to 45%. the second mode is the flowering mode where we aim to fix the soil moisture to 65%.

The user has to use growing mode until the flower in the pot starts to bloom, then switch to flowering mode (with the help of Button 1) after it starts to bloom.

1.2 Design Approach

We designed our project is three steps.

1.2.1 Design of humidity sensor

As a first step, we looked at the projects where the humidity sensor is used to understand the working logic. We tried to understand the working logic by testing the humidity sensor by taking 3 different soils with different moisture rates. After understanding the working logic, we realized that it works with different sensitivity in different environments. For example, a damp napkin did not work with the same precision in water or damp soil, and we had to adjust it ourselves. That's why we adjusted the sensitivity of the humidity sensor according to the soil we will use.

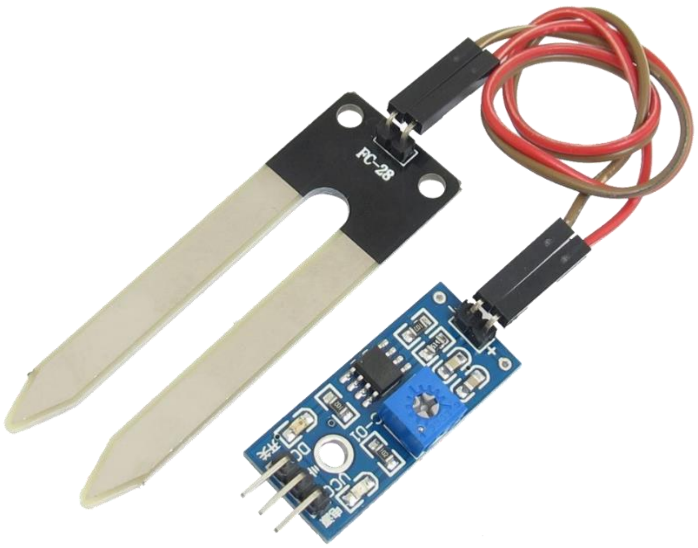


Figure 1. Humidity Sensor [1]

1.2.2 Design of ultrasonic sensor

Secondly, we adjusted the sensitivity of the ultrasonic sensor. Before doing this, we found a water bottle and placed our water pump in it. We changed our code according to the value that the ultrasonic sensor gives when there is a water pump inside. We designed a software to stop irrigation when the ultrasonic sensor water level reaches the same level as the water pump.

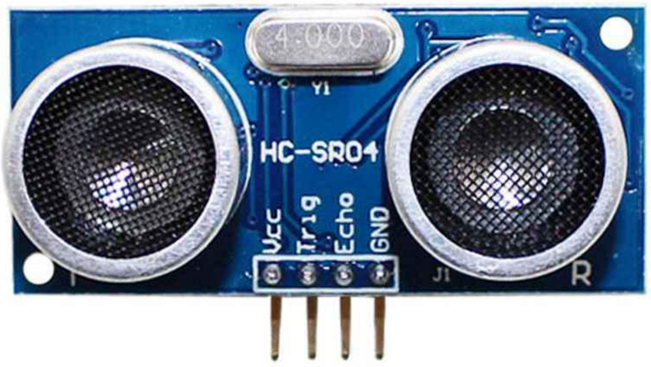


Figure 2. Ultrasonic Sensor [2]

1.2.3 Design of water pump

Thirdly, we worked on the actuation and triggering of the water pump. First, we created a design so that the pump would get its power from the microcontroller, but the current drawn from the microcontroller was not enough to start the motor. That's why we built a small driver circuit with the help of the BD140 transistor and fed the pump with 3 AA batteries (4.5V). We used the digital output value from the controller as a switch.

Figure 3. 5V Water Pump [3]

2. Hardware Design

2.1 High level architecture

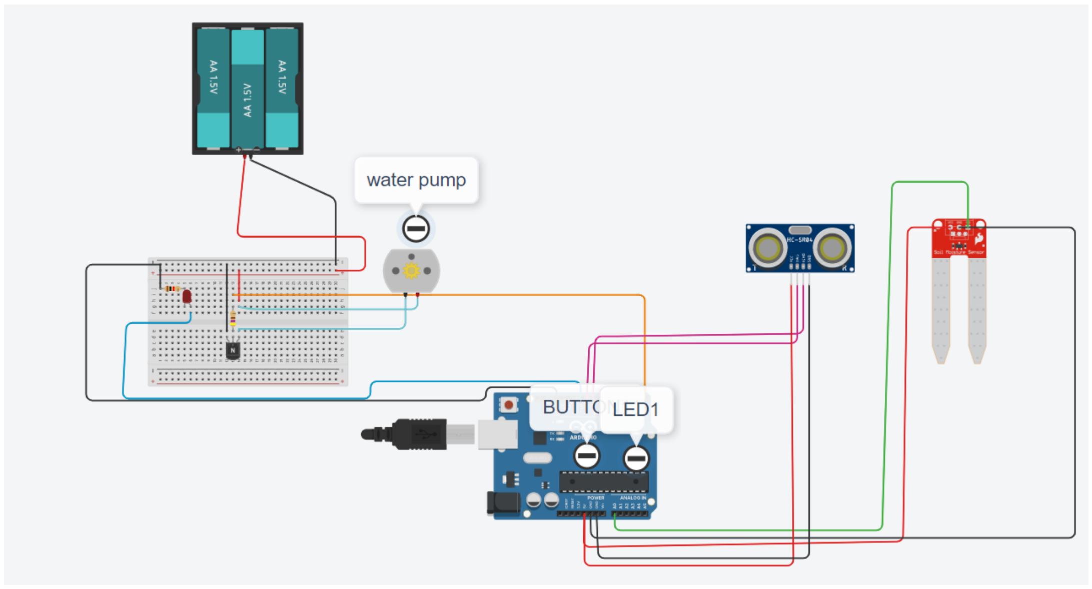


Figure 4. High Level Architecture

Ultrasonic Sensor:

The Ultrasonic sensor Trigger pin is connected to PC_8 on our microcontroller.

The ultrasonic sensor Echo pin is connected to PC_9 on our microcontroller.

The feeding and grounding of the ultrasonic sensor is done through our microcontroller.

Humidity Sensor:

The humidity sensor analog output pin is connected to the PA_0 pin of the microcontroller.

The feeding and grounding of the humidity sensor is done via the microcontroller.

Water Pump:

Both cables of the water pump are connected to the collector leg of the BD 140 transistor, and we are passing the supply from 4.5V batteries.

For the operation of the pump, we get the output from the microcontroller from the PC_0 pin.

PC_0 output from the pin is connected to the Base leg of our BD 140 transistor. The emitter leg of the BD 140 is connected to ground.

Red Led:

The positive pin of the red LED is connected to the PC_10 pin of the microcontroller. The negative leg is grounded with a 1k resistor.

Button 1 and Led 1:

Button 1 is a button on the microcontroller. With the help of this button, we can switch between 2 modes of our flowerpot. While LED 1 on the microcontroller lights green in growing mode, it lights red in flowering mode.

Since our flowerpot is portable and portable, we fed our microcontroller with a power bank.

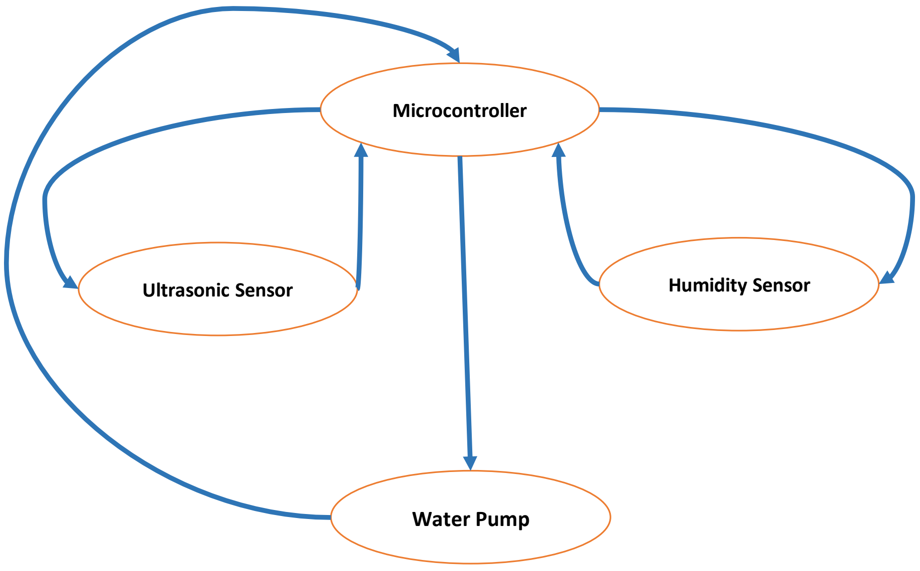


Figure 5. Block Diagram

2.2 Sensor excitation circuit

We used BD140 transistor in our design. The reason we chose this transistor is because we want to make an NPN switch. Since our motor resistance is about 10 ohms (R_c), we also used our R_b resistance as 470 ohms. We added the switch design we made in Figure 1. While we took the Vin voltage from the PC_0 output of our microcontroller, we supplied the V_{cc} from 3 pieces of 1.5V AA batteries.

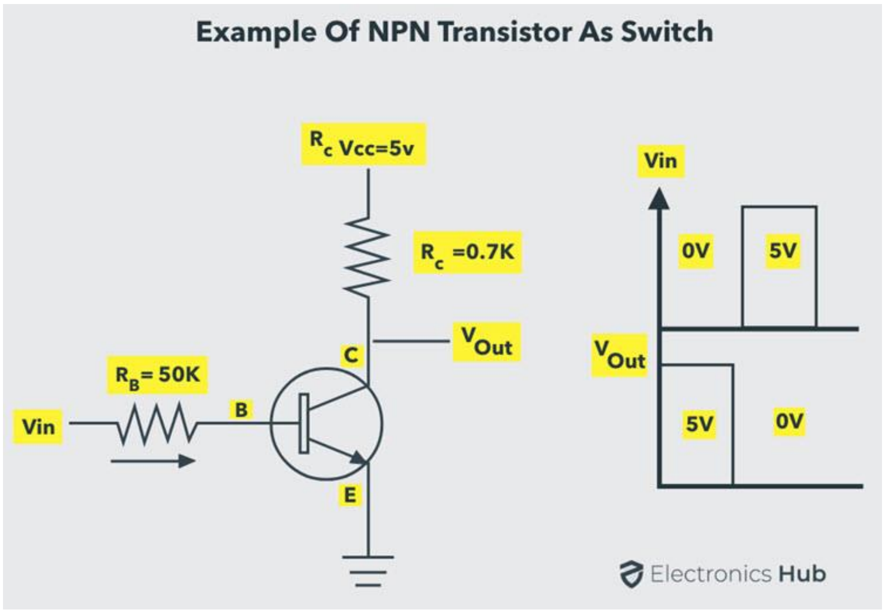


Figure 6. NPN Switch Transistor [4]

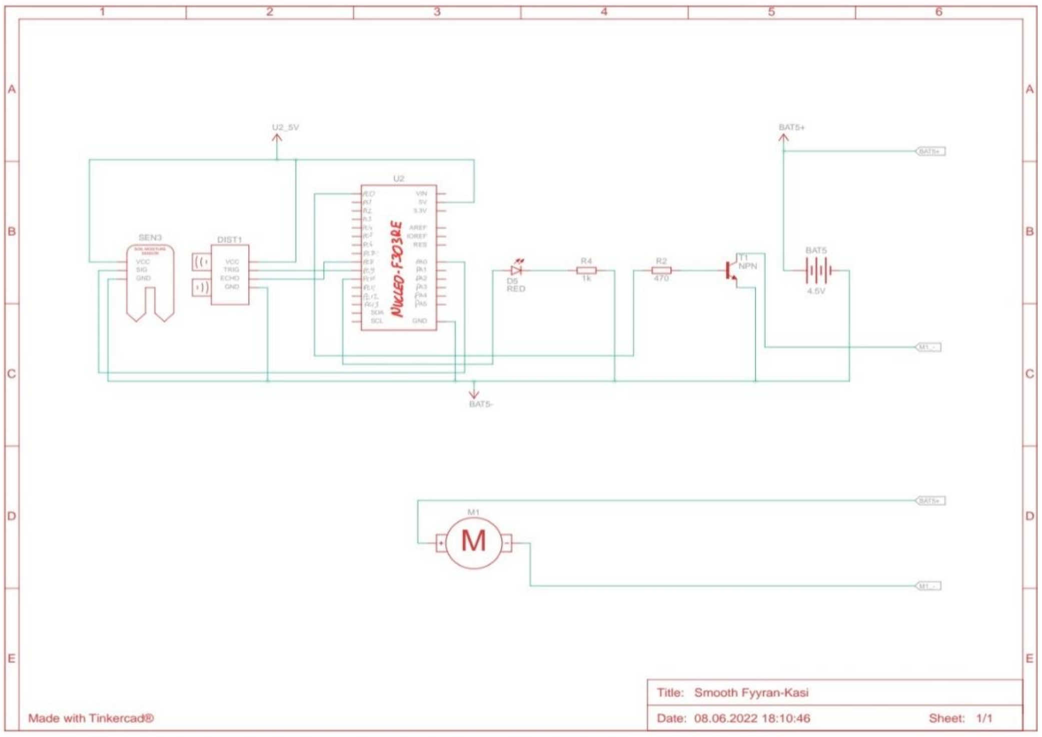
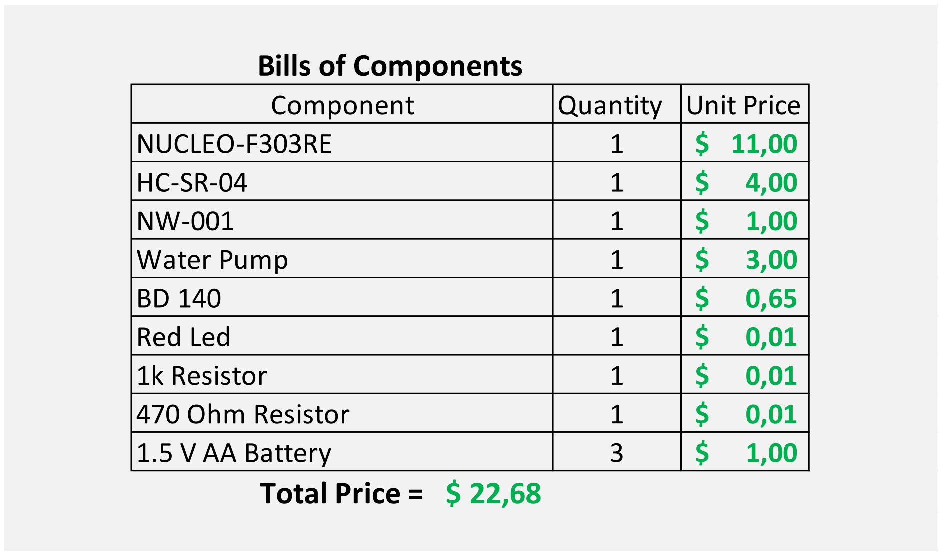


Figure 7. Schematic

2.3 Components



Bills of Components

Component	Quantity	Unit Price
NUCLEO-F303RE	1	$ 11,00
HC-SR-04	1	$ 4,00
NW-001	1	$ 1,00
Water Pump	1	$ 3,00
BD 140	1	$ 0,65
Red Led	1	$ 0,01
1k Resistor	1	$ 0,01
470 Ohm Resistor	1	$ 0,01
1.5 V AA Battery	3	$ 1,00

Total Price = $ 22,68

Table 1. Price Table and Components

3. Software Design

We added Flowchart to appendix A. However, if we need to explain, the system is run first. The humidity mode of the pot is selected, if the soil is at the desired humidity level, the humidity level is continuously controlled by entering the loop. When the humidity falls below the desired level, the water level is controlled. If there is enough water, irrigation is done, if not, the engine is stopped until water is added to the water bottle, and when water is added, it starts to water. Water for about 3 seconds, wait 15 minutes for humidity control, and water again when necessary.

4. Integration and Tests

4.1 Peripheral Tests

We tested our pot in growing mode (45% humidity).

4.1.1 Scenario 1

As long as the humidity was below 45%, pump irrigated and waited for 15 minutes. When the humidity increased above 45%, it finished the irrigation phase and returned to the control loop. We did not encounter any problems.

4.1.2 Scenario 2

In this scenario, we left the humidity of the soil above 45% and operated our pot in that way. The humidity sensor constantly checked the soil moisture and did not irrigate as it did not fall below 45%.

4.1.3 Scenario 3

In this scenario, we kept the soil moisture below 45% but the water level below 20% (lethal level for the water pump). As we expected, the red led turned on and the engine did not start under any circumstances.

When we added water to the water container, the engine started to water and returned to its normal order. As we designed, we did not encounter any problems in this scenario either.

5. Summary and Conclusions

Thanks to the project we did, we learned better the working principles and logic of the sensors we use. We have seen how we can use sensors integrated with each other.

We saw how the logic of a simple driver circuit is and how we can implement it with a transistor. At the same time, we have seen that when the hardware and software part of a project is properly integrated, we are less likely to get errors and we can get a clearer result.

The most difficult part of our project was the engine riding part. Because the amperage value we got from the microcontroller was low, it was not enough to start the motor of the water pump. For this reason, we first set up a basic driver circuit with a small transistor and fed it with a battery. But since we put a small transistor, we could not pass the high current we needed, and the transistor burned out. For this reason, we decided to put the BD 140 transistor with a relatively high current pass, and we used the BD 140.

To improve the project, special humidity modes for each flower can be customized irrigation according to the flowers. By adding a Bluetooth or Wi-Fi module to the microcontroller, the user's instant moisture status of the soil, the remaining water volume in the water bottle can be monitored instantly from the phone. Thanks to these modules, when the water level reaches a level that can damage the pump, the user can be asked to fill in water by sending a notification.

6. References

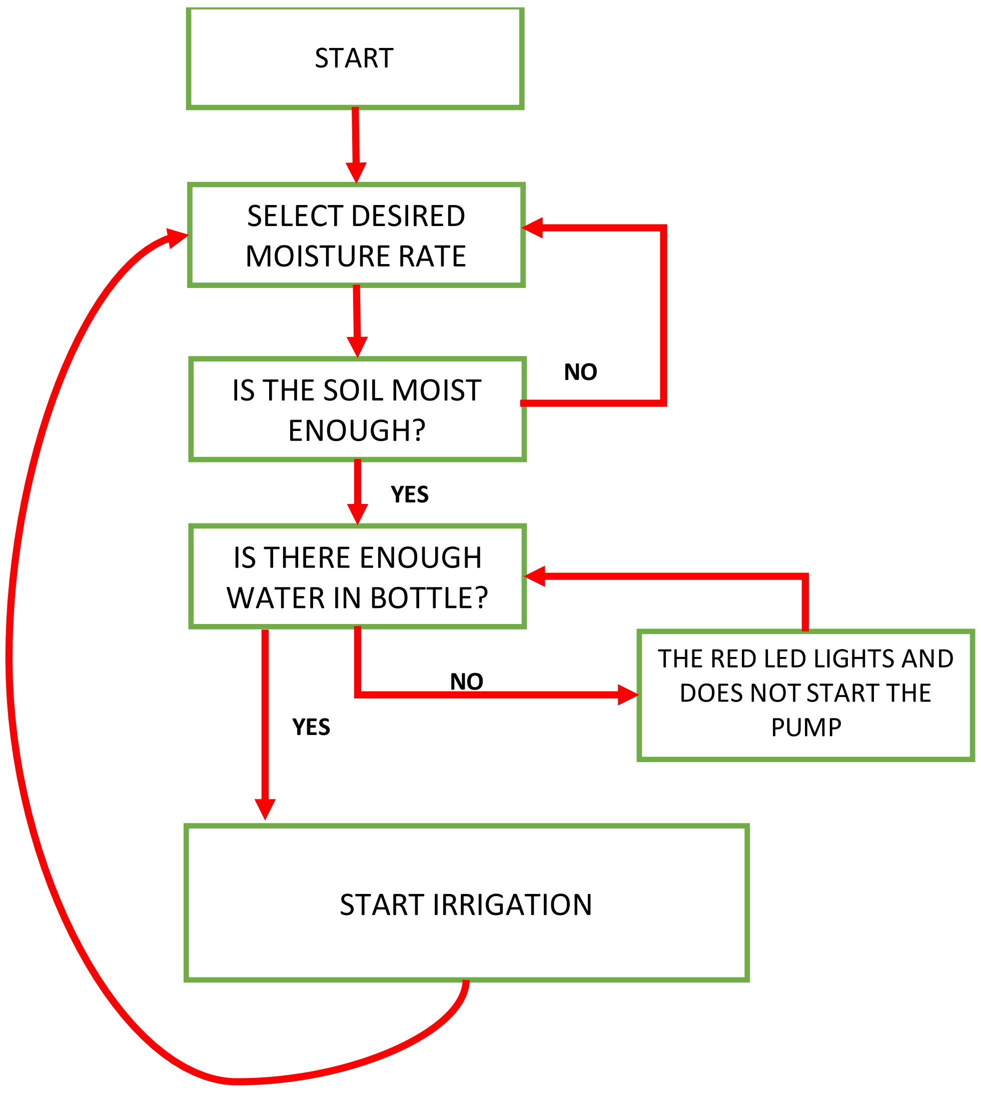
[1] "Open Circuit," [Online]. Available: https://opencircuit.shop/product/ground-humidity-sensor-module.

[2] "Joom," [Online]. Available: https://www.joom.com/tr/products/1508479121355003831-90-1-709-1817041862.

[3] "Robit Shop," [Online]. Available: https://www.robitshop.com/urun/mini-dalgic-su-pompasi-dc-2-5v-6v-120l-h.

[4] R. Teja, 9 4 2021. [Online]. Available: https://www.electronicshub.org/transistor-as-a-switch/.

Appendix A: Software Flowcharts


START
SELECT DESIRED MOISTURE RATE
IS THE SOIL MOIST ENOUGH?
NO
YES
IS THERE ENOUGH WATER IN BOTTLE?
NO
THE RED LED LIGHTS AND DOES NOT START THE PUMP
YES
START IRRIGATION

Appendix B: Assembly codes

```
#include "mbed.h"

Serial MyPC(USBTX, USBRX);

Timer tim; //Timer
```

```
DigitalIn echo(PC_8); //Sensor Input

DigitalOut trigger (PC_9); // User Button is input

DigitalOut moist_ratio (LED1);

AnalogIn moist (PA_0);

DigitalOut pump (PC_0);

DigitalOut RedLed (PC_10);

DigitalIn button(BUTTON1);



void clrscr() // Clear the screen

{

   char clrscr[] = {0x1B, '[', '2', 'J',0};

   MyPC.printf(clrscr);

}



void homescr() // Home the cursor

{

   char homescr[] = {0x1B, '[', 'H', 0};

   MyPC.printf(homescr);

}



int main()

{

   float range;

   trigger = 0;
```

```
clrscr();

homescr();

MyPC.printf("\n\rPLANT");

int watering = 0;

pump = 0;

RedLed = 0;

int desired_moist = 65;

int mode = 1;



while(1) {



   wait_ms(100);



   trigger = 1;

   wait_us(10);

   trigger = 0;

   while(echo == 0);

   tim.start();

   tim.reset();

   while(echo==1);

   tim.stop();

   float time_zzz = tim.read_us();

   range = time_zzz/58.0f;


   double Soil_moist = moist;

   double moistt = 2.23*(100-((Soil_moist)*100));
```

```
if (moistt >= desired_moist) {
   watering = 0;
} else {
   watering = 1;
}

MyPC.printf("\n\rMOIST %0.3f", moistt);
MyPC.printf("\n\rRANGE %0.3f", range);
MyPC.printf("\n\rWATERING %u", watering);
MyPC.printf("\n\rDESIRED MOIST %u", desired_moist);
MyPC.printf("\n\rMODE %u", mode);



if(range > 9) {
   RedLed = 1;

} else {
   RedLed = 0;
}




while(button == 0) {
   if (desired_moist == 65) {
      mode = 2;
```

```
    }

    else if (desired_moist == 45) {
        mode = 1;
    }
    while (button == 0);



    break;
}




if (mode == 1) {
    moist_ratio = 0;
    desired_moist = 65;
}
if  (mode == 2) {
    moist_ratio = 1;
    desired_moist = 45;
}

if (watering == 1 and RedLed == 0) {
    pump = 1;
    wait_ms(3000);
    pump = 0;
    wait_ms(3000);
}
```

```
        clrscr();

        homescr();

        MyPC.printf("\n\rPLANT");


    }

}
```