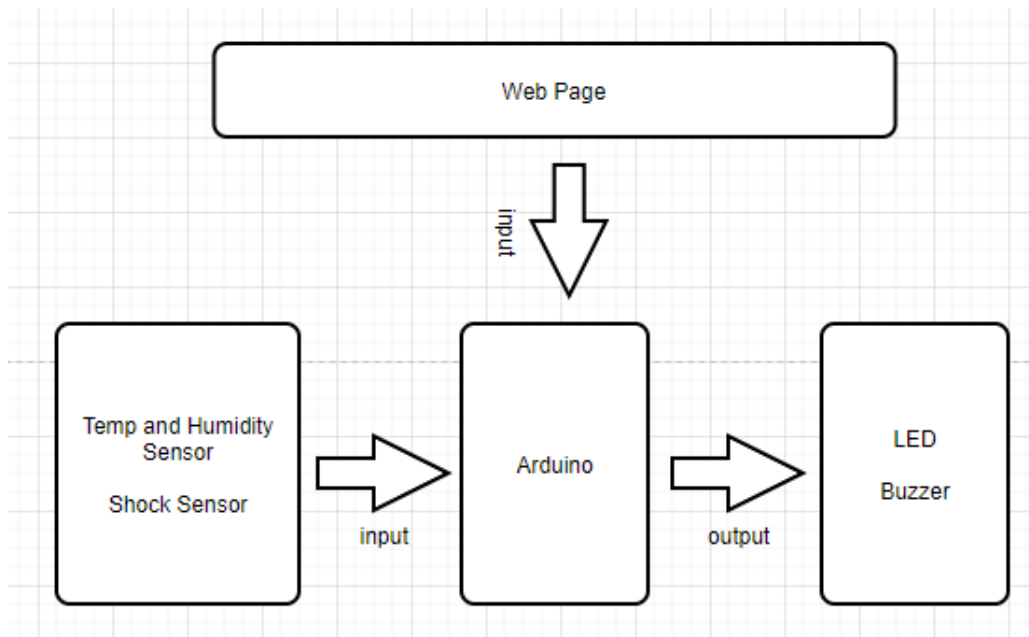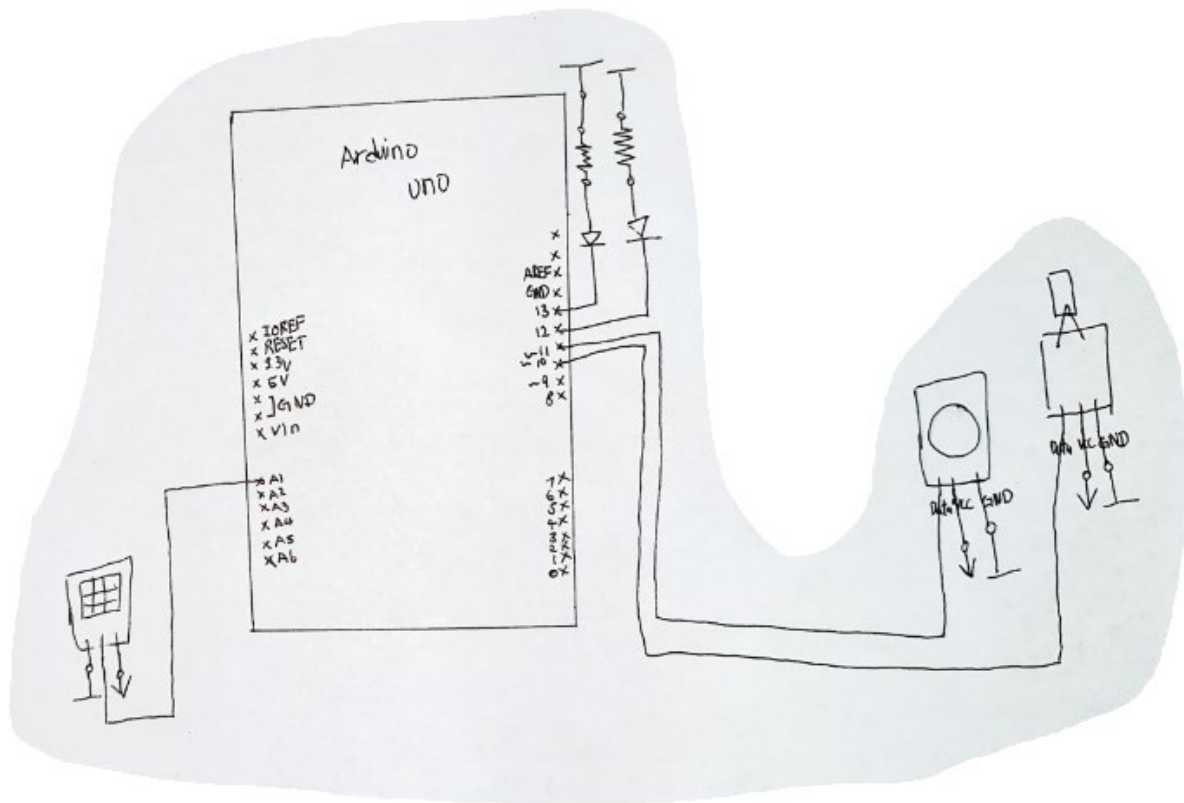# Link

https://youtu.be/FFSBhqY42GU

Summary

Topic Background

Nowadays IoT are stating integrate into people life. It made a lot changed into people lifestyle. It helps people to able to complete some work or task without too much Interventions of human. For example, light up the light automatically when the day become darker, when the day are hot the air conditioning will turn on itself, and auto call the fire department when the gas leakage was detected. It also allows the user to able to control the devices through the internet. Todays, Malaysia's plantation still facing some problems in protecting seeding stage plant, because it's important to keep the seed in a suitable temperature and humidity lever due to Malaysia's hot weather. At the same time, to preventing interference of the other animal such as bird, bog, cat and other. These issues might reduce the seed survival rate. To solve this problem a simple circuit has been developed.

Proposed System

This circuit was developed to help the plant owner to keep trace the temperature and humidity lever of the plant surrounding environment and decide whether the plant need to be watering. If the plant is needed to be watering the system will start to water itself. And the same time observe there are any animal are close to the plant, if the animal is too close to the plant buzzer will be turn on to frighten away the animal to prevent them destroy or eat the plant. The owner can also control the auto watering system and buzzer through the web side.
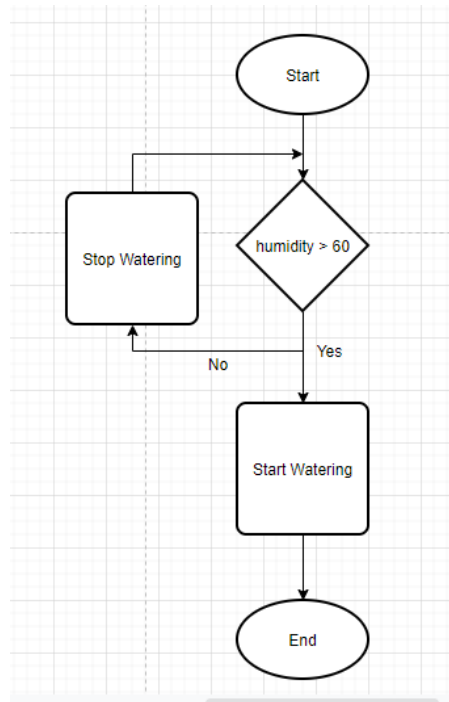
Conceptual Design

Block Diagram

Arduino Uno

x IOREF
x RESET
x 3.3V
x 5V
x GND
x Vin

A1
A2
A3
A4
A5
A6

AREF
GND
13
12
~11
~10
~9
8

7
~6
~5
4
~3
2
1
0

DATA VCC GND

DATA VCC GND

Web Page

input

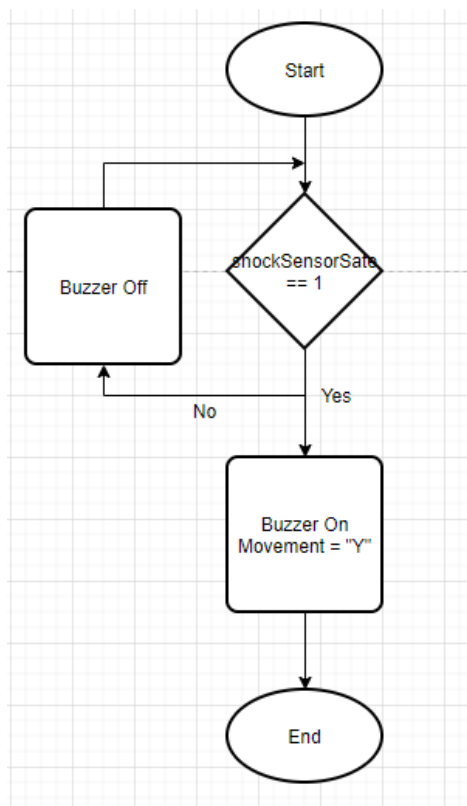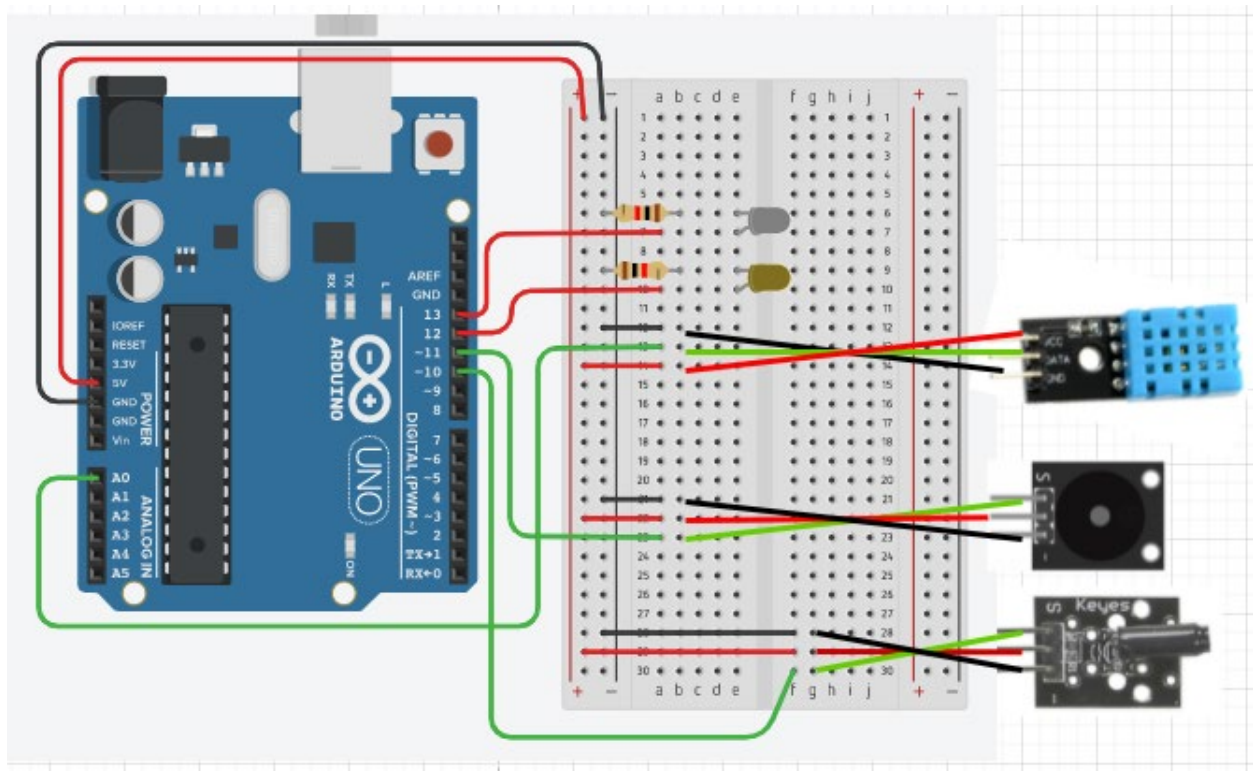| Temp and Humidity Sensor Shock Sensor	input →	Arduino	output →	LED Buzzer

# Flow Chart

Auto Watering

Buzzer

Circute Diagram

Implementation

In this System there are 2 sensor which is temp and humidity sensor and a Shock sensor , 2 LED and a Buzzer included. Moreover, this system is able to execute two different process which is Auto Watering and Buzzer.
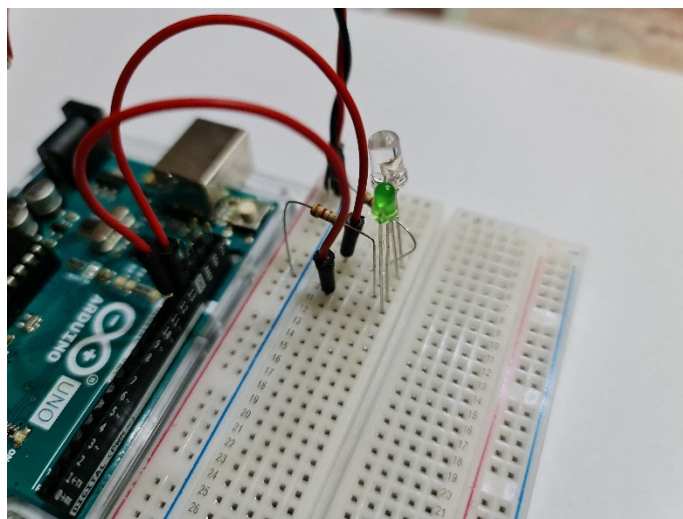
Auto Watering

For Auto Watering process, the temp & humidity sensor is used to collect the temperature and the humidity data from the plant surrounding and the 2 LEDs was used to representing whether the sprinkler is on or off because of the lack of this particular hardware.
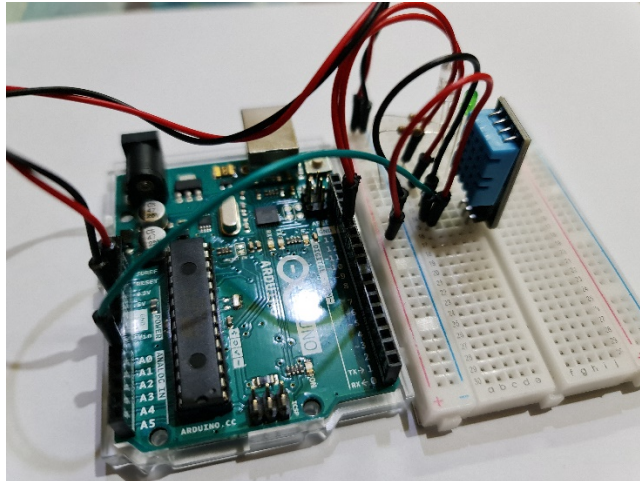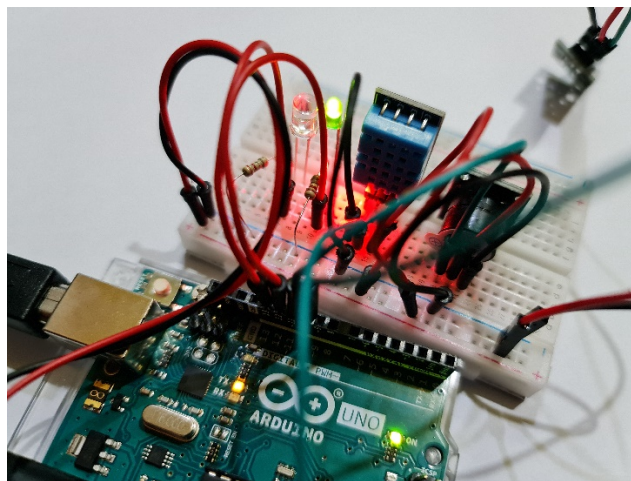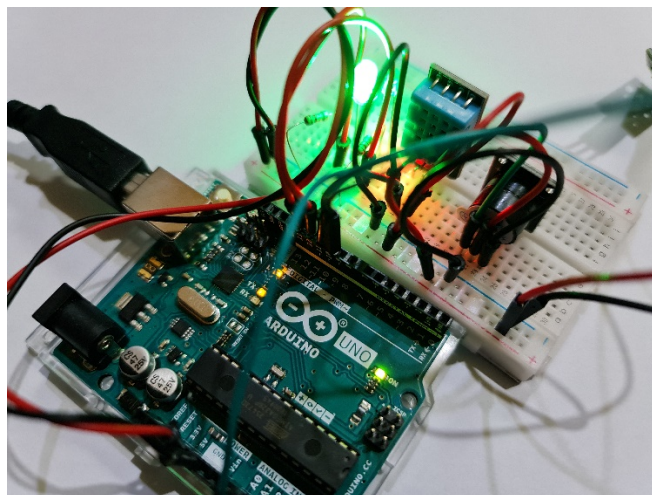
Step 1:
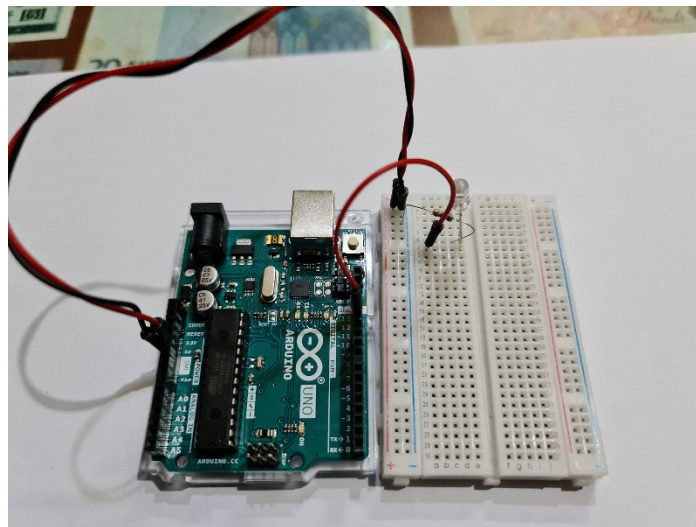
Step 2:

Step 3:

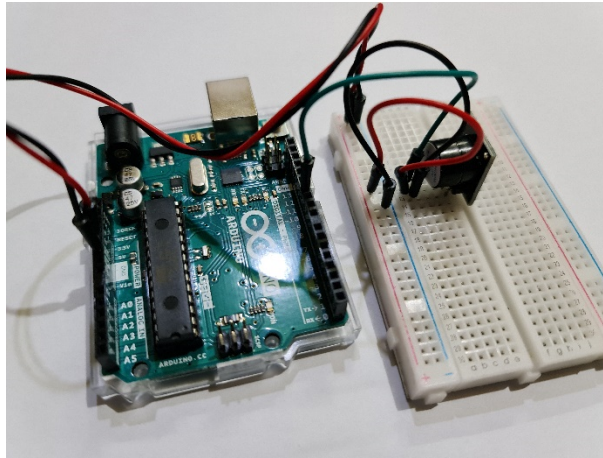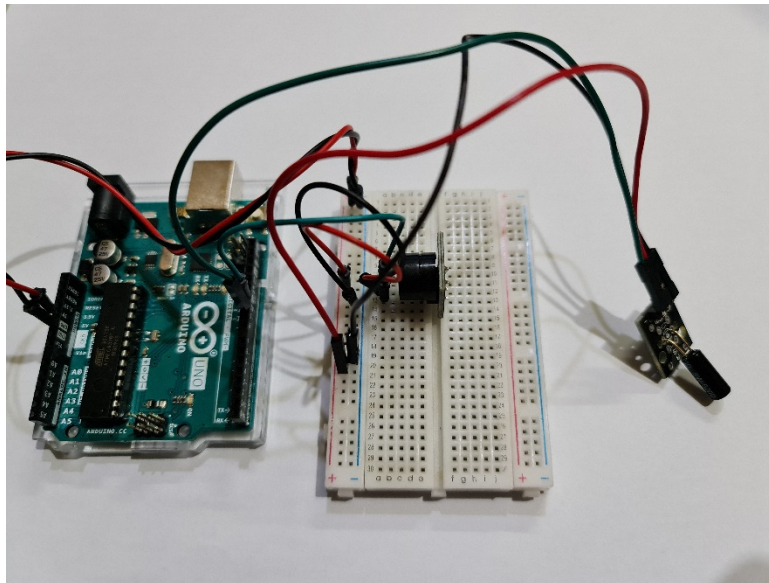When sprinkler on:

When sprinkler off:

Buzzer

For Buzzer , a shock sensor and a 3 pins Buzzer is used. Shock sensor is used to detect whether there is animal are near to the plant. If animal is closer to the plant the buzzer will be turn on to scare away the animal to prevent them damage the plant .
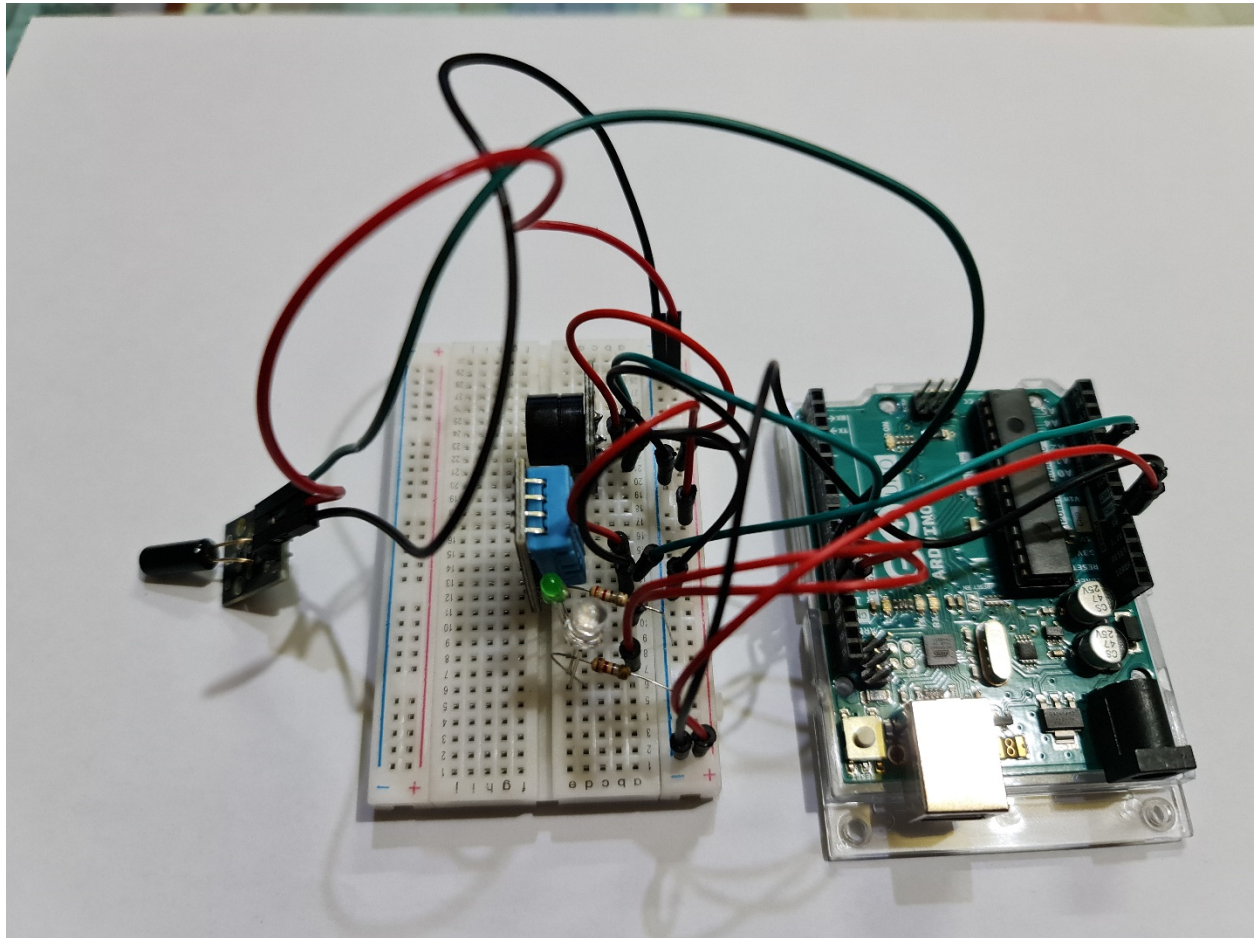
Step 1:

Step 2:

Final Look

Resources

Arduino | 13, S.C. | (2014). *Getting Started with the Arduino - Controlling the LED (Part 1)*. [online] Circuit Basics. Available at: https://www.circuitbasics.com/arduino-basics-controlling-led/#:~:text=To%20turn%20on%20an%20LED [Accessed 29 Apr. 2021].

Brainy-Bits (2020). *How to use the DHT11 Temperature and Humidity Sensor with an Arduino!* [online] Brainy-Bits Editor. Available at: https://www.brainy-bits.com/post/how-to-use-the-dht11-temperature-and-humidity-sensor-with-an-arduino [Accessed 29 Apr. 2021].

Arduino Project Hub. (n.d.). *KY-002 Shock Sensor Module*. [online] Available at: https://create.arduino.cc/projecthub/WellTronic/ky-002-shock-sensor-module-48be92 [Accessed 29 Apr. 2021].

Arduino Project Hub. (2018). *USE a BUZZER MODULE (PIEZO SPEAKER) USING ARDUINO UNO*. [online] Available at: https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45.

www.w3schools.com. (n.d.). *Python Dates*. [online] Available at: https://www.w3schools.com/python/python_datetime.asp.

MySQL Tutorial. (n.d.). *Python MySQL - Insert Data Into a Table Examples*. [online] Available at: https://www.mysqltutorial.org/python-mysql-insert/ [Accessed 29 Apr. 2021].

www.youtube.com. (n.d.). *(pyserial) communicate to arduino through python*. [online] Available at: https://www.youtube.com/watch?v=Lm_xfm1d5h0 [Accessed 29 Apr. 2021].

Stack Overflow. (n.d.). *Delete all records in a table of MYSQL in phpMyAdmin*. [online] Available at:

https://stackoverflow.com/questions/18271951/delete-all-records-in-a-table-of-mysql-in-phpmyadmin [Accessed 29 Apr. 2021].

Appendix

Arduino

```
#include "dht.h"

#define dht_apin A0 // Analog Pin sensor is connected to

#define OpenLED 12

#define CloseLED 13

#define Buzzer 11

#define Shock 10


unsigned int pinStatus = 0;

bool shockSensorSate = 0;

String Movement = "O";

String Input;

dht DHT;


void setup(){
  pinMode(OpenLED,OUTPUT);

  pinMode(CloseLED,OUTPUT);

  pinMode(Buzzer,OUTPUT);

  pinMode(Shock,INPUT);

  shockSensorSate = digitalRead(Shock);

  Serial.begin(115200);
}//end "setup()"


void loop(){
  //Start of Program

    DHT.read11(dht_apin);
```

```
if(Serial.available()>0){
  Input = Serial.readStringUntil('\n');
  if(Input == "on"){
    digitalWrite(CloseLED,LOW);
    digitalWrite(OpenLED,HIGH);
    noTone(Buzzer);
  }else if(Input == "off"){
    digitalWrite(CloseLED,HIGH);
    digitalWrite(OpenLED,LOW);
    noTone(Buzzer);
  }
  if(Input == "onplay" or Input == "offplay"){
    tone(Buzzer,1000);
    shockSensorSate = 0;
  }else if (Input == "1"){
    pinStatus = 1;
  }else if (Input =="2"){
    pinStatus = 2;
  }else if (Input =="3"){
    pinStatus = 3;
  }

  switch (pinStatus)
  {
  case 1:
    digitalWrite(CloseLED,HIGH);
    digitalWrite(OpenLED,LOW);
    noTone(Buzzer);
    break;
```

```arduino
      case 2:
        digitalWrite(OpenLED,HIGH);
        digitalWrite(CloseLED,LOW);
        noTone(Buzzer);
        break;
      case 3:
        tone(Buzzer,1000);
        break;
      default:
        break;
      }
    }


    if (shockSensorSate == 1){
      Movement = "Y";
      shockSensorSate = digitalRead(Shock);
     }
    else {
      shockSensorSate = digitalRead(Shock);
      Movement = "N";
    }
    Serial.println("Current humidity = " + String(DHT.humidity)+ "%  " + "temperature = " +
DHT.temperature + "C  " + "Movement = " + Movement);
    delay(2000);
}// end loop()
```

Database

```python
import serial

import MySQLdb

import datetime

import time

from flask import Flask, render_template


device = '/dev/ttyACM0'


arduino = serial.Serial(device,115200)


while 1:
    dbConn = MySQLdb.connect("localhost","pi","","PlantCare_db") or die("Could not connect to the database")

    while(arduino.in_waiting == 0):
        pass
    line = arduino.readline()
    Humidity = str(line[19:25])
    HumidityInt = float(line[19:21])
    Temp = str(line[41:47])
    TempInt = float(line[41:43])
    Movement = str(line[60:61])
    Today = datetime.datetime.now()
    Date = str(Today.strftime("%x"))
    Time = str(Today.strftime("%X"))

    command = "";
```

```python
print(arduino.readline().decode('ascii'))


if(HumidityInt < 60):

    command += "on"

    #arduino.write("on".encode())

    Watering = "Yes"
elif(HumidityInt >= 60):

    command += "off"

    #arduino.write("off".encode())

    Watering = "No"


if(Movement == "b'Y'"):

    command +="play"

    #arduino.write("play".encode())

    Movement2 = "Yes"
else:

    command += ""

    Movement2 = "No"


if(command !="" ):

    arduino.write(command.encode())

    #print(command)



#print (Movement)
time.sleep(0.2)
```

```python
with dbConn:

    cursor = dbConn.cursor()

    sql = "INSERT INTO activityLog (Date, Time, Humility, Temperature, Shocked, Watering) VALUES (%s,%s,%s,%s,%s,%s)"

    value = (Date, Time, str(HumidityInt) + "%", str(TempInt) + "C", Movement2, Watering)

    cursor.execute(sql,value)

    dbConn.commit()

    cursor.close()
```

Web

```
# led_control.py

import serial

import time

import MySQLdb

import datetime

from flask import Flask, render_template

app = Flask (__name__)
# Dictionary of pins with name of pin and state ON/OFF


# Main function when accessing the website

@app.route("/")

def index():

 # TODO: Read the status of the pins ON/OFF and update dictionary

 # This data will be sent to index.html (pin dictionary)

 templateData = {

    'pins1' : data1

 }

 # Pass the template data into the template index.html and return it

 return render_template('index.php', **templateData)

# Function to send simple commands

@app.route("/<action>")

def action(action):

 data1=list()

 if action == 'action1' :

    ser.write("1")

 if action == 'action2' :

    ser.write("2")

 if action == 'action3' :
```

```python
    ser.write("3")
if action == 'action4' :

  command = ""

  line = arduino.readline()

  Humidity = str(line[19:25])

  HumidityInt = float(line[19:21])

  Temp = str(line[41:47])

  TempInt = float(line[41:43])

  Movement = str(line[60:61])

  Today = datetime.datetime.now()

  Date = str(Today.strftime("%x"))

  Time = str(Today.strftime("%X"))


  if(HumidityInt > 60):

    command += "on"

    #arduino.write("on".encode())

    Watering = "Yes"
  elif(HumidityInt <= 60):

    command += "off"

    #arduino.write("off".encode())

    Watering = "No"


  if(Movement == "b'Y'"):

    command +="play"

    #arduino.write("play".encode())

    Movement2 = "Yes"
  else:

    command += ""

    Movement2 = "No"
```

```python
    if(command !="" ):

        ser.write(command.encode())

        #print(command)


    with dbConn:

        cursor = dbConn.cursor()

        sql = "INSERT INTO activityLog (Date, Time, Humility, Temperature, Shocked, Watering) VALUES
(%s,%s,%s,%s,%s,%s)"

        value = (Date, Time, str(HumidityInt) + "%", str(TempInt) + "C", Movement2, Watering)

        cursor.execute(sql,value)

        cursor.execute("SELECT * FROM activityLog")

        dataList = cursor.fetchall()

        for x in dataList:

            data1.append(x)

        cursor.close()

    with dbConn:

        cursor = dbConn.cursor()

        cursor.execute("SELECT * FROM activityLog")

        dataList = cursor.fetchall()

        for x in dataList:

            data1.append(x)

        cursor.close()

    # This data will be sent to index.html (pins dictionary)

    templateData = {

        'pins1' : data1

    }

    # Pass the template data into the template index.html and return it

    return render_template('index.php', **templateData)
```

```python
# Main function, set up serial bus, indicate port for the webserver, and start the service
if __name__ == "__main__" :
    data1 = list();
    device = '/dev/ttyACM0'
    ser = serial.Serial(device, 115200, timeout = 1)
    arduino = serial.Serial(device,115200)


    dbConn = MySQLdb.connect("localhost","pi","","PlantCare_db") or die("Could not connect to the database")
    with dbConn:
        cursor = dbConn.cursor()
        cursor.execute("SELECT * FROM activityLog")
        dataList = cursor.fetchall()
        for x in dataList:
            data1.append(x)
        cursor.close()
    ser.flush()
    app.run(host='0.0.0.0', port = 80, debug = True)
```

HTML

```html
<!DOCTYPE html>
  <head>
    <title>Plant Care System Server</title>


  </head>
  <body>

    <h1> Plant Care System Server </h1>
    <h2> Commands </h2>
    <h3><a href="/action1" > Watering </a></h3>
    <h3><a href="/action2" > Stop Watering </a></h3>
    <h3><a href="/action3" > Play Buzzer </a></h3>

    <h1> Table of Data </h1>
    <p><a href="/action4" > Update Table </a></p>
    <table border = "1">
        <tr>
          <th>Activity ID</th>
          <th>Date</th>
          <th>Time</th>
          <th>Humidity</th>
          <th>Temparature</th>
          <th>Shocked</th>
          <th>Watering</th>
        </tr>
        {% for pin in pins1 %}
        <tr>
          <td>{{pin[0]}}</td>
          <td>{{pin[1]}}</td>
          <td>{{pin[2]}}</td>
          <td>{{pin[3]}}</td>
          <td>{{pin[4]}}</td>
          <td>{{pin[5]}}</td>
          <td>{{pin[6]}}</td>
        </tr>
        {% endfor %}
        </table>


    <!--<h2> Toggle buttons </h2>
    {% for pin in pins %}
      <h3>{{ pins[pin].name }}
      {% if pins[pin].state == 1 %}
```

```
            is currently <strong>on</strong></h3>
        <div class="row"><div class="col-md-2">
        <a href="/{{pin}}/off" class="btn btn-block btn-
lg btndefault" role="button">Turn off</a>
        </div></div>
      {% else %}
        is currently <strong>off</strong></h3>
        <div class="row"><div class="col-md-2">
        <a href="/{{pin}}/on" class="btn btn-block btn-lg btn-primary"
role="button">Turn on</a>
        </div></div>
      {% endif %}
    {% endfor %}-->
  </body>
</html>
```