Automated Plant Watering System

By Nisal Ovitigala & Danica Dong

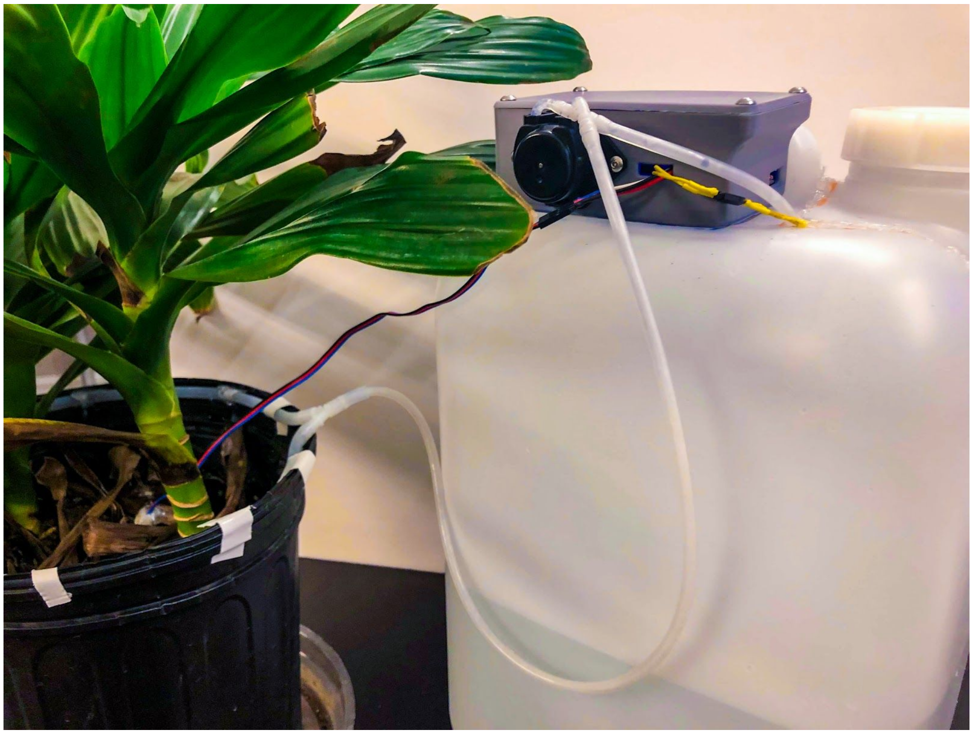
Overview:

The automated plant watering system uses a capacitive soil moisture sensor to detect when the plant needs to be watered. When the moisture is below the programmed threshold, the system uses the pump to add more water to the soil. The system also includes a float switch that identifies when water is low in the tank. Through the float switch, the user is able to receive email notifications on when the tank needs to be refilled.

Total Hours: ~108 hours

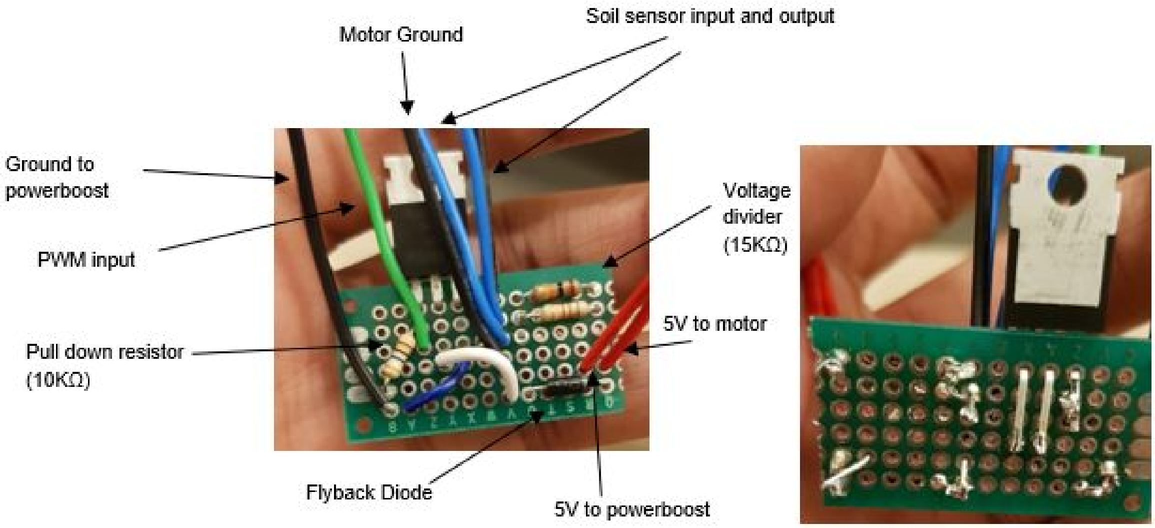
Tools Used:

Soldering iron (Bevel & conical tip)
Wire cutter
Wire stripper
3D printer
Drill (3/32 bit)
Hot glue gun
Tape
Tweezers

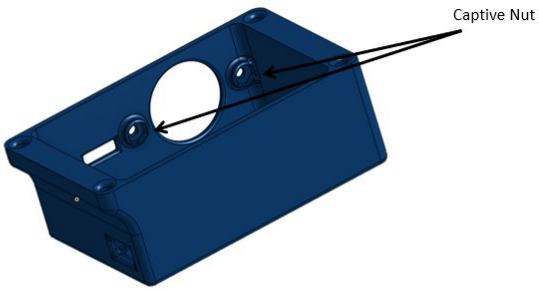


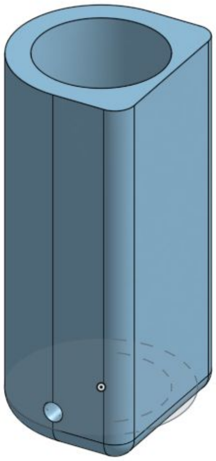
Assembly:

1) Connect the resistors, diode, and MOSFET to the ESP8266 using the protoboard as shown in the schematic.

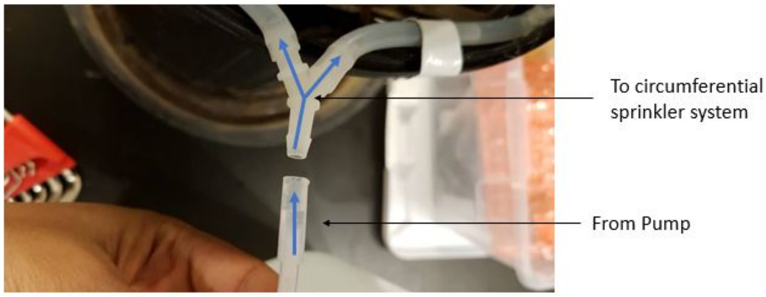


2) Print out the box, the lid, and the float switch holder.
3) Place the soldering iron with the conical tip into the heat insert and using tweezers, place the heated insert over the hole and let the weight of the soldering iron sink the insert into the plastic.
4) Place 6-32 nuts into the captive nut

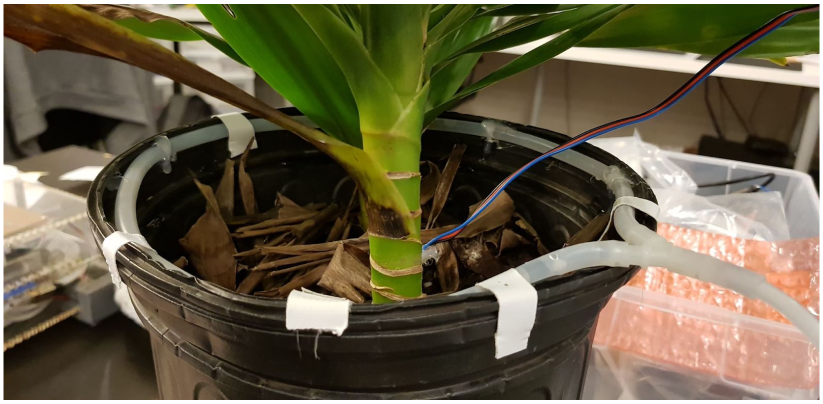




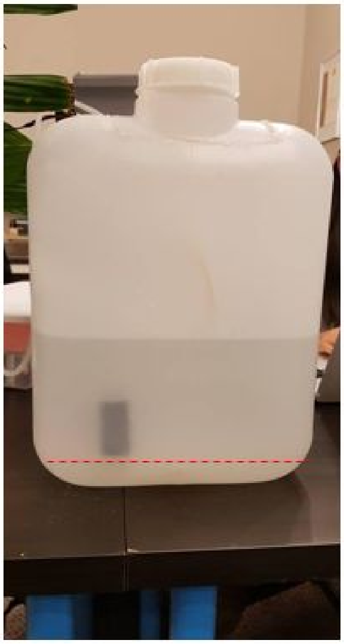
5) Slide the motor through the big circular cutout and screw it into the nut in the captive nut.
6) Slide the battery underneath the motor.
7) Cut a 3 cm piece of tubing and affix it to the end of the pump that will be releasing water (This tubing acts as the connector between the water supply and the Y junction)



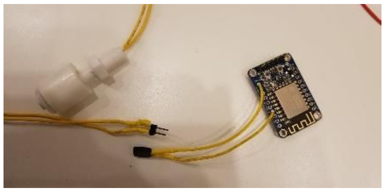
8) Measure out another piece of tubing whose length is the circumference of the pot.
9) Make holes in this tube (using a drill bit, nail or pin) which are 2 cm apart to let the water sprinkle onto the plant and attach it to the other two ends of the Y Junction.
10) Tape the tube to the circumference of the pot.
11) Utilizing the Y connector, attach the tubing all together.
12) Affix the float switch to the holder using hot glue. Make sure the hot glue does not interfere with the float switch functionality.



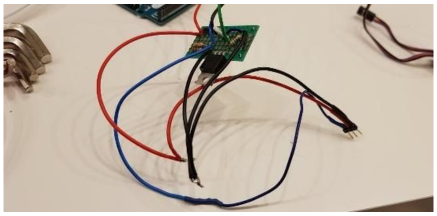
13) Cut an opening in the jerrycan large enough to attach the float switch holder to the bottom of the jerrycan.
14) Attach the float switch holder to the bottom side of the jerry can by the red dashed line using hot glue.

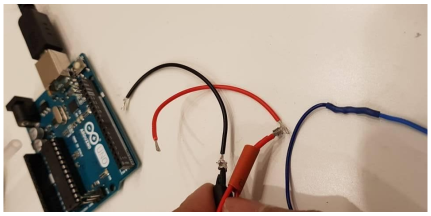


15) Insert the suction side of the motor pipe into the jerry can and hot glue to the bottom.
16) Seal the opening of the jerry can using hot glue.
17) Waterproof the soil sensor by adding a layer of hot glue over the exposed electronics.
18) Solder the soil sensor and float switch to a removable connector so they can be detached from the circuit enclosure for ease of transportation (Heat shrink tubing helps keep the wires isolated).

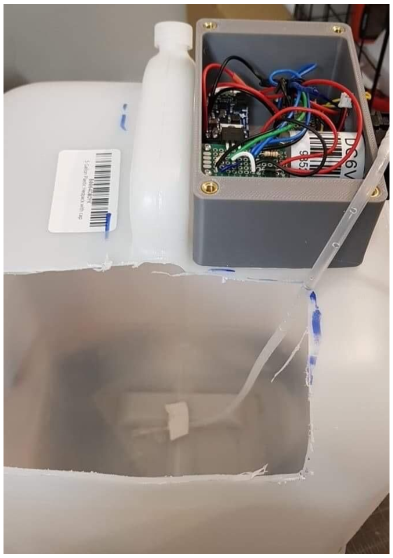


19) Solder the 5V lines and grounds from the soil sensor and on the protoboard together to a single 5V and ground wire.





20) Attach the single 5V and ground to the powerboost.
21) Attach the float switch to the micro controller and the soil sensor to one of the blue wires on the protoboard the protoboard.
22) Upload the program onto the ESP8266.
23) Drill the holes on the protoboard with the 3/32 drill bit to make it larger.
24) Fasten the protoboard, ESP8266, and powerboost to the box using the 2-56 screws
25) Connect the battery to the powerboost as shown in the schematic (below).
26) Ensure that all the wiring is within the box and fasten the printed lid to the box (Twist ties help with cable management).
27) Tape the box to the jerry can.
28) Insert the soil sensor into the plant's soil.
29) Using the serial monitor check what value from the soil sensor would be ideal for the motor to turn on and edit that in the code.



Program:

The program includes 3 tabs. The first tab "email_sender" includes the main code that checks the soil sensor and the float switch. Also change the code to include the correct wireless network name and password and the correct email of the receiver.

```
#pragma region Globals
const char* ssid = "insertWirelessNetworkName"; // WIFI network name
const char* password = "insertPassword";        // WIFI network password
uint8_t connection_state = 0;                  // Connected to WIFI or not
uint16_t reconnect_interval = 10000;         //If not connected wait time to
      try again
#pragma endregion Globals
```

This is the function that sets up the motor to run

```
void motorON (int pinNumber, int timeInterval) //function for turning the
      motor on
{
      analogWrite (pinNumber, 1023);
      delay (timeInterval);
      analogWrite (pinNumber, 0);
      delay (100);
}
```

The following code is where you input your email address to send the low water warning email

```
  if (sentEmail == 0){ //if the email is not sent
            Gsender *gsender = Gsender::Instance();    // Getting pointer to
class instance
            String subject = "REFILL WATER TANK!";
            if(gsender->Subject(subject)->Send("EXAMPLE@EMAIL.COM", "REFILL"))
{
            Serial.println("Message send.");
            sentEmail = 1;
      }
```

This following piece of code is what reads the soils moisture level. The value in the conditional is what dictates at what voltage value the motor turns on and has to be tweaked for each and every different plant/

```
void loop() {

   unsigned long currentTime = millis();
   int soilMoisture = analogRead (A0);
   Serial.println(soilMoisture);
   delay (10);

   if (soilMoisture > 900){
        motorON (5,10000);
```

The second tab called “Gsender” includes code necessary to send the email.

In the last tab named Gsender.h, convert the email and the password of the account used to send the emails to be Base64 encoded.

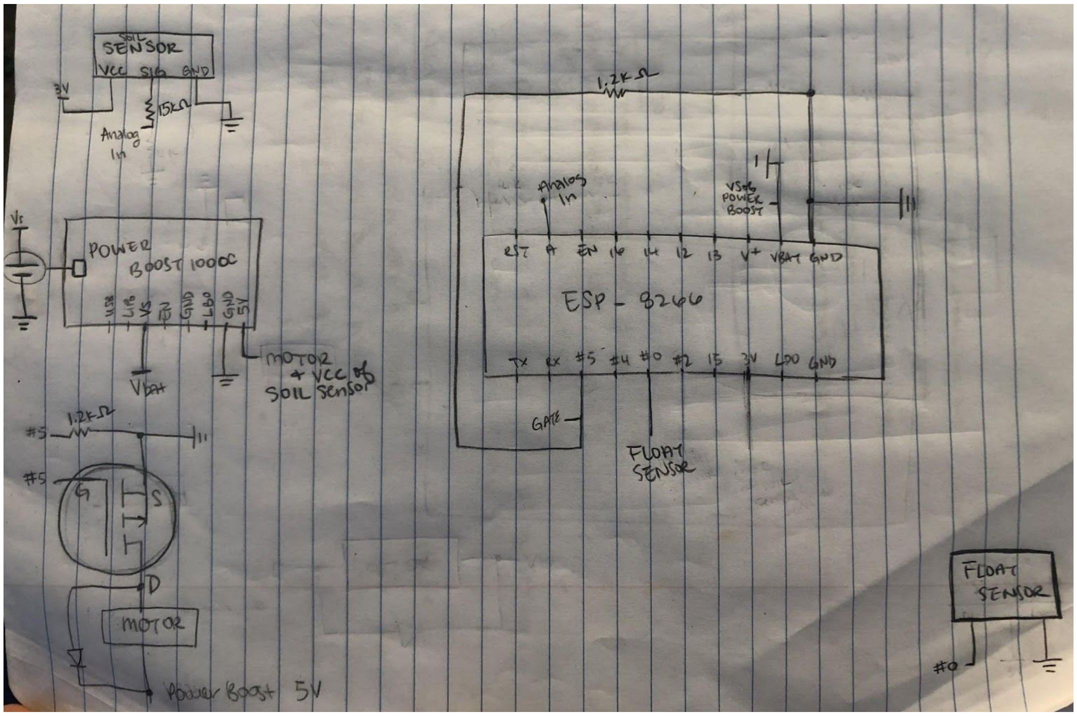
Bill of Materials:

Item	Quantity	Price
Soil Moisture Sensor	1	8.39
Liquid Pump – 5V to 6V DC Power	1	24.95
Jerry Can	1	16.39
Adafruit Huzzah ESP8266	1	14.43
Lipo Battery	1	16.88
power boost 1000	1	19.68
MOSFET	1	1.00
Resistors (15KΩ and 10KΩ)	2	0.30
Tubing (5mm ID)	1	5.67
Float switch	1	1.83
Y connector (3/16”)	1	1.20
Diode	1	0.35
Protoboard	1	1.06
Wires – 3 ft	1	0.394
2-56 1/2” Screw	8	0.07
2-56 3/8” Screw	2	0.02
8-32 ¾” Screw	2	0.02
8-32 ¼” Screw	2	0.02
6-32 3/8” Screw	2	0.05
6-32 Nut	2	1.40
2-56 Heat Insert	10	0.45
8-32 Heat Insert	4	0.24
Total		122.24

- Moisture sensor - https://www.dfrobot.com/product-1385.html
- Float switch - https://www.amazon.com/dp/B07211T6Y6/ref=sspa_dk_detail_1?psc=1&pd_rd_i=B07211T6Y6&pd_rd_w=Wgmro&pf_rd_p=21517efd-b385-405b-a405-9a37af61b5b4&pd_rd_wg=49ind&pf_rd_r=313RVSG4WS77GFH77CS4&pd_rd_r=b7495c0a-1453-11e9-a820-63076806f502
- Motor - https://www.adafruit.com/product/3910

- Microcontroller - https://www.amazon.com/802-11-Development-HUZZAH-ESP8266-Breakout/dp/B0131V0WYI/ref=sr_1_1_sspa?ie=UTF8&qid=1546977037&sr=8-1-spons&keywords=Adafruit+HUZZAH+ESP8266&psc=1
- MOSFET - https://www.amazon.com/WeiMeet-RFP30N06LE-N-Channel-Mosfet-Arduino/dp/B07CTF1JVD/ref=sr_1_4?ie=UTF8&qid=1546978994&sr=8-4&keywords=mosfet+arduino
- LiPo Battery - https://www.amazon.com/Adafruit-328-Battery-Lithium-Polymer/dp/B01NAX9XYG/ref=pd_bxgy_2?_encoding=UTF8&pd_rd_i=B01NAX9XYG&pd_rd_r=3a0bef30-1376-11e9-8ec8-8d4482e8e4d7&pd_rd_w=LT6NU&pd_rd_wg=hbdum&pf_rd_p=6725dbd6-9917-451d-beba-16af7874e407&pf_rd_r=XR9G051CSYSX9M8W5Q50&psc=1&refRID=XR9G051CSYSX9M8W5Q50
- PowerBoost 1000C - https://www.amazon.com/Adafruit-PowerBoost-1000-Charger-Rechargeable/dp/B01BMRBTH2/ref=sr_1_1?ie=UTF8&qid=1547059292&sr=8-1&keywords=power+boost+1000c

Schematic:



Improvements:

- After assembling everything, I was trying to understand why the voltage divider wasn't working as expected. Turns out the first resistor is built into the soils sensor and we only needed to add a 15KΩ resistor to complete the divider circuit. This is specific for this soil sensor; other resistors (or no divider circuit) would work better for different soil sensors. The reason we're putting a divider circuit in is because the microcontrollers analog input only accepts a max of 1V and the sensor goes up to 2.5V.
- A more powerful motor would have worked better. Right now the water drips out of the tubing with some strong streams.
- A bigger battery or having a external power source would have been better since the whole system drains the battery relatively fast.