CS 362: Milestone 7
Mustafa Habeeb - mhabeeb2 - mhabeeb2@uic.edu
Daniel Aguilar - daguil9 - daguil9@uic.edu

Garduino - The Gardening Assistant

Abstract (100 Words or Less):

The Garduino is a system that enables anyone to succeed in growing and caring for their plant by allowing sensors such as the UV light sensor, soil moisture sensor, SD card reader, temperature/humidity sensor, and a water pump to read in data and take care of your plant. The user will have access to the information taken from the sensors through the SD card by plugging it into a computer and looking at the graphs via Excel. The Arduino will also send current plant information to a mobile device via Bluetooth when the user requests for it.

Detailed Project Ideas:

Requirements:

1. **Description:** The Garduino will collect data such as the soil moisture, amount of sunlight the plant is exposed to, humidity of the air surrounding the plant, temperature, and water the plant as necessary. Collecting this data and exporting into an Excel spreadsheet will allow the user to know how the plant is progressing through time and at what stages the plant is facing any kind of difficulties. With the Excel spreadsheet, the user can highlight the tables that will be created from the text file and create a line-of-best-fit. This information will show the steadiness of the amount of light being obtained, plants being watered based on soil humidity, and humidity surrounding the area of the plant, and temperature, ensuring the best quality of care for the plant in this all-in-one system.
2. **Project Design:** The project will use two arduinos where they will communicate via bluetooth and also send information to SD cards. One arduino will be connected to all sensors (UV light sensor, soil moisture sensor, water aqua pump, temperature/moisture sensor, and ultrasonic sensor) that will measure the amount of sunlight coming in on that position, measure the moisture of the soil, water the plant accordingly based on the soil's moisture, read in the moisture levels of the air and the temperature, and let the user know when to refill the container with more water. This device will then send information to the other arduino via bluetooth, which is connected to the SD card Read/Write sensor, and will write the data taken in from the first arduino to the SD card and also send this information to a mobile device via bluetooth.

   (Adjustment due to the COVID-19 pandemic) Due to the pandemic, some adjustments of how we will be approaching the project will affect the entire thing. My partner and I each have a couple of the sensors and will therefore work with the sensors

that we each have. The main idea is still the same, but we tried to do as much as possible as we could with the sensors we have. With that being said, one arduino will have the soil moisture sensor, water pump, bluetooth module, and ultrasonic sensor. The other arduino will have the SD card read/write sensor, UV sensor, bluetooth module, and Temperature/humidity sensor. Essentially each arduino will be communicating with an application called *BlueDuino Published by HassaanAkbar.obj*. This application can be found on the Microsoft store and it assists in bluetooth communication with other devices. One arduino can send when requested by the user either the number of times the water pump was activated or to check if we need to refill the water container. This information assists in seeing if the temperature is affecting the soil and keeps the system consistent in assisting the plant. The other arduino will be collecting values and will be writing values into an SD card. We then can create graphs with a time being the X-axis and temperature/humidity or UV light being the Y-axis.

3. **Communication Between Arduinos:** Ideally, the communication between the two arduinos was going to communicate via bluetooth, where one would have all sensors connected to it and send this data to the second arduino that would only have the SD card reader and bluetooth module attached to it. However, we adjusted so that both arduinos would communicate with an external device instead of each other. Based on user input, certain information will show up on your screen when the bluetooth module is connected to the BlueDuino. The first arduino when the user inputs 1, will output if the water container needs to be refilled. When the user inputs the value 2, it will let the user know how many times water was pumped into the soil, this can let the user know if certain locations are not great for the plant based on the humidity in the soil. The second arduino when the user inputs 1, will know about the amount of sunlight the plant is receiving along with the temperature around the plant. This information could be very useful once you look at the graphs and see what the trends look like for the plant and its growth.
4. **Input/Output:** Our inputs for this project are the UV light sensor, soil moisture sensor, ultrasonic sensor, and moisture/temperature sensor. The UV sensor, and the humidity/temperature sensors are our inputs for the values that they read will be sent to our output, the SD Card Reader. That way we can plot the data and analyze the growth of the plant. The Soil sensor acts as an input by letting us know when the moisture in the soil is dry (the ranges are included as comments in our code in Garduino_MH.ino file). Once we know the soil is dry, our output, the water pump will pump water in 5 second intervals. Lastly, the Ultrasonic sensor also acts as an input, for when the user requests information the ultrasonic sensor measures the distance of the water container and calculates the percentage of water that is left which is outputted on an external device that uses the BlueDuino app.

   The first arduino contains the following…

   a. Inputs: Ultrasonic sensor, and soil sensor.

b. Outputs: Water pump, and bluetooth module

The second arduino contains the following…

c. Inputs: UV sensor, and Humidity/temperature sensor.

d. Outputs: SD card reader, and bluetooth module

5. **Original Work:** The original piece of this project is that we are able to analyze and see what is working to keep the plant's health in good standing. We determine that by looking at the trends of each of the following charts that could be produced.
   a. UV Light and Time (line graph)
   b. Number of Water pumps vs Time (number of pumps per day) (bar graph)
   c. Soil Moisture vs Time (line graph)
   d. Humidity vs Time (line graph)

   UV Light vs Time could tell us if the plant is receiving consistent amounts of light throughout the day and what times is it receiving the most, helping the user find the perfect spot in their home. The number of water pumps vs time can tell us when the plant is receiving abnormal amounts of water per day. This information is useful because that means the soil is drying up quicker than usual. The goal is to keep the number of pumps consistent because this shows the plant is on steady growth. Soil moisture vs Time and Humidity vs Time act hand in hand with each other. If the soil moisture is dry when the humidity in the air is at a certain level, we can use this information to create comparisons to find the perfect setting for the plant. The information gathered from these sensors assists in finding the perfect setting and also self cares for the plant.
6. **How to build:** This project is being created with two different sets that can be put together. Refer to "required materials" for what you need. Next, You need to look at the fritzing diagrams we provide below and create the connections. Then with the code provided, upload it to your arduino. Here are some steps below.

   <u>Steps:</u>

   a. Gather Materials.
   b. Download the code that is provided via the github link below (or type out the code as it is also shown in the report).
   c. Begin the setup with the first Arduino (the one that belongs connected and closest to the plant) by looking at the Fritzing diagram below titled "Soil Sensor/ Water Pump/ Ultrasonic Sensor/ Bluetooth module Setup".
      i. One of the tricky parts of building the first arduino will be when setting up the Ultrasonic Sensor above the water container. The height we had measured it to was 14 cm in distance. If the distance for some reason needs to be changed go back to Garduino_MH.ino and look at line 91 where mapping the value and change it to whatever distance you have set it up to.

d. Next, set up the second Arduino with the help of the other fritzing diagram titled "UV/Humidity/Temp. Sensor/ SD Card Reader/Bluetooth module Setup"
    i. Be sure to have a microSD card that is no bigger than 8GB or else the SD Card Sensor will not function properly.
e. Load the code onto its respective Arduino, and begin running the code.

7. **How to be used:** First, we must check that an SD card inserted for the data to be written into. Ideally, this device could be left on any window sill and will record data all day. And whenever the client will like, they could stop the device, remove the card, and input the text file (.txt) into Excel. Next the user can simply highlight all the data and see the projections on how the plant is doing.
   Steps to insert text file to excel to plot points:
   click on the 'Data tab' -> select 'From Text' -> choose your file -> check off delimiter and hit 'next' -> than select 'comma' for the delimiter -> hit 'Finish' -> Afterwards, Highlight columns you would like to graph -> Within the 'Insert tab' -> and click on the charts.

<u>Required Supporting Materials:</u>

1. **Proposed Timeline:**
    a. **February 23rd - 29th:** Purchase parts and other materials we need for the overall design.
    b. **March 1st - 7th:** Start looking at bluetooth/Serial communication and developing code.
    c. **March 8th - 14th (COVID-19, had to leave UIC):**Figure out the kind of analysis that can be made to determine if the plant is growing well for the user. Communicated online and worked on visualizing certain graphs and charts.
    d. **March 15th - 21st:** Redesign what each member will do in the group in order to get as much done with the parts that we had.
    e. **March 22nd - 28th:** Calibrate sensors (UV/Soil Moisture/ water pump) and figure out parsing and sending data to the SD card for certain sensors (humidity/temp Sensor and UV light Sensor).
    f. **March/April 29th - 4th:** Continue developing code.
    g. **April 5th - 11th:** Figure out communication between sensors, submit abstract, create videos, and powerpoint.
    h. **April 12th - 18th:** Finalize project and submit by April 17, 2020.

2. **Materials:**
    a. UV sensor
    b. Soil Moisture sensor

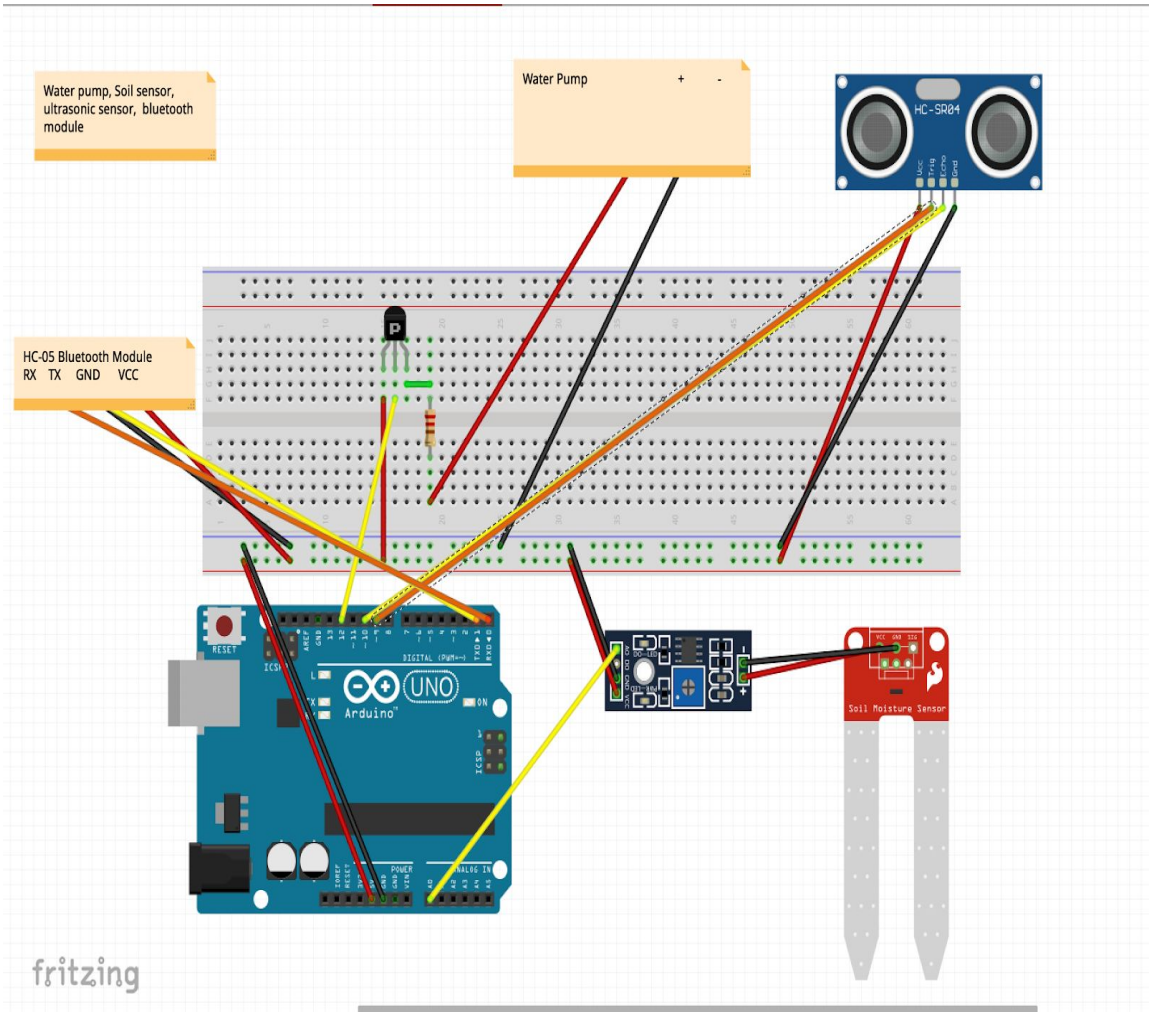
c. Water Pump
d. Bluetooth module
e. SD card reader
f. Pot for soil/plant
g. 2x Power supply 9v
h. 2x Battery snap on connectors
i. Male/female wires
j. A plant
k. Water container (Cup)
l. Electric tape
m. 3x Breadboards
n. Ruler
o. Cardboard (assist in creating setup for the ultrasonic sensor)

3. References:

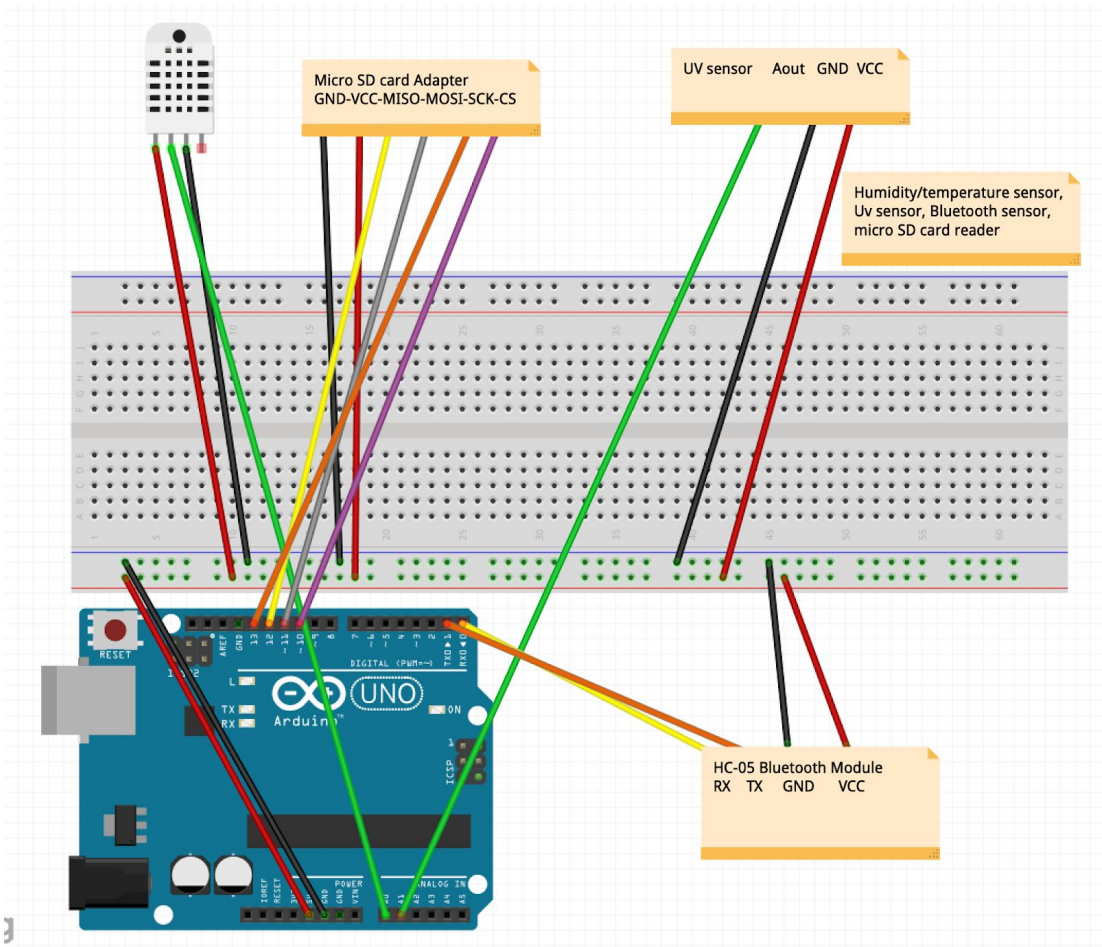
a. https://create.arduino.cc/projecthub/electropeak/sd-card-module-with-arduino-how-to-read-write-data-37f390
b. https://www.arduino.cc/en/Tutorial/ReadWrite
c. http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc-06-pair-bind-and-link/
d. http://www.martyncurrey.com/arduino-to-arduino-by-bluetooth/
e. http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc-06-pair-bind-and-link/
f. https://www.youtube.com/watch?v=E-1w7dL3Cps
g. https://www.brainy-bits.com/dht11-tutorial/
h. https://www.waveshare.com/wiki/UV_Sensor
i. https://www.youtube.com/watch?v=FkPbwRjNjCE
j. https://www.youtube.com/watch?v=0XvcTOI3-1Y -------> ultrasonic sensor
k. https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/ -------> humidty sensor
l. https://www.youtube.com/watch?v=O_Q1WKCtWiA ------> talks about graphing
m. http://www.circuitstoday.com/arduino-soil-moisture-sensor ------> soil moisture sensor
n. https://www.best-microcontroller-projects.com/arduino-map.html
o. https://www.instructables.com/id/Soil-Moisture-Measurement-With-Arduino/
p. https://www.youtube.com/watch?v=BXXAcFOTnBo
q. https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/ -----> bluetooth module

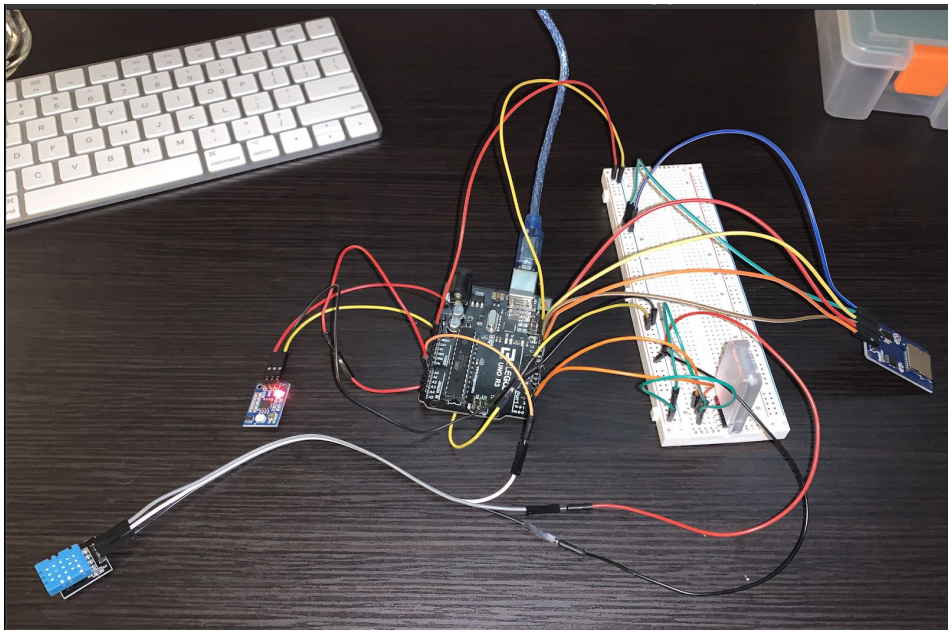
r. https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/
s. https://www.arduino.cc/reference/en/language/functions/math/map/
t. https://www.electronics-lab.com/get-sensor-data-arduino-smartphone-via-bluetooth/
u. https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/ -------> ultrasonic sensor
v. Get BlueDuino -----> BlueDuino application

4. **Hardware Diagram:**
Soil Sensor/ Water Pump/ Ultrasonic Sensor/ Bluetooth module Setup



UV/Humidity/Temp. Sensor/ SD Card Reader/Bluetooth module Setup





5. Code:

Github link to see code:
https://github.com/MustafaHabeeb20/CS362_FinalProject_Garduino

Soil Sensor/ Water Pump/ Ultrasonic Sensor/ Bluetooth module CODE

```
/*
 *
 *  Mustafa Habeeb and Daniel Aguilar - Group 28
 *  Title: Garduino
 *  Water Pump
 *  Soil Sensor
 *  Bluetooth module using serial
 *  Ultasonic Sensor to show if the bottle is filled
 *
 *
*/

//SETUP SENSORS
int soilSensor = A0;          //analog pin
int waterPump = 12;           //digital pin
int trigPin = 9;              //digital pin for UltraSonic Sensor - emits UltraSound
int echoPin = 10;             //digital pin for UltraSonic Sensor - recieves UltraSound

//OTHER VARIABLES
int state = 0;                //initialize at 0, assists in taking input and reading via blutooth
int soilSensorVal = 0;        //initialize at 0, but will be read from analog
int readCounter = 0;          //initialize at 0, used to count the number of reads when the soil is moist
int waterPumpCounter = 0;     //initialize at 0, used to count the number of times the water has been pumped
int USSensorVal = 0;          //initialize at 0, but will be read from after we caclulated distance
long duration = 0;            //initialize at 0, used to record soundwave time traveled in microseconds
int distance = 0;             //initialize at 0, calculates the distance using duration, velocity = 340 m/s,
                              //  and divinding by 2 becuase of the sound travels forward and backward

void setup(){

  Serial.begin(9600);         //Starts the serial communication
  pinMode(soilSensor, INPUT); //will take in input of how the soil is (either wet or dry)
  pinMode(waterPump, OUTPUT); //if the soil is dry, the waterPump will send water.
                              //  When the soild is moist (wet), the water pump will do nothing.
```

```
void setup(){

  Serial.begin(9600);         //Starts the serial communication
  pinMode(soilSensor, INPUT); //will take in input of how the soil is (either wet or dry)
  pinMode(waterPump, OUTPUT); //if the soil is dry, the waterPump will send water.
                              //  When the soild is moist (wet), the water pump will do nothing.
                              //UltraSonic Sensor...
  pinMode(trigPin, OUTPUT);   //Sets the trigPin as an Output
  pinMode(echoPin, INPUT);    //Sets the echoPin as an Input
  delay(500);
}//End of setup

void loop(){

  if(Serial.available() > 0){                     //Checks whether data is comming from the serial port
    state = Serial.read();                        //Reads the data from the serial port
  }

  soilSensorVal = analogRead(soilSensor);
  soilSensorVal = map(soilSensorVal,1023,0,0,100); //The soil Sensor reads values from 0 to 1023, cetain ranges mean different things...
                                                   //source: https://www.instructables.com/id/Soil-Moisture-Measurement-With-Arduino/
                                                   //    Ranges: around 1000 - 1023 -> not in soil.
                                                   //            around 600 - 1000  -> in dry soil.
                                                   //            around 370 - 600   -> in humid soil.
                                                   //            around 0 - 370     -> in water.
                                                   //With with this knowlege we were able to map the values to show percentages,
                                                   //  and after running a few different trials, 55% of more, shows the soild will
                                                   //  be humid, otherwise, the soil is considered dry and will need to be watered.



  Serial.print(readCounter);                      //Output should look like ...
  Serial.print(". ");                             // "0. Moisture : (some percentage value)%
  Serial.print("Mositure: ");
```

```
  Serial.print(readCounter);                    //Output should look like ...
  Serial.print(". ");                           // "0. Moisture : (some percentage value)%
  Serial.print("Mositure: ");
  Serial.print(soilSensorVal);
  Serial.println("%");

  int dryVal = 55;                              //We know 55% is the cutoff between humid and dry soil,
                                                // for thsi reason we will set the dry value at.

  if(soilSensorVal <= dryVal){
   Serial.println("watering");
   digitalWrite(waterPump, HIGH);               //Pump water
   delay(3000);                                 //3 second delay after every time it needs to be watered till the soild sensor reads above 55%
   digitalWrite(waterPump, LOW);                //Stop pumping water
   delay(5000);                                 //5 second delay after watering.
   readCounter = 0;                             //set readCounter to 0 to see where it breaks in Serial monitor.
   waterPumpCounter++;                          //Count every time the water was pumped
  }//End of if-statement
  readCounter++;                                //Add to the counter as long as the soilSensorVal reads higher than the dryVal

                                                //source: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/
  digitalWrite(trigPin, LOW);                   //Clears the trigPin
  delayMicroseconds(2);                         //Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);            //Reads the echoPin, returns the sound wave travel time in microseconds
  distance= duration*0.034/2.0;                 //Calculating the distance
  Serial.print("   Distance: ");                //Prints the distance on the Serial Monitor
  Serial.println(distance);

  USSensorVal = map(distance,14,2,0,100);       //Map distance from 2cm to 14cm to a range from 0 to 100 representing percentages
                                                // where 14cm maps 0% where the cup is empty and 2cm maps to 100% cause the cup is filled
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);              //Reads the echoPin, returns the sound wave travel time in microseconds
  distance= duration*0.034/2.0;                   //Calculating the distance
  Serial.print("   Distance: ");                  //Prints the distance on the Serial Monitor
  Serial.println(distance);

  USSensorVal = map(distance,14,2,0,100);         //Map distance from 2cm to 14cm to a range from 0 to 100 representing percentages
                                                  // where 14cm maps 0% where the cup is empty and 2cm maps to 100% cause the cup is filled


  //STATES used to show blueTooth communication Inputs are 1 and 2
  if(state == '1'){                               //If input for State is equal to 1
    Serial.print("water level is at ");           //Give percentage of water-level from the cup the water pump is reciving from
    Serial.print(USSensorVal);
    Serial.println("%");

    if(USSensorVal <= 40){                        //If the USSensorVal is less than equal to 40% than the USER will know to
                                                  //  fill up the cup so they never run out of water
      Serial.println("Fill up water cup");
    }
    state = 0;
  }//End of if-statement

  if(state == '2'){                               //If input for State is equal to 2
    Serial.print("Number of pumps: ");            //Give number of pumps
    Serial.println(waterPumpCounter);
    state = 0;
  }//End of if-statement


  delay(5000);                                    //5-second delay
}//End of loop
```

UV/Humidity/Temp. Sensor/ SD Card Reader/Bluetooth module CODE

```
#include <Time.h>
#include <TimeLib.h>
#include <dht.h>
#define dht_apin A0 //Analog pin sensor that is connected to the
                        //temp/humidity sensor
#include <SPI.h>
#include <SD.h>

dht DHT;
File myFile;

//time_t t = now();
int hr;
int mint;
int state = 0; //Bluetooth
int uv_ain=A1;
int ad_value;

void setup() {
  pinMode(uv_ain, INPUT);
  Serial.begin(9600);
```

```
  delay(500); //A small delay to let the temp/humidity system to boot
  //Serial.println("DHT11 Humidity & Temperature Sensor & UV Sensor");
  delay(1000);
  while (!Serial) {
    ;//Wait for the serial port to connect. Needed for native USB port
     //only (For the SD card reader)
  }
  Serial.print("Initializing SD card...");

  if (!SD.begin(10)) {
    Serial.println("initialization failed!");
    while (1);
  }

  Serial.println("initialization done.");

  // open the file. Only one file can be open at a time,
  // so you have to close this one before opening another.
  myFile = SD.open("garden.txt", FILE_WRITE);

  delay(2000);
}

void loop() {

    if(Serial.available() > 0){
      state = Serial.read();
    }
    DHT.read11(dht_apin);
    ad_value=analogRead(uv_ain);


    // if the file opened okay, write to it:
    if (SD.exists("garden.txt")) {
      //Open the file
      myFile = SD.open("garden.txt", FILE_WRITE);

      hr = hour();
      mint = minute();


      myFile.print(hr);
      myFile.print(":");
```

```
      myFile.print(mint);
      myFile.print(",");
      myFile.print(DHT.humidity);
      myFile.print(",");
      myFile.print(DHT.temperature*1.8+32);
      myFile.print(",");
      myFile.print(ad_value);
      myFile.println("");

      //Close the file:
      myFile.close();
      Serial.println("done.");

      Serial.print("Current humidity = ");
      Serial.print(DHT.humidity);
      Serial.print("%  ");
      Serial.print("Temperature = ");
      Serial.print(DHT.temperature*1.8+32);
      Serial.println("F  ");
      Serial.print("UV Light = ");
      Serial.println(ad_value);
    }

    if(state == '1'){
      Serial.print("Current humidity = ");
      Serial.print(DHT.humidity);
      Serial.print("%  ");
      Serial.print("Temperature = ");
      Serial.print(DHT.temperature*1.8+32);
      Serial.println("F  ");
      Serial.print("UV Light = ");
      Serial.println(ad_value);
      state = 0;
    }

    delay(1000);//Wait 20 seconds before accessing the sensor again.

}
```