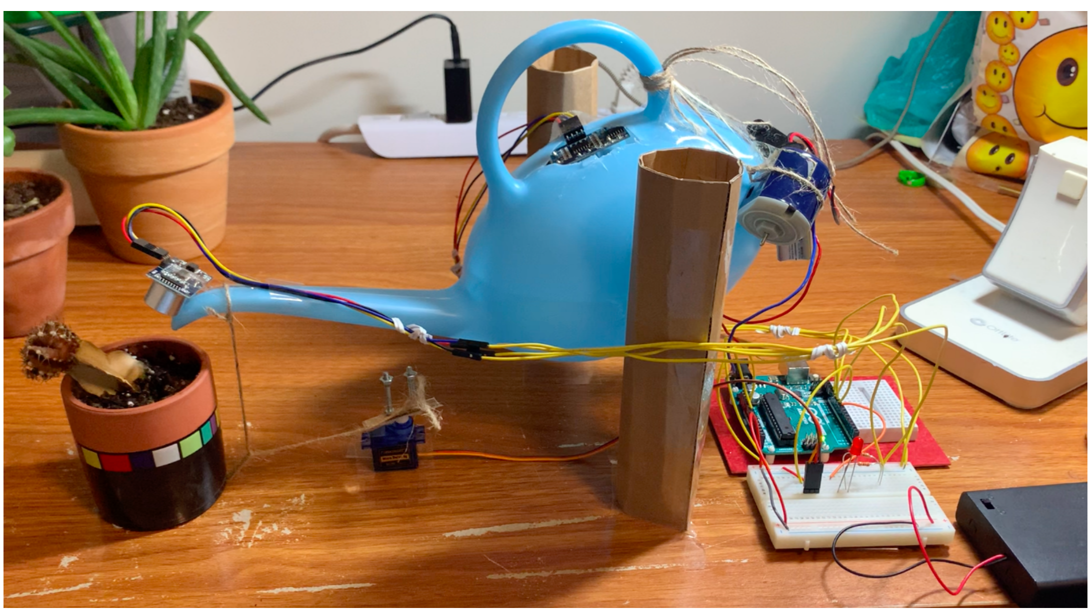
MAE 3780 Individual Project Final Report

THE SAP (SAVE ALL PLANTS) WATERING DEVICE

Charlotte Lee cyl53

1. OVERVIEW

This project is an automated plant-watering device, called the SAP (Save All Plants). It features two inputs in the form of two ultrasonic distance sensors. It has two outputs - a 9g micro servo motor and a red LED. There is a watering pot that is suspended upright in the air by two columns made from 5x7 in slabs of cardboard, and fastened by machine screws and hex nuts. One sensor is attached to the end of the watering pot. It detects the distance the spout of the pot is from the plant. The second sensor sits on the opening of the watering pot, looking into the pot. It measures the water level. The motor sits below the pot. As it rotates counterclockwise, it pulls a string. The string is attached to the spout of the watering pot. It lowers the spout, allowing water to flow out of the pot into the plant. The sensor attached to the spout detects that the distance between the spout and the plant is decreasing, and once the distance reaches less than 5 cm, the motor begins to turn clockwise until it returns to its initial position. The pot is once again upright. The sensor that sits in the watering pot will signal the red LED to turn on when it senses that the water level is more than 5 cm away. This distance of 5 cm between the water level sensor and the water level equates to the amount of water that the watering pot needs in order for the system to work. Lastly, there are weights attached to the back of the watering pot, that counteract the change in center of gravity that the rotation of the pot and the movement of the water within it cause to the structure. These weights allow the pot to return to its upright position when the string is loosened.

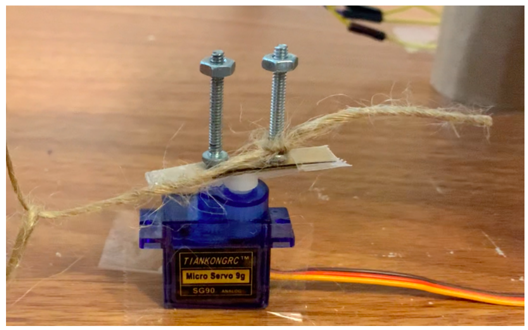


2. DESIGN CONSIDERATIONS

I would tell someone wanting to replicate this project to consider using a more powerful motor. The 9g micro servo motor works fine, so long as the watering pot that is chosen is not heavy, and the pot is not filled to the top with water.

a. What I would do differently

I would not spend money on Krazy Glue, cardboard, screws, and hex nuts in order to make the attachment to the motor (pictured below). Instead, I would either 3D print, or laser cut a custom motor attachment using CAD. I do not think the time, effort, and money put into creating this attachment was more efficient than 3D printing or laser cutting.



b. How I would improve the project

If I had more time, I would have improved the method of mounting the watering pot to the two cardboard legs. While they held up fine for the amount of water that I put in the pot, I am unsure if the screws/cardboard would have held the pot up if I had filled the pot to the top. To make it more stable, I would have used something more precise than scissors to make the holes in the pot and cardboard. If I had a larger budget, I would have used a different material than cardboard to make the legs. I would have liked to use sturdy plastic, which would have deformed less than cardboard when subject to loading. I also would have invested in small washers that could be taped to the watering pot to serve as counterweights. The DC motor and two 9V batteries that I used worked effectively, but were clunky and did not look aesthetically pleasing. Lastly, I would have liked to buy or manufacture a funnel that could allow me to refill the watering pot without having to remove the sensor that hung from its opening, while also ensuring that the sensor remained dry. Lastly, I would have invested in a large sheet of cardboard, wood, or plastic that I could have used as a stand/base for the entire device. This would have made the device portable, which would have allowed it to be used on plants in other locations in my apartment.

3. ASSEMBLY INSTRUCTIONS

Here are the steps for assembling the mechanical structure of the SAP:

1. Assemble circuit, referring to the circuit diagram in Appendix B.

2. Score two 5x7 in cardboard sheets by using scissors to indent (but not cut all the way through) cardboard along its longer side. Use the tip of the scissors to create hole in each sheet 5” from the bottom of the 7” side. See Figure 1 in Appendix C for drawing.
3. Use the tip of the scissors to cut a hole on either side of the watering pot. The holes should be in the center of the sides of the pot, at approximately ⅗ of the height of the pot. This step is dependent on the size of the pot used.
4. Check to see whether the ultrasonic distance sensor fits through the opening at the top of the watering pot. If it does not, use scissors to create a wider opening.
5. Take two 4x40 1” length machine screws, and place one in each hole of the watering pot, with the heads of the screws inside of the pot.
6. Take one cardboard sheet, and align the hole made in Step 2 with the end of the screw that is sticking out of one side of the watering pot. Push the cardboard through the end of the screw. Take a hex nut and secure it to the end of the screw.
7. Repeat this step for the other side of the watering pot.
8. Roll each cardboard sheet into a cylinder using the scores made in Step 2. The cylinders will have 5” circumferences. See Figure 2 in Appendix C for drawing. Use tape to join the two ends of the cylinder together. Tape the base of each cylinder to whatever surface you want the device to rest on. I used a desk.
9. Fill the pot up halfway with water.
10. Take one of the ultrasonic distance sensors, and use tape to secure it to the spout of the watering pot, with the sensor facing in the same direction that water would flow out of the watering pot. See Figure 3 in Appendix C for drawing.
11. Use tape to secure the other ultrasonic distance sensor to the opening at the top of the watering pot, with the sensor facing down into the pot.
12. Cut a piece of cardboard into a rectangle, approximately the same size as the two-armed horn that comes with the 9g micro servo. Tape the cardboard to the horn, making sure that the connection is secure.
13. Use Krazy Glue to glue the heads of two 4x40 1” length machine screws to the cardboard - one on either arm of the horn. See Figure 5 in Appendix C for drawing. Wait several hours for glue to dry.
14. Cut at least 18” of string. Tie one end of the string to one of the screws that you glued to the horn. Tie the other end of the string to the spout of the watering pot, right behind where the sensor was mounted. Tape the string to the spout. See Figure 3 in Appendix C for drawing.
15. Tape the 9g micro servo to the ground underneath the watering pot.
16. Cut a piece of string to approximately 4”. Wrap this string around the 18” string you made in Step 14. Tape the two ends of the 4” string to the surface you taped the cardboard columns to. The 18” string should now have an “L-shape.” See Figure 6 in Appendix C for drawing.
17. Use string to tie the DC motor and two 9V batteries to the back of the pot, around the handle. Tape down string/weights as needed.

18. Place a plant below the spout of the watering pot, such that the spout will hit the dirt in the plant pot if rotated far enough.

4. OPERATION INSTRUCTIONS

Here are the steps for operating the SAP:

1. Turn on the battery pack.
2. Watch as the SAP waters your plants!

If the red LED on the breadboard is lit up, it's time to refill the watering pot!

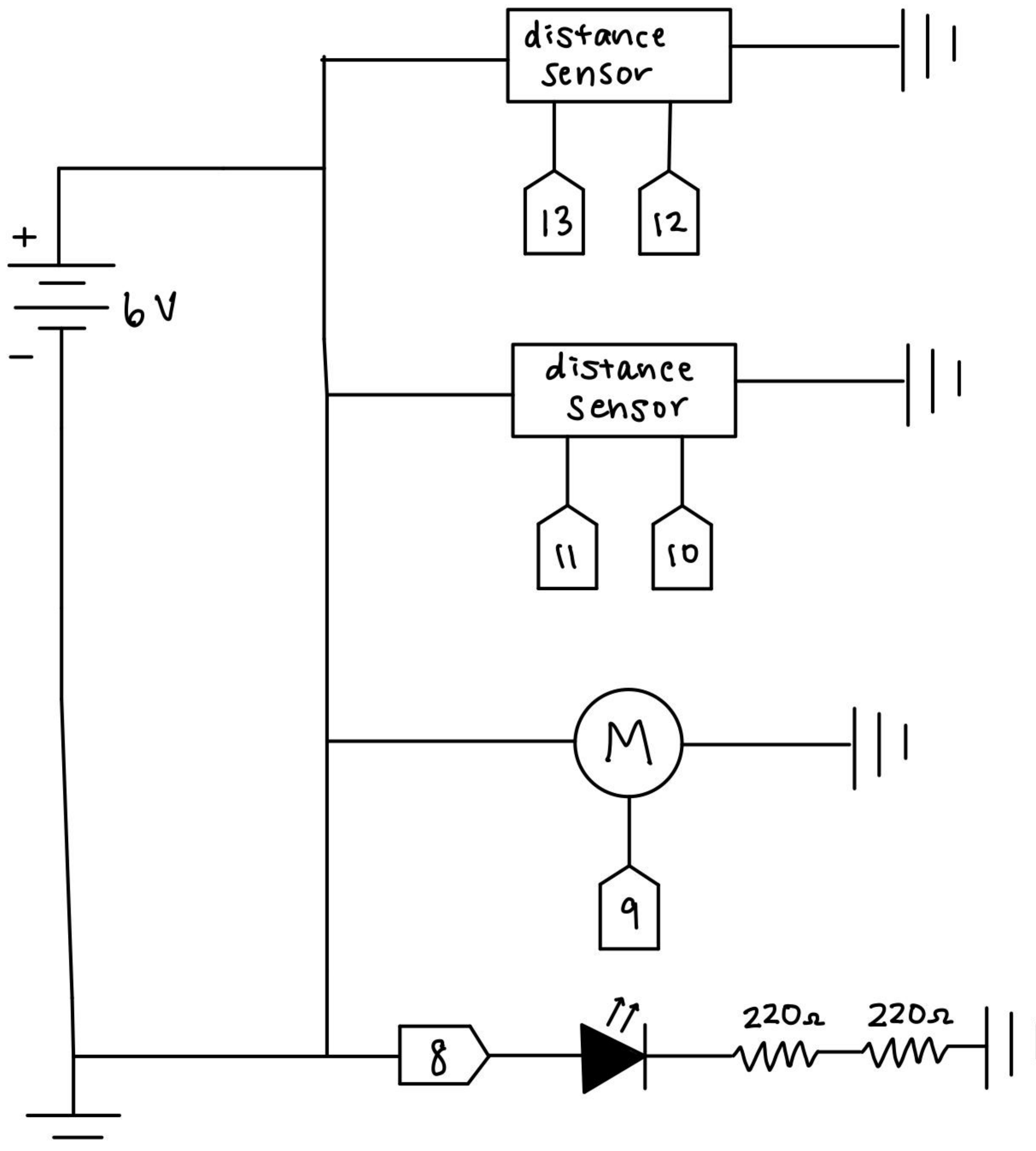
1. Remove the sensor that is taped to the top of the watering pot, and place it to the side.
2. Pour water into the pot.
3. Put the sensor back on top of the pot. Make sure it is secured by the tape.
4. If the LED turns off, you have filled the watering pot successfully.
5. If the LED remains on, repeat Steps 1-3.

APPENDIX A: BILL OF MATERIALS

ITEM	VENDOR	PART #	QUANTITY	UNIT PRICE	COST
Watering pot*	Amazon	B08TM6YM35	1	$3.99	$3.99
String*	Amazon	B08BBZLKMC	1	$3.69	$3.69
Krazy Glue*	Amazon	B000H5SFNW	1	$2.77	$2.77
Ultrasonic sensor*	Digi-key	1528-2711-ND	2	$3.95	$7.90
#4-40 machine screws 1” length*	Upson 264	N/A	4	$0.01	$0.04
#4-40 hex nuts*	Upson 264	N/A	4	$0.02	$0.08
5”x7” cardboard sheet*	Upson 264	N/A	3	$0.11	$0.33
1 ft wire*	Upson 264	N/A	8	$0.10	$0.80
Micro servo positional motor	Dfrobot	SER0006	1	$3.30	$3.30
Red LED	Jameco	697602	1	$0.05	$0.05
DC motor	Jameco	2209094	1	$1.25	$1.25
9V battery	McMaster-Carr	71455K68	2	$1.68	$3.36
9V battery cap (barrel)	Jameco	2207056	1	$1.39	$1.39
9V battery cap (leads)	Jameco	216452	1	$0.39	$0.39
220Ω resistor	Digi-Key	220QBK-ND	2	$0.01	$0.02
Arduino Uno R3	Digi-key	1050-1024-ND	1	$20.90	$20.90
Breadboard	Newark	79X3922	1	$2.71	$2.71
4-AA battery holder	Jameco	216187	1	$1.75	$1.75
Wire kit	Amazon	B07PQKNQ22	1	$2.17	$2.17
4-wire harness	N/A	1568-1931-ND	2	$1.35	$2.70
AA batteries	McMaster-Carr	71455K58	4	$0.40	$1.60

Total for parts not included in kit (denoted with *) = $19.60
Total for all parts (including ones in kit) = $61.19

APPENDIX B: CIRCUIT DIAGRAM



APPENDIX C: CAD FILES AND DRAWINGS

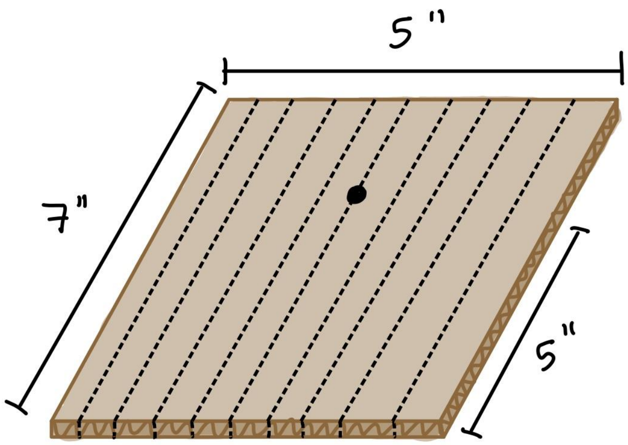


Figure 1. Scored 5” by 7” cardboard sheet with hole 5” from bottom of 7” side

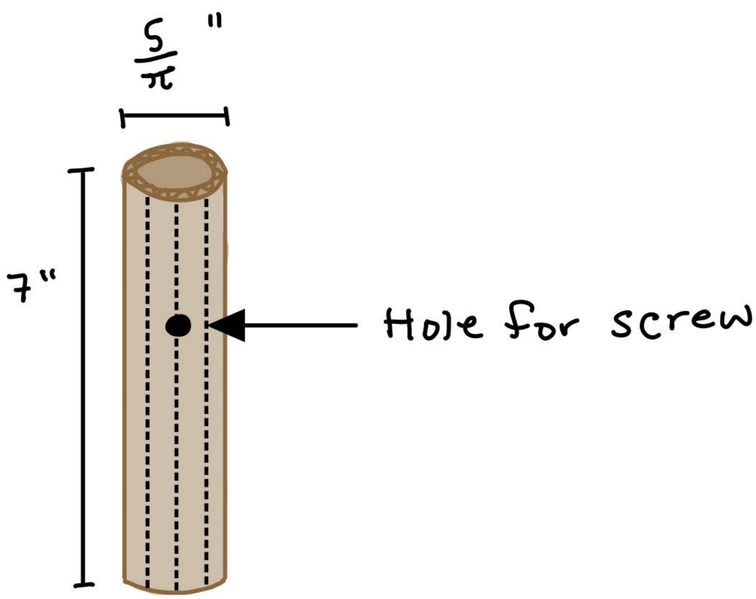


Figure 2. 5” x 7” cardboard sheet rolled into a cylinder, with height of 7” and diameter 5/π

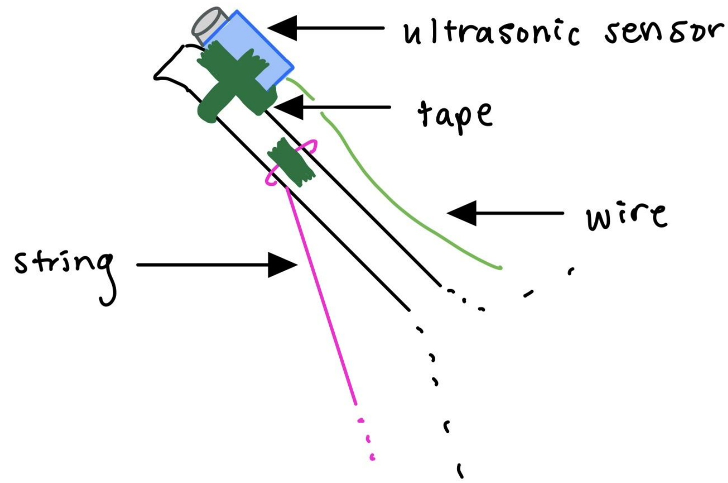


Figure 3. Spout of watering pot with ultrasonic distance sensor and string taped on

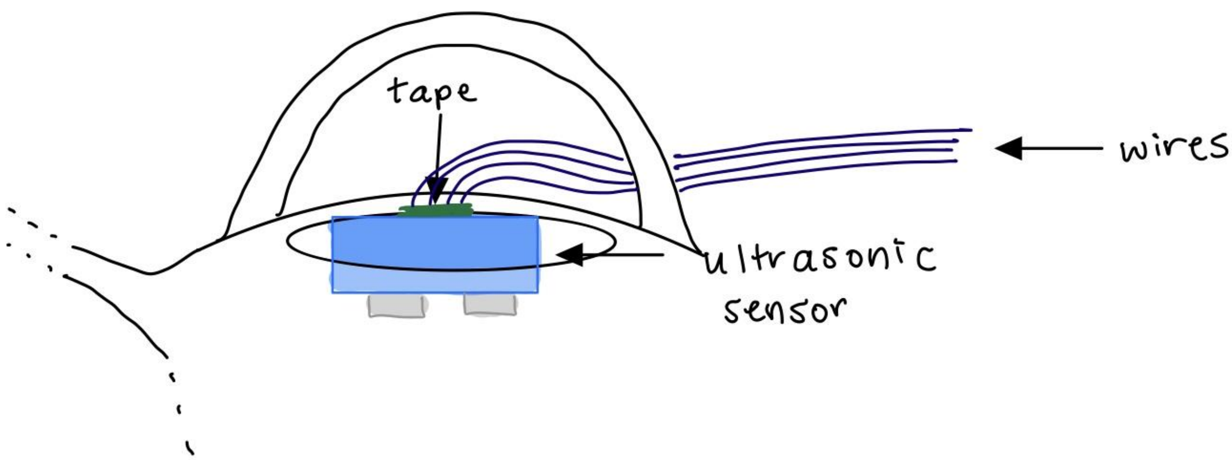


Figure 4. Top of watering pot with ultrasonic distance sensor taped to the top of it.

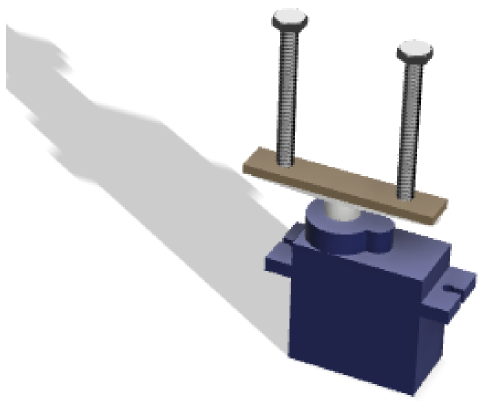


Figure 5. Motor and string setup with watering pot and cardboard cylinders

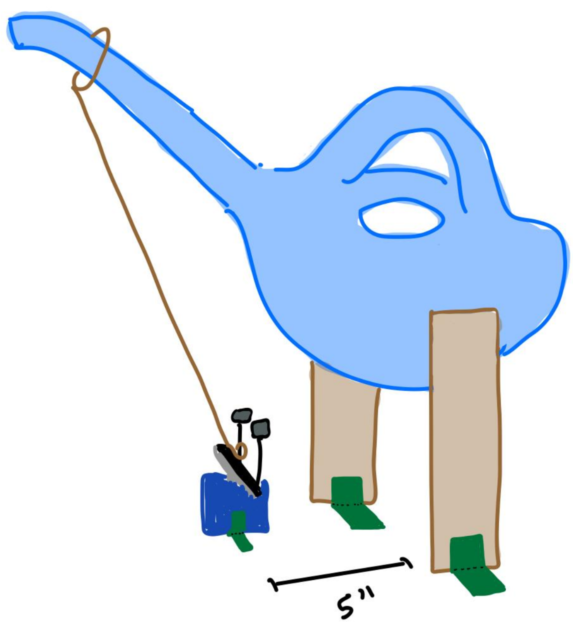


Figure 6. Motor with cardboard taped onto two-armed horn and machine screws glued onto cardboard

APPENDIX D: ARDUINO CODE

```
// This Arduino Code corresponds to the SAP Watering Pot design by Charlotte Lee (cyl53)
// The loop () portion of the code has 4 parts: sensors (code from Arduino IDE), motor, LED, delay
//    1. SENSORS: tells ultrasonic sensors to detect distances of water and plant
//    2. MOTOR: controls rotation of micro servo motor, based on input data from sensor "Sensing Plant
Pot"
//    3. LED: controls LED, which lights up based on input data from sensor "Sensing Water Level"
//    4. DELAY: delays rotation of motor and sensing of sensors until next time plant needs to be watered

#include <Servo.h>

Servo myservo;

const int water_sensor_pin_trig = 13;
const int water_sensor_pin_echo = 12;
const int plant_sensor_pin_trig = 11;
const int plant_sensor_pin_echo = 10;
const int servo_pin = 9;
const int LED_pin = 8;

bool within_5cm = false; // determines if spout is too close to sensor
int angle = 0;  // declare variable for servo angle [degrees]
int dt = 15;    // short delay time
int increment = 2; // amount that angle of servo motor changes by
int time_between_watering = 10000; // amount of time between when plant needs to be watered
long h = 5; // level that water in watering pot needs to be at or above [cm]


void setup()
{
  myservo.attach(servo_pin);
  pinMode(LED_pin, OUTPUT);
  pinMode(water_sensor_pin_trig, OUTPUT);
  pinMode(water_sensor_pin_echo, INPUT);
  pinMode(plant_sensor_pin_trig, OUTPUT);
  pinMode(plant_sensor_pin_echo, INPUT);
  Serial.begin(9600);
}

void loop()
{
  // 1. SENSORS

  // The following code chunk was copied and then modified from the "Sensors - Ping" example code in //
the Arduino IDE
  // ------------------------------------------------------------------------------------------------------------------------ //
```

```
// FOR WATER LEVEL SENSOR
long duration_water, cm_water; // initialize variables for measuring water level

// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

digitalWrite(water_sensor_pin_trig, LOW);
delayMicroseconds(2);
digitalWrite(water_sensor_pin_trig, HIGH);
delayMicroseconds(5);
digitalWrite(water_sensor_pin_trig, LOW);

duration_water = pulseIn(water_sensor_pin_echo, HIGH); //time (ms) from sending of ping to
reception of its echo off of object

cm_water = microsecondsToCentimeters(duration_water); // convert the time into a distance

delay(100);

Serial.print("Distance: ");
Serial.print(cm_water);
Serial.println(" cm");

// PLANT DISTANCE SENSOR
long duration_plant, cm_plant; // initialize variables for measuring plant distance

// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

digitalWrite(plant_sensor_pin_trig, LOW);
delayMicroseconds(2);
digitalWrite(plant_sensor_pin_trig, HIGH);
delayMicroseconds(5);
digitalWrite(plant_sensor_pin_trig, LOW);

duration_plant = pulseIn(plant_sensor_pin_echo, HIGH); //time (ms) from sending of ping to reception
of its echo off of object

cm_plant = microsecondsToCentimeters(duration_plant); // convert the time into a distance
delay(100);
// ------------------------------------------------------------------------------------------------------------------------- //



// 2. MOTOR
```

```
  bool watered = false; // declare variable that tells whether plant has been watered yet
  if (angle<180 && within_5cm == false) // rotate forward if spout is far enough from plant
  {
    angle = angle + increment; // increment angle
    myservo.write(angle); // move to new angle
    delay(dt);
    if (cm_plant < 5) // if spout is too close to plant, change boolean within_5cm to stop from rotating
more
    {
      within_5cm = true;
    }
  }
  else // rotate back to rest position (angle = 0)
  {
    if (angle >= 0) // if not at rest position yet
    {
        angle = angle - increment; // increment angle
        myservo.write(angle); // move to new angle
        delay(dt);
    }
    else // once pot has returned to rest position
    {
      within_5cm = false;
      watered = true;
    }
  }



  // 3. LED
  if (cm_water > h) // Turn on LED if the level of the water in the pot is low
  {
    digitalWrite(LED_pin, HIGH); // LED on
  }
  else
  {
    digitalWrite(LED_pin, LOW); // LED off
  }



  // 4. DELAY
  if (watered == true) // delays cycle from repeating by time "time_between_watering"
  {
    delay(time_between_watering);
  }
}
```

```
// The following code chunk was copied and then modified from the "Sensors - Ping" example code in the Arduino IDE
//
// ------------------------------------------------------------------------------------------------------------------------ //

long microsecondsToCentimeters(long microseconds) {
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the object we
  // take half of the distance travelled.
  return microseconds / 29 / 2;
}
//
// ------------------------------------------------------------------------------------------------------------------------ //
```