UNIVERSITY AT ALBANY
State University of New York

IECE 553

Project

Smart Home Garden applying Hydroponics

Arijit Sengupta
Shuvam Chakraborty

DECEMBER 04, 2019

Why smart home garden again?

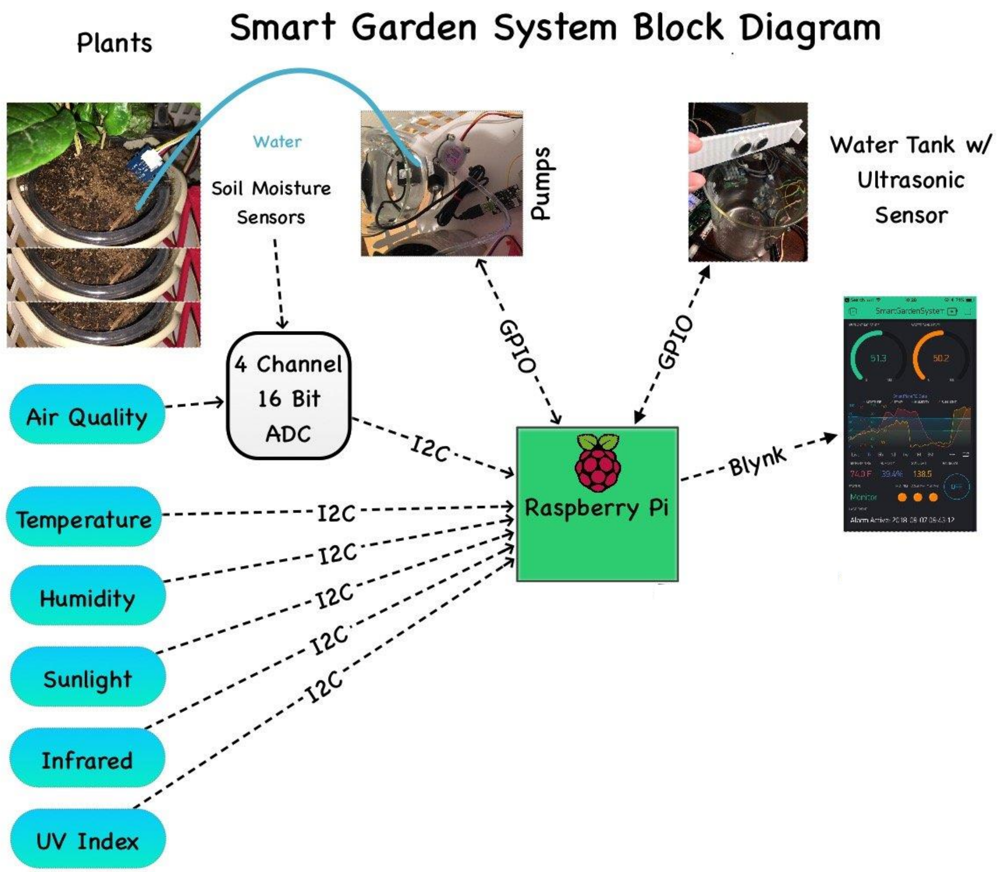
- Smart home garden is a system where plant growth is monitored using a plethora of sensors and the growth environment as required.
- It has started becoming popular for last few years because of ease of automation employing smart phones and other intelligent systems as master devices.
- Incorporating hydroponics in smart home garden combines its efficiency and environmental sustainability with the portability of the latter and forms the main focus of this project.

What is Hydroponics?

- As the name suggests Hydroponics is the method of plant growth without using soil as primary nutrition source.
- Water with different nutrient solution is used to grow plants.
- It requires 8-10% of the amount of water required for traditional method of plant growth.
- Plants does not need pesticides.
- Takes a lot less space to design the system.
- Some studies suggest specific plants have shown higher growth rate in this method.

Motivation

- Unchecked household water utilization in towns and metropolitan cities often lead to scarcity of water supply locally.
- Again with increase in automation in domestic devices has somewhat necessitated the ease of use in home garden handling, which leads us to smart home garden.
- Incorporation of hydroponics into this makes the system eco-friendly.

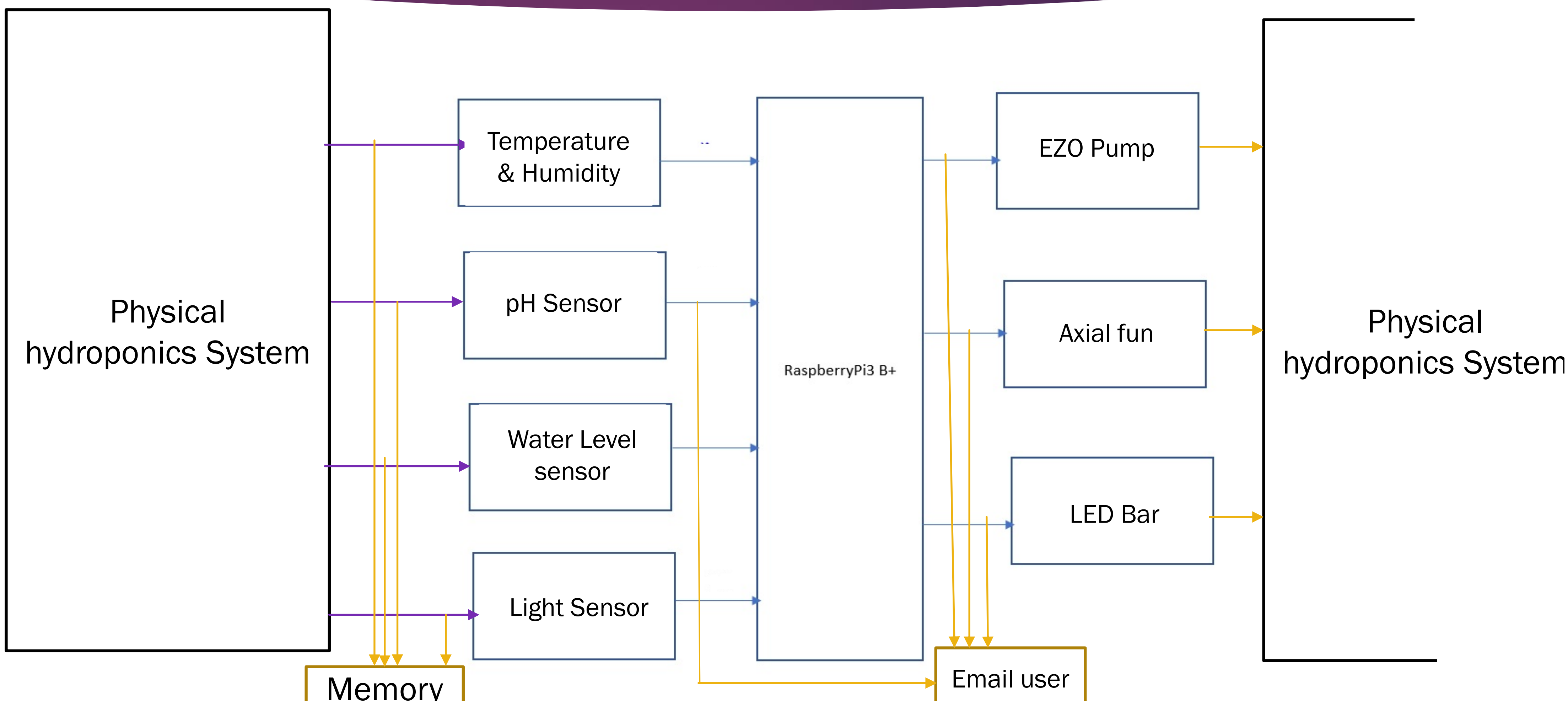


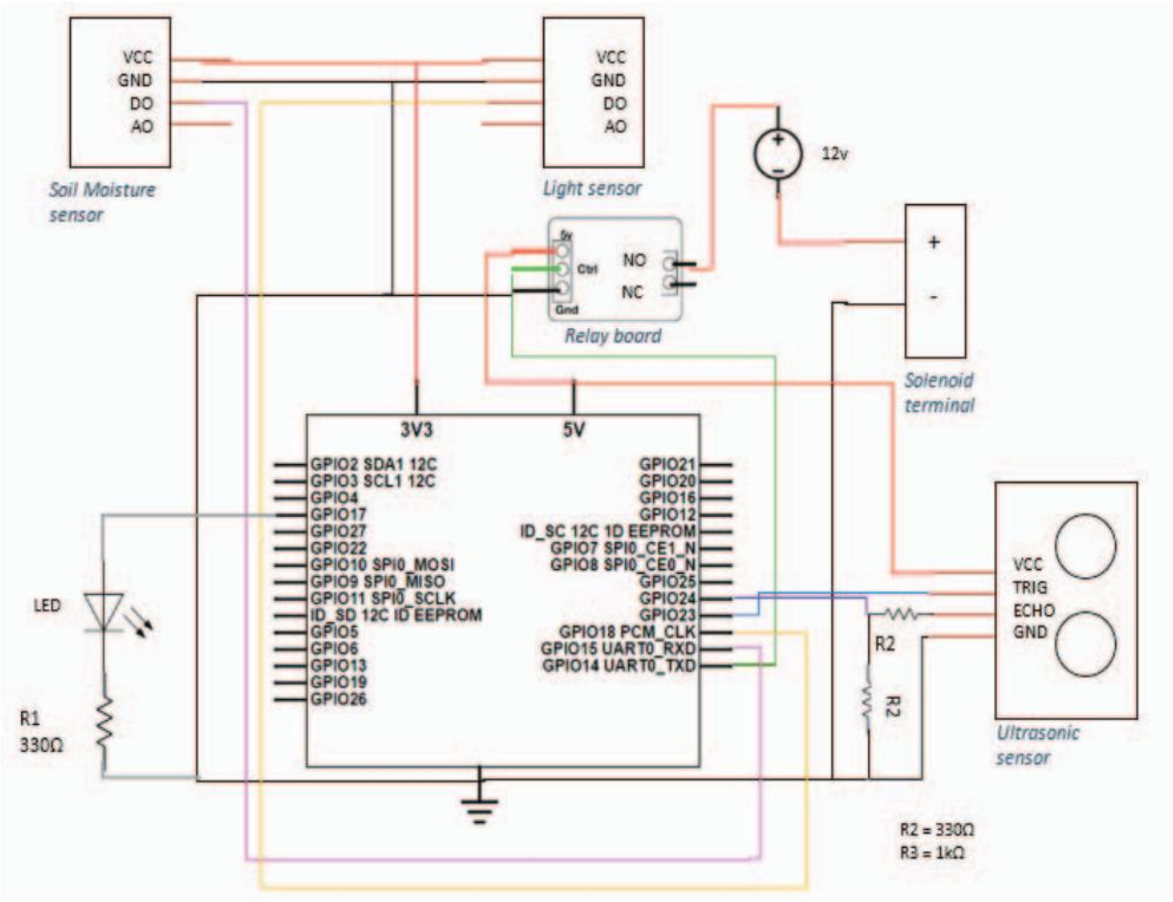
System model:

- Modelling consists of multi-sensor fusion.
- Continuum of synchronous sensor reading.
- Synchronous actuator actions.
- Carefully generating threshold values for optimal system.
- Since the project is done in a place with severe weather variation and weather not only important but the only controlling parameter, the system model needs to be updated.
- The following block diagram shows the entire system model in a nut-shell.

System model

Fig 1





Circuit block:

Circuit Block

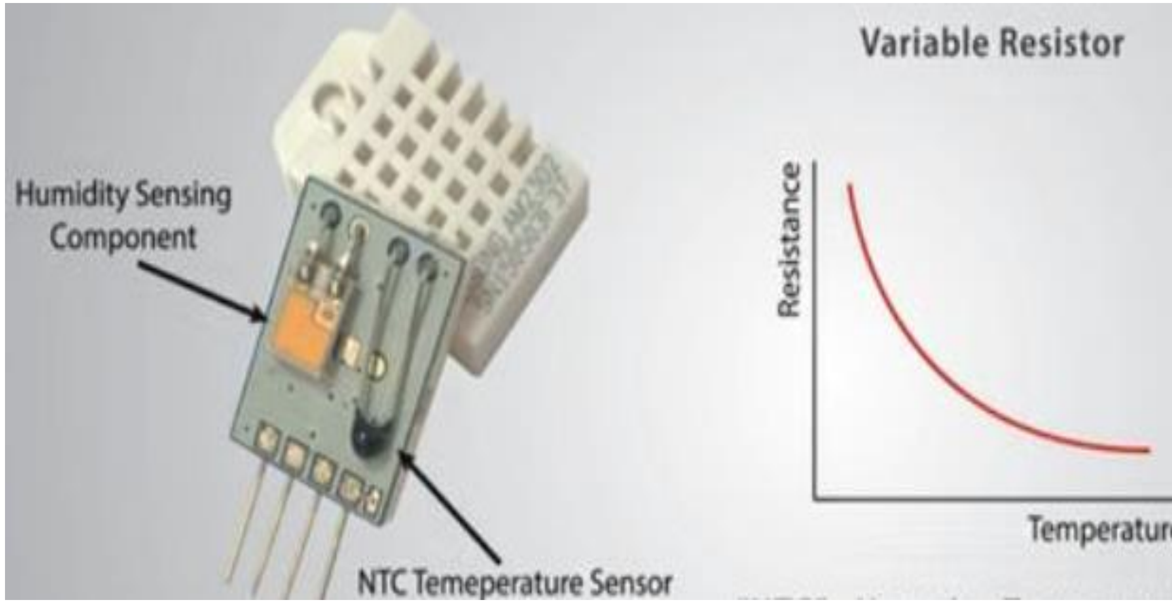
- The above circuit block shows a generic circuit diagram of the system.
- The relay module used here is a generic single pole single throw relay module which is the simplest form of relay module.
- The ultrasonic sensor inputs and connections are not maintained in the exact same way.
- Since there are no soil used. Instead, a mixture of compost and tea leaves is used, the soil moisture sensor is absent here as well.
- Further being generic, we will see many differences between these two systems.

Sensor Models:

- We will discuss individual sensors and there models in the following sections, owing to our requirements…

Temperature sensing model:

- The block diagram used earlier effectively sums up the model of the *Weather Monitoring System*.
- Temperature and humidity is the most important component for plant growth. It is measured using a negative temperature coefficient sensor along with a separate humidity sensing component. Since the values are obtained up to two decimal point float it is precise enough for our cause.
- Required Temperature is assumed 25*C for optimal plant growth, and humidity is assumed 60%.

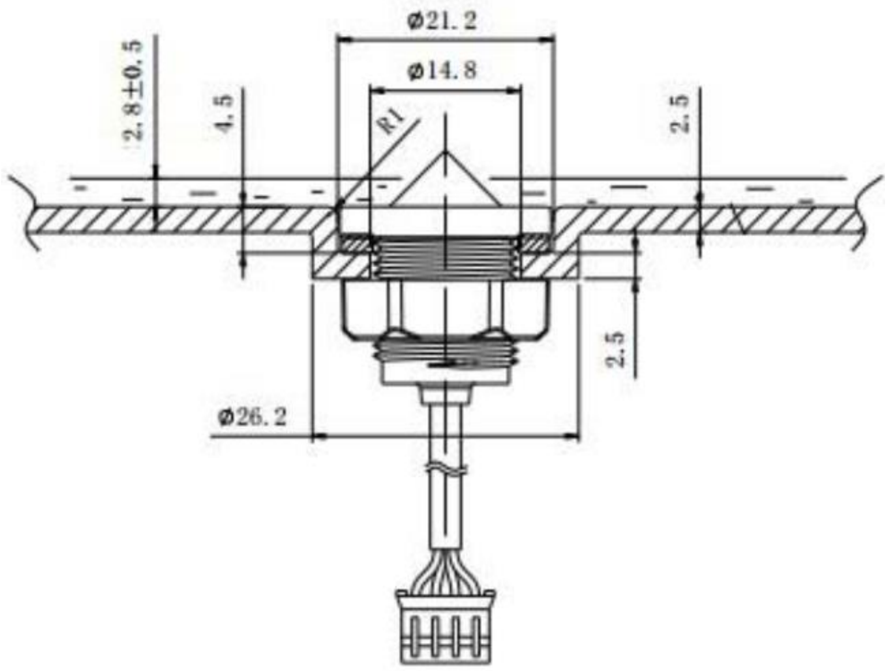


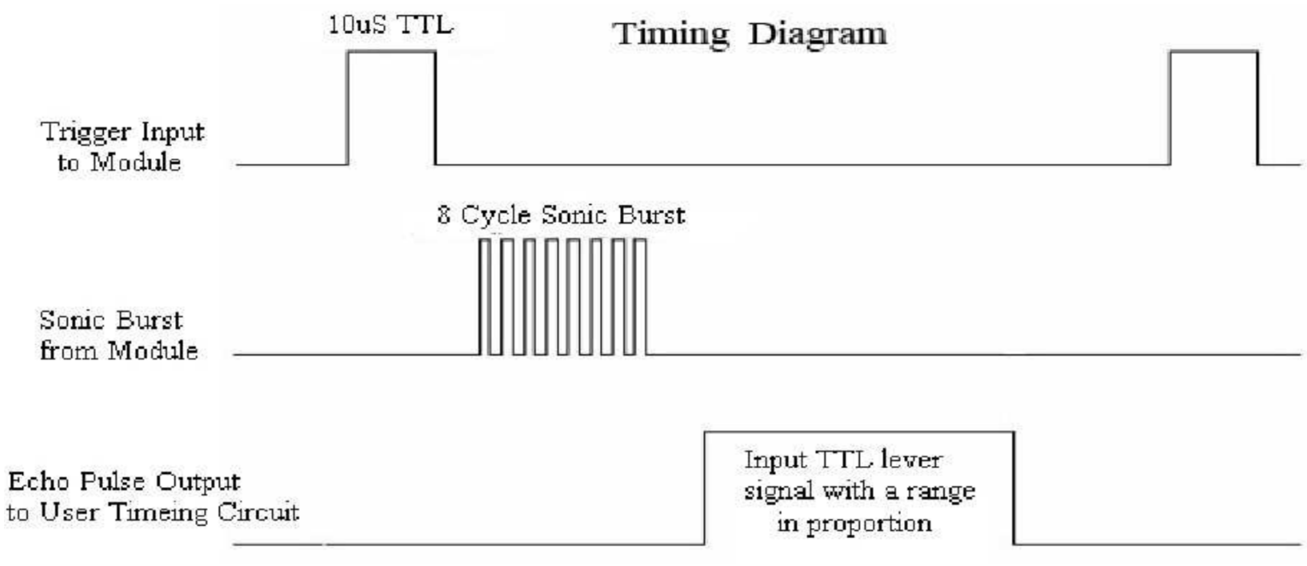
Water level control

- The water level is said to be in tolerable limit if it is within 5-14 cm limit. The sensor is a super sonic sensor.
- It can detect the sound reflection which in turn gives coarse result of height of water column in the reservoir.
- Again a photo electric water level sensing mechanism is used which probes different output than default when it comes in contact with liquid.
- As soon as water level goes lower or higher than tolerable limit, the corresponding actuator is triggered to either reduce or increase the water level.

UNIVERSITY AT ALBANY
State University of New York

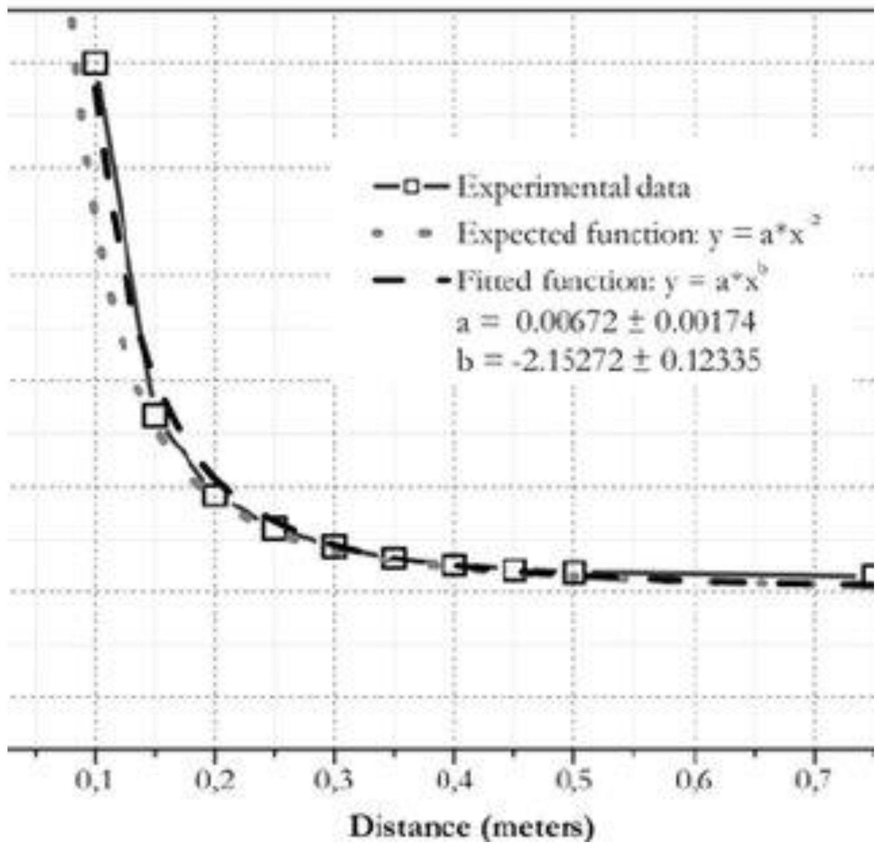
Water level sensing model





Light sensing and control

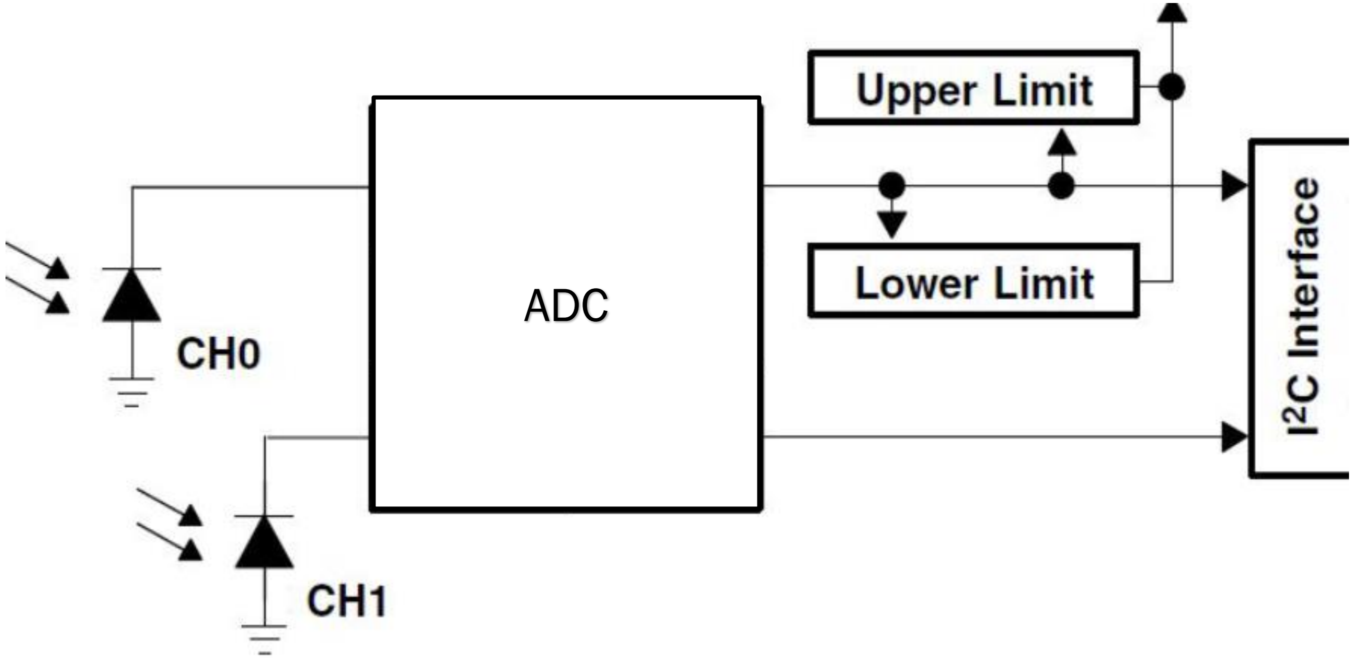
- Light sensing can be most easily controlled by a photoresistor which generates voltage proportional to the incident light, simulating a varying resistance
- It implements a voltage divider circuit internally.
- Calculations show required luminosity level for optimal plant growth generates an ADC output from the entire photoresistor set-up is almost 190.



Source: researchgate.net

Light sensing and control

- This depicts another possible way of light sensing.
- A threshold of ambient light intensity is fixed. The figure shows the photodiode inputs will be converted to digital value which governs the control of the additional LED bar.
- If light intensity goes below that, a pwm module turn on the light and controls the additional light to the threshold intensity level.



Sensors and Actuators

The three main components of this system are:

- Temperature and Humidity
- Light
- Water Level

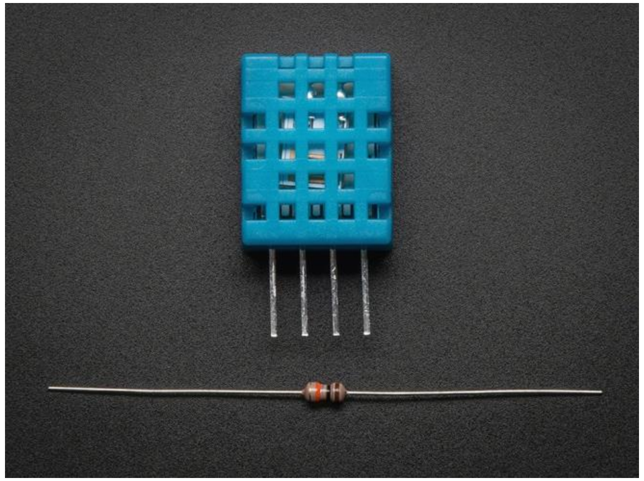
Sensors		Actuators
1. DHT11 Temperature and Humidity Sensor	→	12V Axial Fan
2. TSL2591 High Dynamic Range Digital Light Sensor	→	LED Light Bar, Daylight
3. SEN0205 Photoelectric Water / Liquid Level Sensor using Relay Module and Ultrasonic Sensor	→	Peristaltic EZO Pump and 2-way Solenoidal Valve

Sensors and Actuators:

- In the following section we will describe working of different sensors and corresponding actuators and other additional hardware.
- Besides hardware, we will mention the software requirements and other design aspects as well.
- We will also mention the limitations we faced along our path to model and design the entire project.

DHT11 Temperature and Humidity Sensor

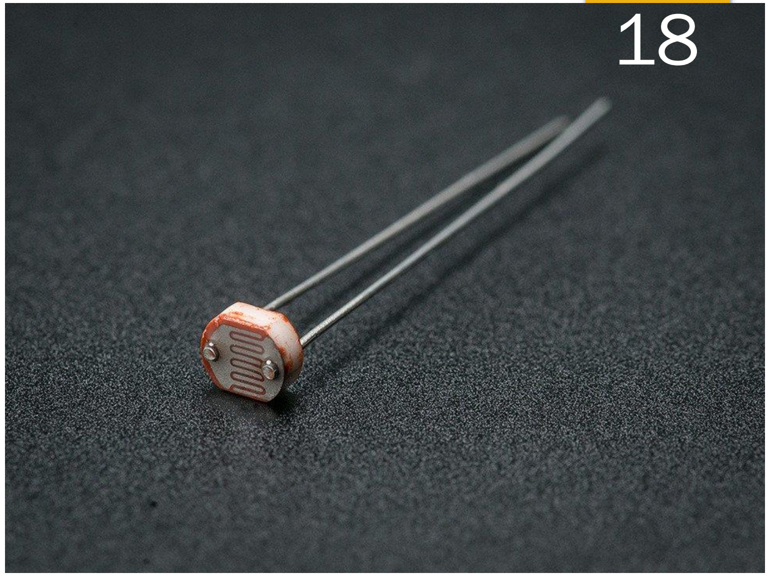
- Initialize DHT Pin elements to 0 and set PINMODE to output
- Read value using read_dht11_dat() function and display on 16x2 LCD screen
- Check if fanauto == true is enabled
- Check if Temperature >= 25, turn ON the fan by digitalWrite (pin number, HIGH) else Temperature < 25, turn OFF the fan by digitalWrite (pin number, LOW)

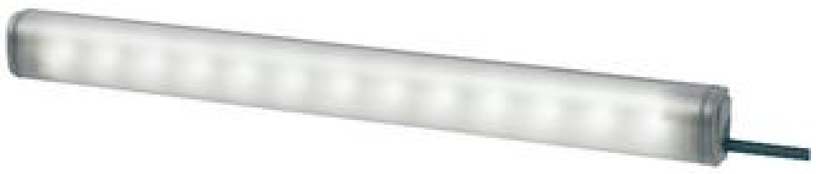


https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet

Light Sensor (Photocell)

- The range of the photocell is 0 to 255
- Whenever the value is less than 190, a buzzer goes off to notify the user to turn on artificial lighting
- The artificial lighting is turned on by the user



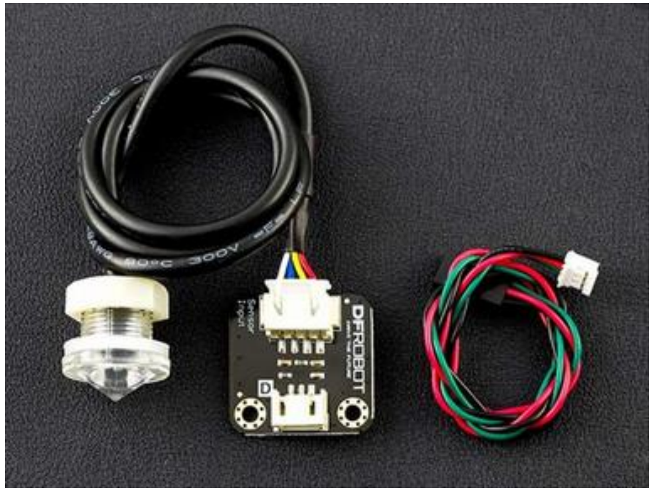


https://cdn-shop.adafruit.com/datasheets/TSL25911_Datasheet

SEN0205 Photoelectric Water / Liquid Level Sensor using Automation pHAT Relay Module

- Initialize the waterlevelauto as false, input water level as 0 and max water level as 15
- Read input water level keeping the pump on until it reaches 5 cm. The water level should lie between 5cm and 14cm.
- The ultrasonic sensor measures the input water level, by calculating distance from the sound wave travel time.

https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0205



Source: https://cdn.sparkfun.com/datasheets/Sensors/Proximity

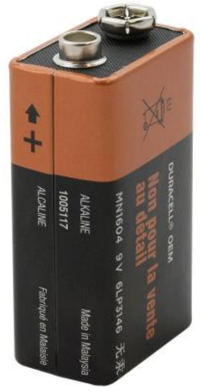
SEN0205 Photoelectric Water / Liquid Level Sensor using Automation pHAT Relay Module

- If it is less or more than this range, a warning message is displayed to the user to manually turn off the pump.
- Using the 2-way valve, the excess water can be removed as a continuous loop runs checking the above mentioned condition using the Automation pHAT Relay Module and sets the pin value to HIGH or LOW for the valve and water pump accordingly.

https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0205

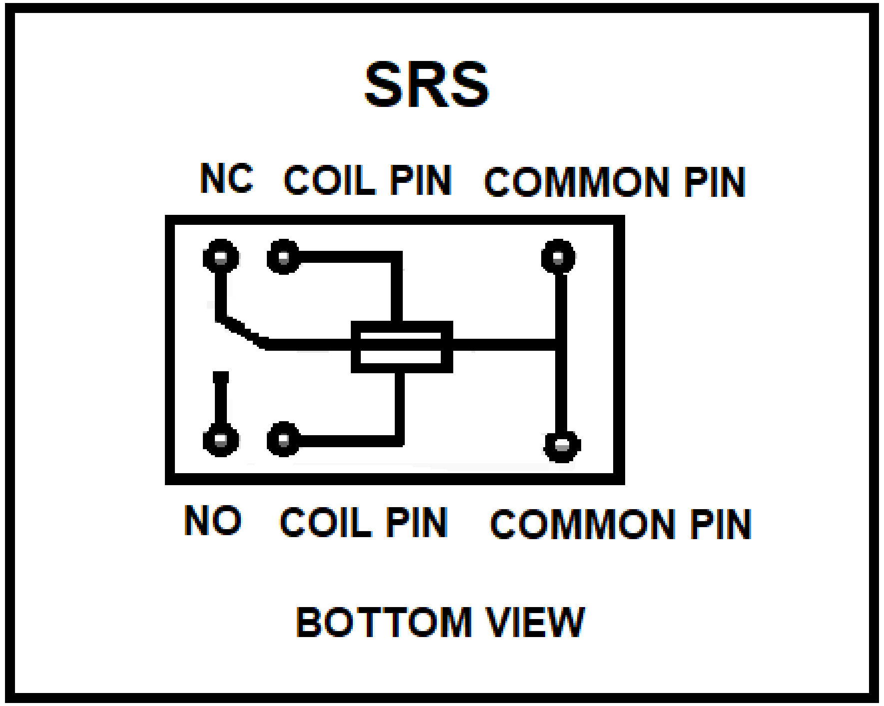
Additional hardware requirement:

- The axial fan, ezo pump and the solenoidal valve all of them has a voltage rating of 12V with minimum current 0.1A.
- Duracell dry cell 12V batteries are used to provide necessary power to these actuators, since raspberry pi can not provide 12V supply.
- SRS-5V DPST relay module has been used for each of these sensors to control the output accordingly with the variation of sensor results.



Working of the relay module:

- As depicted in the pin-out diagram, NC or NO requires the output voltage that is required.
- Coil needs 5V activation which is clear from the name.
- The common ports generate the output. We have used the NC(normally closed) terminal for all three actuators.



https://www.cytron.io/p-songle-relay-srs-03vdc-sh

Software requirement:

- Primarily embedded C programming language is used for the scripts.
- Further API required for pHAT and TSL039.
- MATLAB2019 is used for the data analysis and weather prediction.

Optimal Composite Implementation of Sensors and Actuators

Using a char array, readings from the sensors are taken as input and the actuators are automated accordingly

A switch-case implementation is done for each sensor, which contains nested for loops within it for each mode

The EZO water pump and the valve has either ON or OFF mode

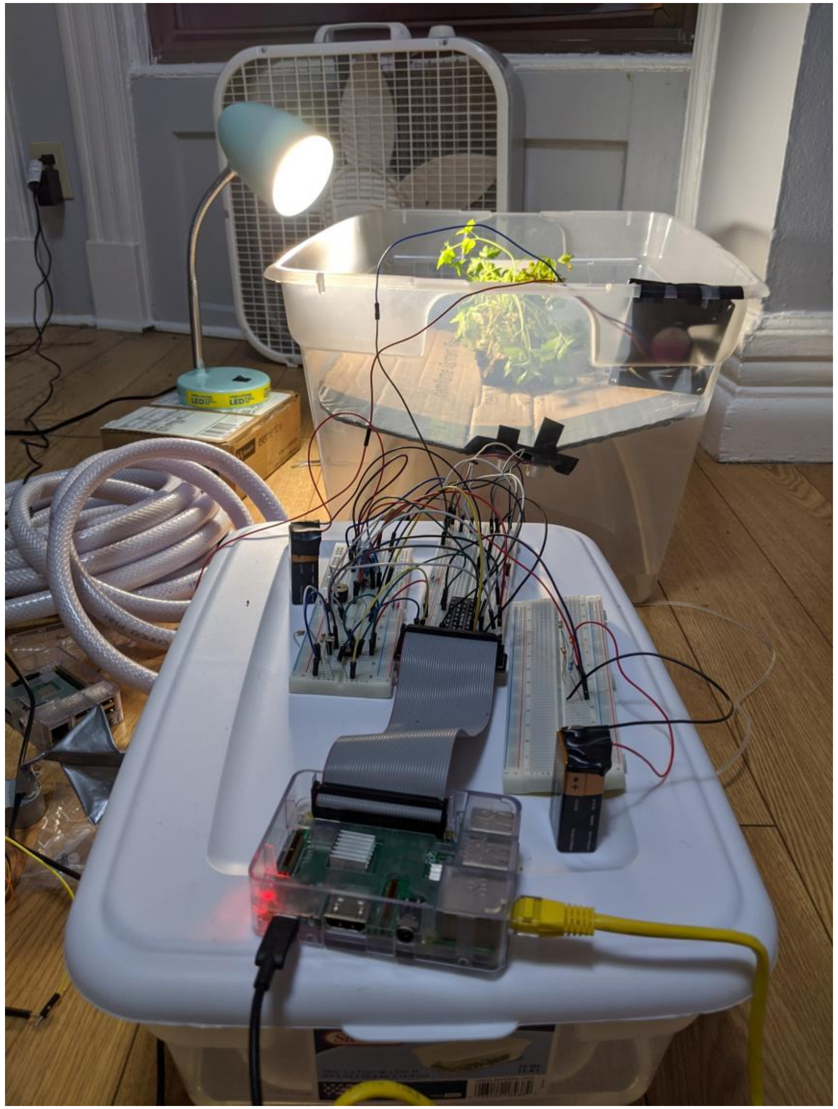
The Axial fan and the LED bar can be implemented with 3 modes namely auto, switch and remote mode which will be set accordingly as per the returned raw values from the sensors

Experiments:

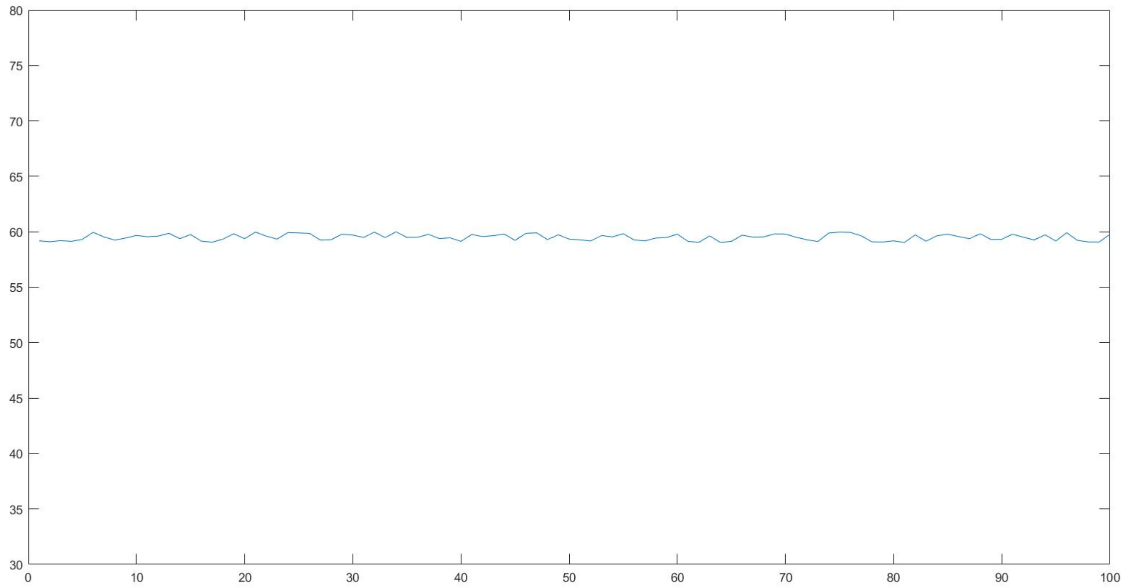
- The following section shows the scripts written for different sensor actuator pair and their corresponding output in small clips.
- We have also included the sensor results in the following section.

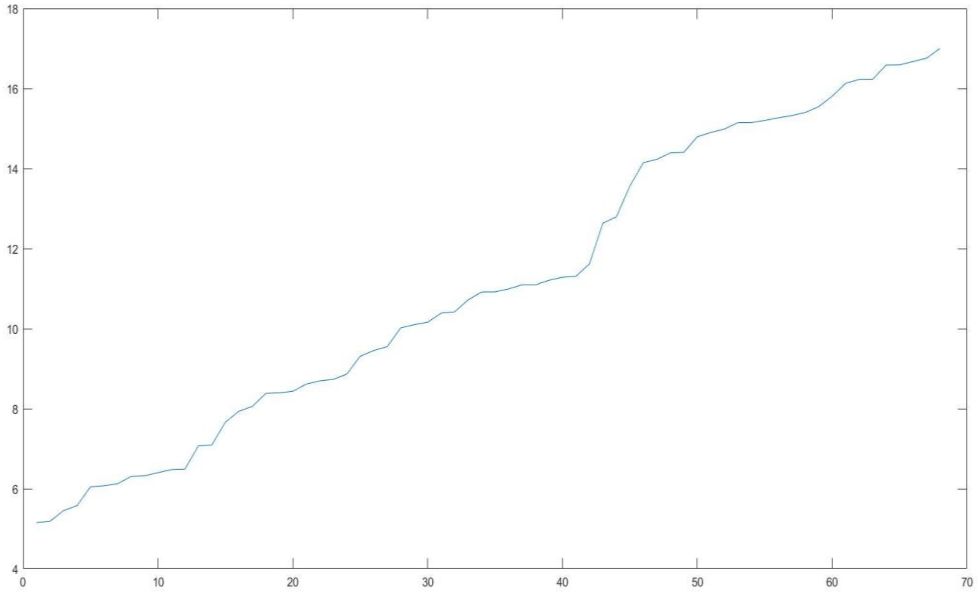
Implementation

The image shows the entire set-up along with circuits, control unit Raspberry-Pi and the physical plant system.

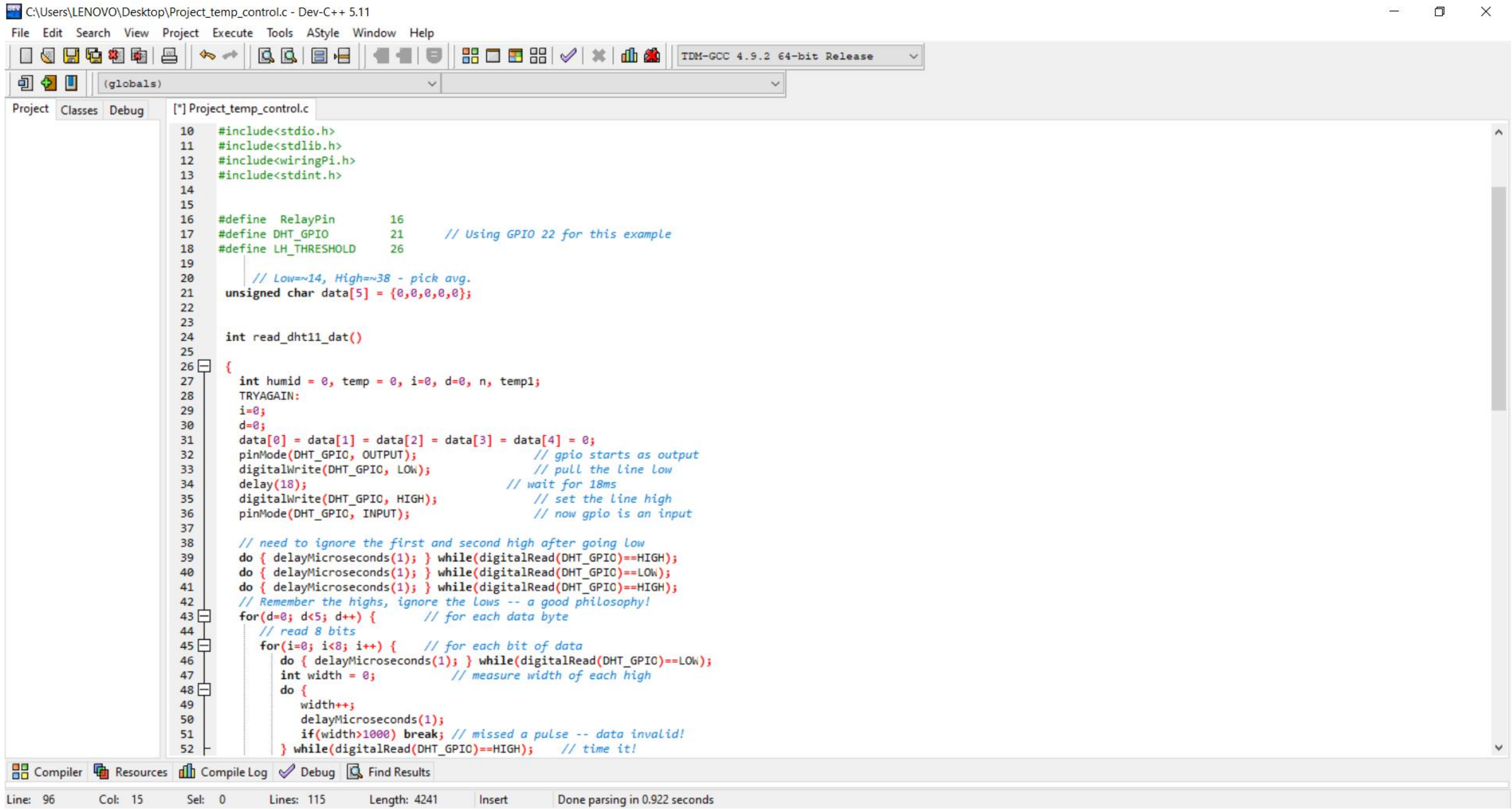


Temperature and humidity control:



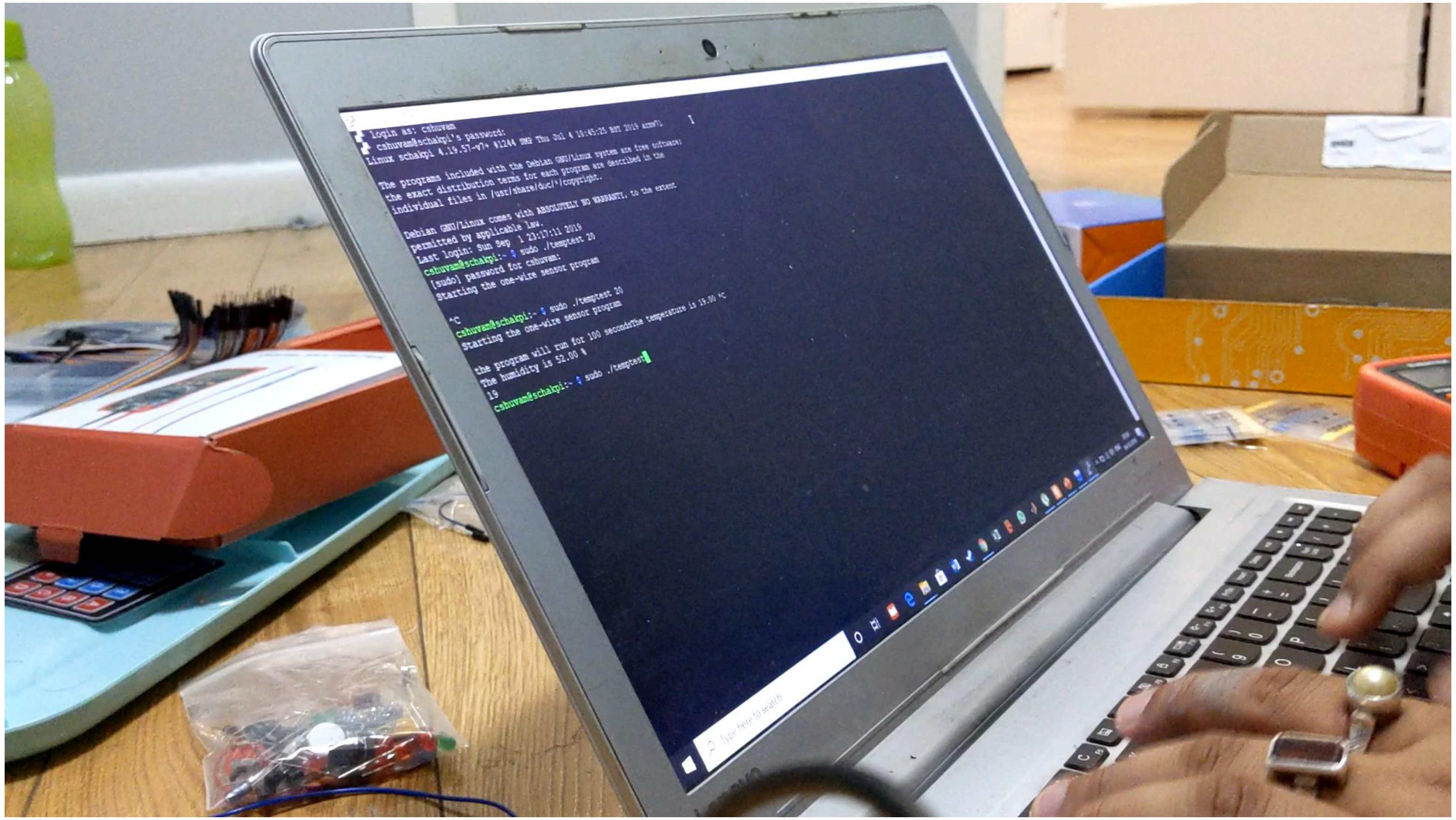


Temperature and humidity control:

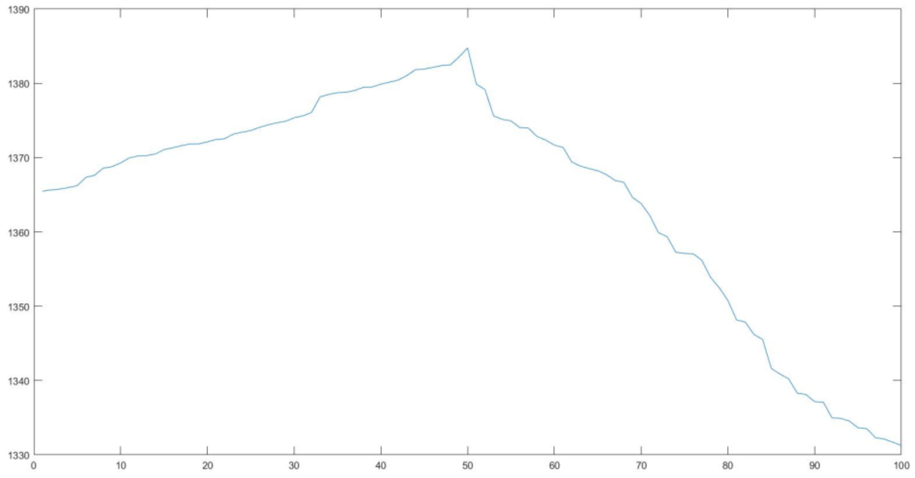


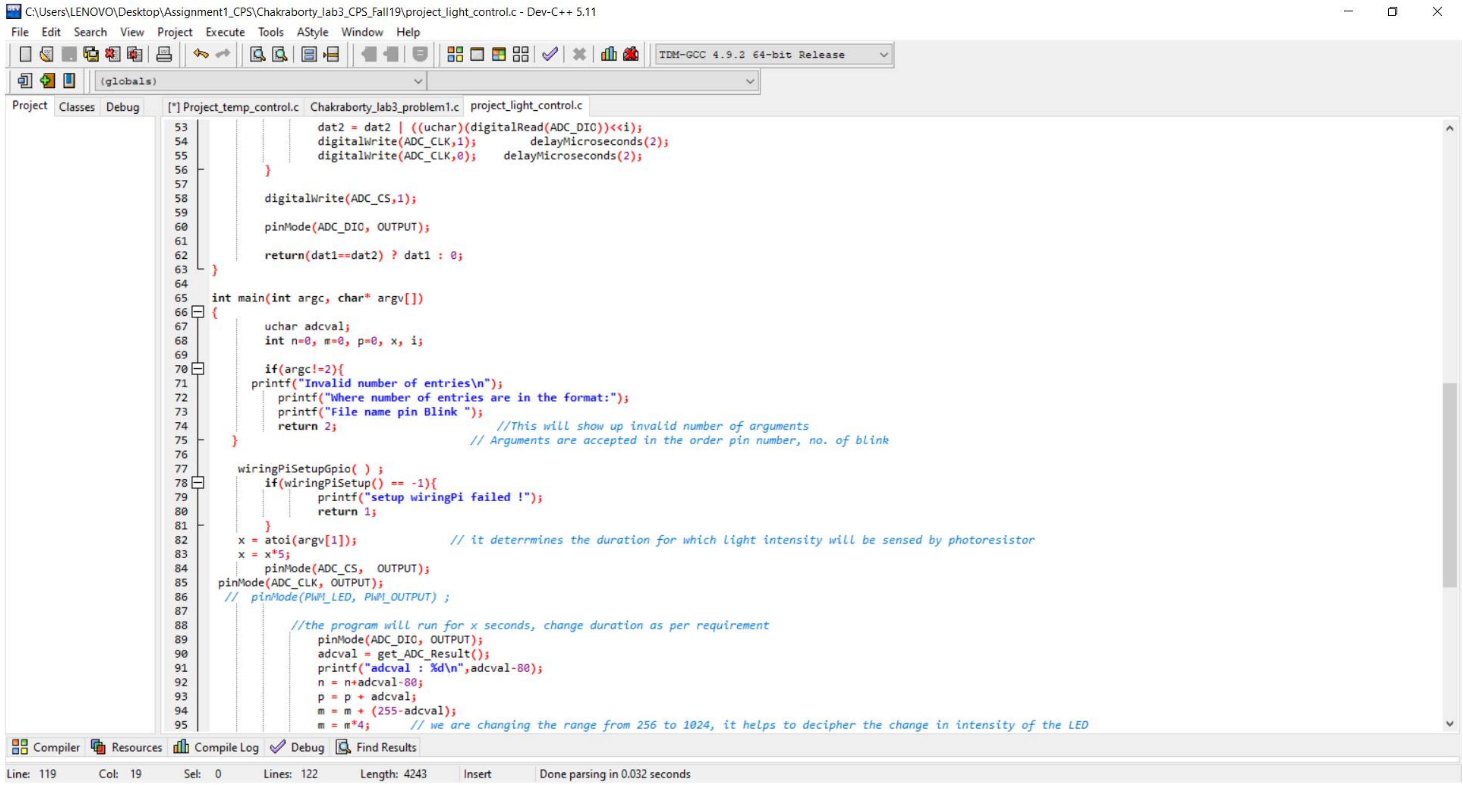

```c
#include<stdio.h>
#include<stdlib.h>
#include<wiringPi.h>
#include<stdint.h>


#define  RelayPin        16
#define DHT_GPIO         21      // Using GPIO 22 for this example
#define LH_THRESHOLD     26

    // Low=~14, High=~38 - pick avg.
 unsigned char data[5] = {0,0,0,0,0};


 int read_dht11_dat()

 {
   int humid = 0, temp = 0, i=0, d=0, n, temp1;
   TRYAGAIN:
   i=0;
   d=0;
   data[0] = data[1] = data[2] = data[3] = data[4] = 0;
   pinMode(DHT_GPIO, OUTPUT);                   // gpio starts as output
   digitalWrite(DHT_GPIO, LOW);                 // pull the line low
   delay(18);                                // wait for 18ms
   digitalWrite(DHT_GPIO, HIGH);                // set the line high
   pinMode(DHT_GPIO, INPUT);                    // now gpio is an input

   // need to ignore the first and second high after going low
   do { delayMicroseconds(1); } while(digitalRead(DHT_GPIO)==HIGH);
   do { delayMicroseconds(1); } while(digitalRead(DHT_GPIO)==LOW);
   do { delayMicroseconds(1); } while(digitalRead(DHT_GPIO)==HIGH);
   // Remember the highs, ignore the lows -- a good philosophy!
   for(d=0; d<5; d++) {       // for each data byte
      // read 8 bits
      for(i=0; i<8; i++) {    // for each bit of data
         do { delayMicroseconds(1); } while(digitalRead(DHT_GPIO)==LOW);
         int width = 0;           // measure width of each high
         do {
            width++;
            delayMicroseconds(1);
            if(width>1000) break; // missed a pulse -- data invalid!
         } while(digitalRead(DHT_GPIO)==HIGH);    // time it!
```

Temperature and humidity control:

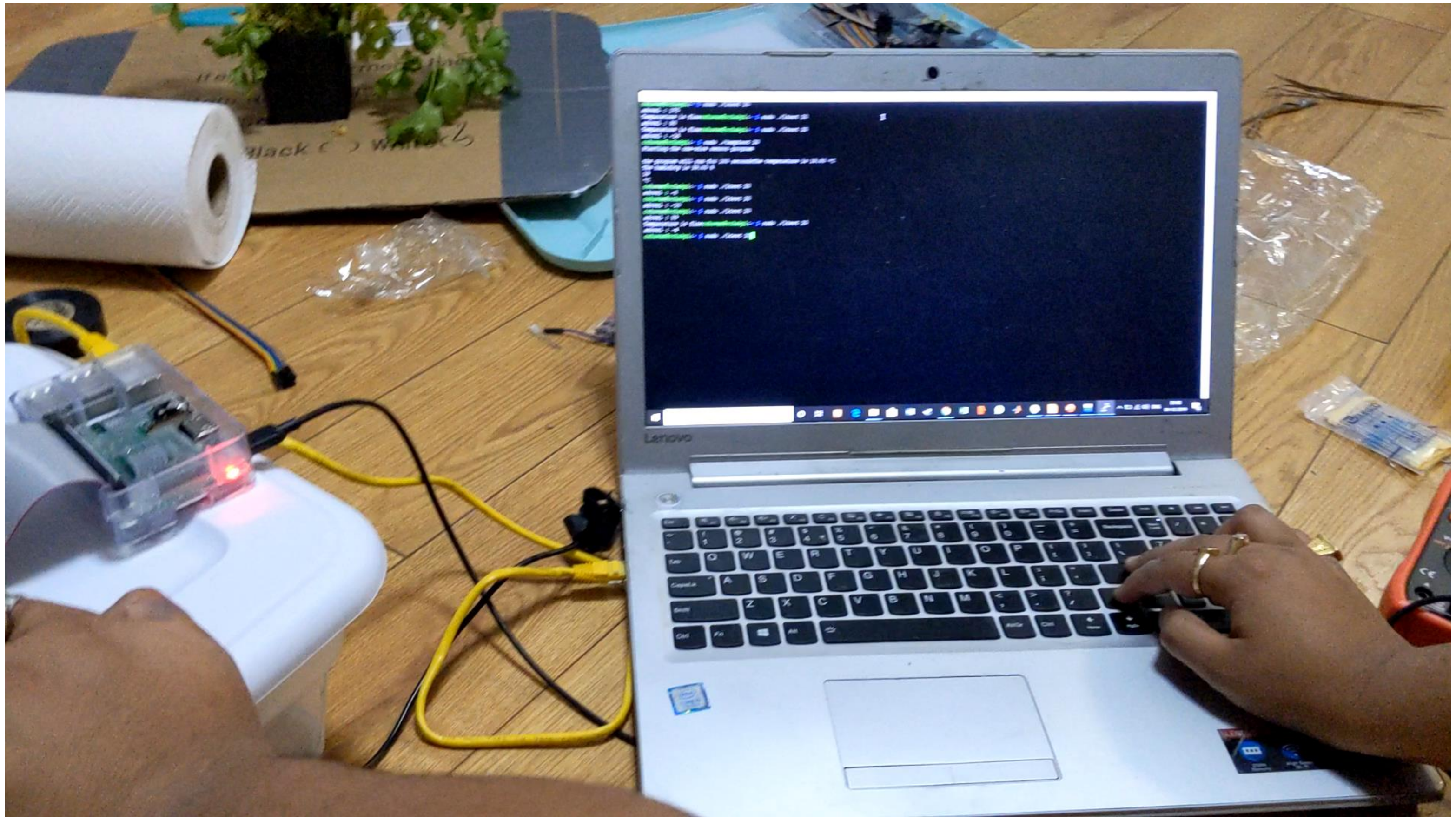


Ambient light monitoring:





Ambient light monitoring:



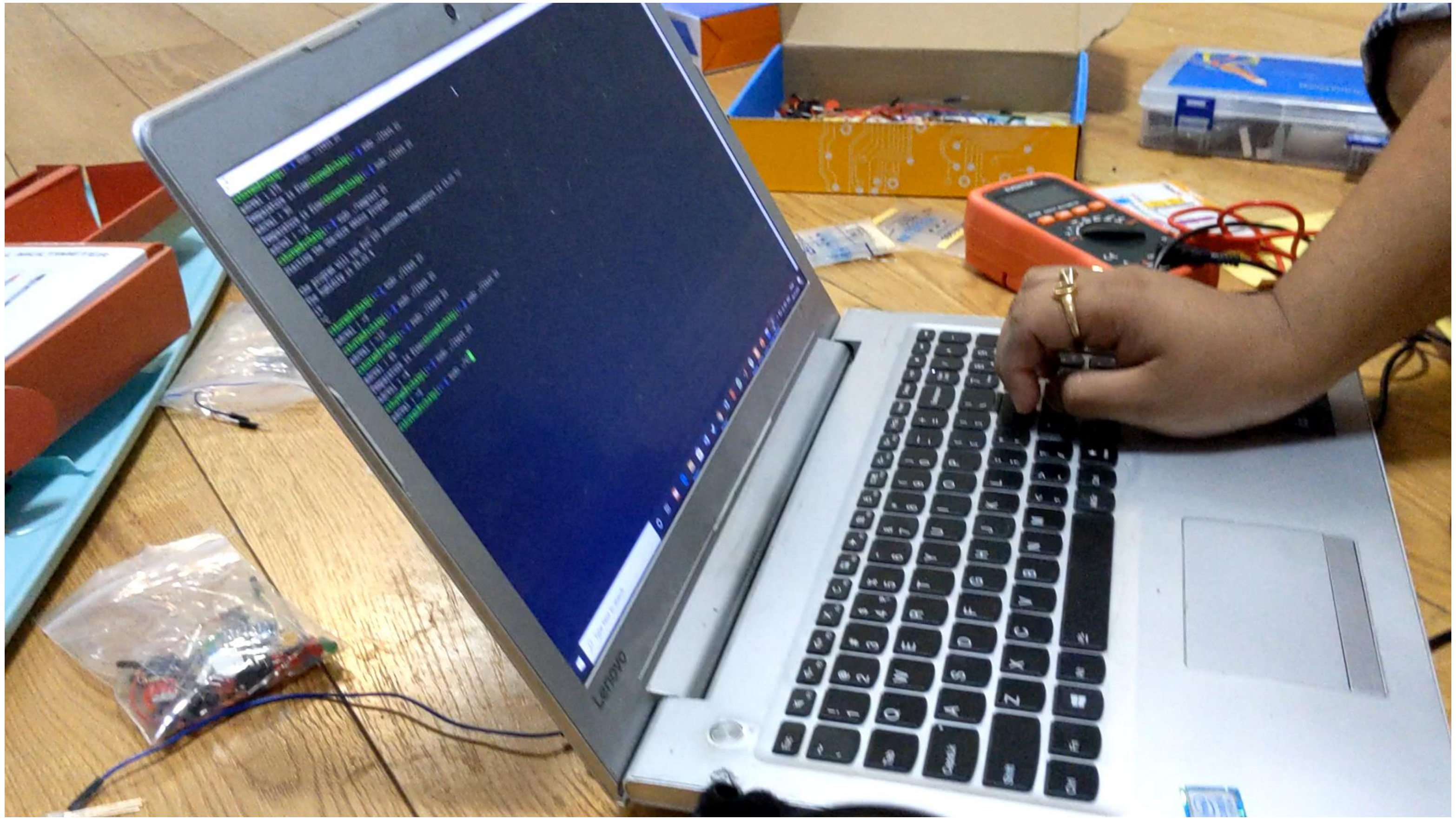
Water level control:

C:\Users\LENOVO\Desktop\Assignment1_CPS\Adeept_Ultimate_Starter_Kit_C_Code_for_RPi-master\14_ultrasonicSensor\Project_waterlevel_control.c - Dev-C++ 5.11

```c

    while(!(digitalRead(Echo) == 0));
    gettimeofday(&tv2, NULL);           //current time

    start = tv1.tv_sec * 1000000 + tv1.tv_usec;
    stop  = tv2.tv_sec * 1000000 + tv2.tv_usec;

    dis = (float)(stop - start) / 1000000 * 34000 / 2;  //count the distance

    return dis;
}

int main(void)
{
    float dis;

    if(wiringPiSetup() == -1){ //when initialize wiring failed,print message to screen
        printf("setup wiringPi failed !\n");
        return -1;
    }

    ultraInit();

    while(1){
        dis = disMeasure();
        printf("Distance = %0.2f cm\n",dis);
        delay(1000);
        if (dis>=14)
        {
        digitalWrite(RelayPin1, 1);
        delay(2000);
        digitalWrite(RelayPin1, 0);}
        elseif(dis<5)
        {digitalWrite(RelayPin2, 1);
        delay(2000);
        digitalWrite(RelayPin2, 0);}
        else
        {printf("Water level is in check");
        }
    }

    return 0;
}
```

Line: 70 Col: 21 Sel: 0 Lines: 82 Length: 1653 Insert Done parsing in 0.344 seconds

Water level control:



Limitations in implementation:

- To monitor the system continually, 12V dry cells are insufficient.
- Ambient light sensor TSL039 required an API. Only photo resistor used.
- Lack of number of relay modules and very high cost of compatible LED bars.
- Only alarm used for Light.
- Manual change in the system is required in case of lack of light.
- Power requirement of solenoid valve needs higher current, so its outwards flow was really slow.

Data acquisition and analysis

- Temporal data from every sensor is analysed for optimum condition prediction.
- Data set of one hour data is created. Which means 24 data sets for one day full operation.
- Every data points in the data set is labelled as whether it that iteration required any actuation because it means variation of weather conditions out of tolerance bound.
- Without loss of generality variance is calculated for the 10 datasets with lowest number of labelled data. The dataset with least variance yields the optimum result.
- Average value of the parameters of the optimal dataset is chosen as the optimum value of parameters.

Data acquisition and analysis

- The optimal values of data obtained in 24 hours can further be extrapolated. If the optimum results are stored with a time stamp and the similar process is iterated for one year of data, it can be used to predict weather data and corresponding optimal parameter values in which plant growth requires minimal user control.

Data acquisition and analysis

- Since the set-up was done in a very cold adversary weather, analysis report does not generate any optimal values of weather parameters that indicate plant growth.
- Using the concept for analysis discussed above, the results we get,
- **Optimal Temperature: 21 degree C.**
- **Optimal Humidity: 62%**
- **Optimal light intensity: 1250 lux**
- **Water Level: 12cm**

Future Scope and Sophistication

- Considering the scope and cost incurred, the current system will be functional in most households
- For further sophistication, a few more sensors can be incorporated to this system namely ph-level sensor and a nutrient sensor
- A ph-level sensor can return real-time ph value of the water used in the system, warning the user to manually change the water if it is too alkaline or too acidic
- A nutrient sensor can measure the nutrient content of the water and can similarly warn the user if the nutrient level is below or above the permissible limit set by the user for the particular plant.
- This system can be used a remotely automated using home AI devices like Alexa.

References

The following literatures and project guides helped us build the idea.

[1] Prabhu, Boselin and Pradeep, M. and Gajendran, E., "An Analysis of Smart Irrigation System Using Wireless Sensor Network" Star Vol.5 Issue 3(3), March (2017)

[2] Ryu M, Yun J, Miao T, Ahn IY, Choi SC, Kim J (2015) "Design and implementation of a connected farm for smart farming system" 2015 IEEE, pp 1–4 (2015).

[3] https://magpi.raspberrypi.org/articles/hydroponic

[4] CPS/Fall19/Lab-slides