Smart Farming   Cannabis Incubator

Feedback Control System

December 22, 2019

Date Performed: November 10, 2019
Members:
Warissara Chawalitsoonthorn
Thanaphat Jiamcharoendet
Petchtechin Petchsiripan
Instructor:
Assoc.Prof.Dr.Pitikhate Sooraksa

Objective

This project was created for planting Cannabis and using the study of feedback control in real application. For this project will solve the problem of planting because it can control and send feedback to user. Also, providing the realtime access via IOT mobile application.

Overview

The objective of this report is to create a smart farming for cannabis which is allowing farmers to have more productive farm than usual. Moreover, it enable farmers to control the weather condition such as, moisture, light, temperature and remote observing. The structure of the report will be as follows:
Chapter 1 will cover over the structure of our prototype model.
Chapter 2 will cover the algorithm and flowchart of the system, including programs in Arduino and IOT enabling technologies, and how IOT application in agriculture. Using feedback control theory to evaluate the accuracy of data.
Chapter 3 will cover the final result of the project and illustrate in graph.

Introduction

Cannabis

Cannabis is one genus of plants in the family Cannabaceae and while the number of species of cannabis within the genus is disputed, there are three basic species: Cannabis sativa, Cannabis indica, Cannabis ruderalis; Some taxonomists include C. ruderalis within C. sativa. Additionally, all three have been classified as subspecies of a single species, C. sativa; or C. sativa may be accepted as a single undivided species. This taxonomic classification is not important except that it can help with understanding the differences within cannabis varieties and how and why their medicinal properties vary. Cannabis strains have been bred to produce varied levels of tetrahydrocannabinol (THC),the principal psychoactive constituent. The strength of THC is enhanced by curing the flowers. Cannabis is also known as hemp, although this term is often used to refer only to varieties cultivated for non-drug use. Cannabis has long been used for hemp fiber, hemp seeds and their oils, hemp leaves for using as vegetables and as juice, medicinal purposes, and as a recreational drug. Our focus in this lesson will be on what the medical literature says about the risks of recreational marijuana use and the medicinal properties of Cannabis. As mentioned earlier, some authorities only recognize one species in the genus Cannabis. This is because human intervention has produced many varieties within the species selecting and breeding plants to either produce plants with more fiber or plants with greater THC content. Ultimately there are five chemotaxonomic (classified based on chemical components) types of Cannabis: one with high levels of THC, one which is more fibrous and has higher levels of CBD, one that is an intermediate between the two, another one with high levels of cannabigerol (CBG), and one almost without cannabinoids.

Importance of Proper Harvest for Cannabinoids

At harvest there are well over 500 phytochemicals present in the cannabis plant. The trichome is the glandular part of the flower of cannabis and contains all these phytochemicals
On the phytocannabinoid chart, CBGA is the mother cannabinoid, the precursor to both the CBD and THC sides of the family tree of cannabinoids. For this reason, harvesting hemp very early yields higher levels of CBGA and CBG compared to later in the plant's life when there are minuscule amounts. Studying the biological and chemical processes in the plant reveal that CBG converts to CBC and then to CBL and then CBT. This also applies to the lineage of CBGA, CBD, CBE, and CBF. These cannabinoids, all trace back to CBGA, which is technically called the olivetol series.
Different enzymatic reactions in the plant create the varinol branch of the family and orcinol branch. The highly sought after and rare cannabinoid, THC- V, doesn't even come from the THC side of the family but from the varinol branch. Of critical importance to hemp growers is closely tracking the increase in THC levels during the last few weeks before harvest. In just a few days during late stage maturation, hemp can go "hot" with the THC level shooting up from low concentrations in high CBD potency hemp varieties to THC levels that exceed the Federal limit of no more than 0.3% to qualify as legal hemp. A crop can

become illegal after a particularly hot day if not immediately harvested and lawfully must be destroyed.
Growing conditions are constantly changing. Daily and nightly temperatures differ during each growth season. The character and amount of light, humidity, timing of water input, soil and nutrients factors, stresses like pests and biologic contaminants, and other conditions are never exactly the same, harvest to harvest, even with indoor grown hemp. Variation is inherent in Mother Nature.

Seed Quality

Quality is very important when it comes to selecting seeds and clones. Genetics plays a big part in the success of the plant. It is important to choose seeds from a reputable seller and breeder but also to choose a variety or strain suited for your desired application. Seeds can be purchased as feminized if you only plan to grow female plants but research shows that feminized seeds have a higher chance of mutating into a hermaphrodite. With that said this is typically a good option instead of wasting money on male seeds because usually in a batch of seeds about half (50%) are male and you can't tell which is which until after germination and well into the growing process.

Seed health is important and shouldn't be overlooked otherwise you could be wasting time trying to use seeds that aren't suitable. The healthy cannabis seed will be a darker brown color with dark stripes. Young and immature seeds will appear light brown to white and green in color. Cracked seeds may still germinate and grow but the stress may cause your plant to grow differently than expected.

Figure 1: Sensor observe and control flowchart

Germination

Germination is the process of bringing a seed out of a dormant stage by providing conditions that encourage the seed to crack open and sprout a taproot,sometimes referred to as "popping." There are a few ways to do this but ensuring the proper environmental temperature and providing adequate oxygen and water are essential. A healthy and mature seed will grow a taproot within a few days to a week. During this stage the seeds should not be exposed to direct light. The ambient temperature of the environment you're germinating in should be around 68-72F (20-22C) but there are hardier strains that can handle both cooler and hotter temperatures. During this early stage the amount of water needed is very little, just enough to dampen the seeds and not saturate them. Saturation can drown your plants before they have a chance to grow. If it doesn't drown them, it will stress them out and overwatering hemp plants early in their life (less than 4 weeks) will stunt their growth and negatively affect yields. A successful method commonly used is to germinate from soil.

Another commonly used method for germination is placing your seeds between a moist paper towel and keeping them in a dark place. Check on the progress daily because you may notice the paper towel drying up and if you do, add a small amount of water. A similar method is simply putting seeds in a cup of water. Oversaturation can drown your seeds so this method should be used with caution. A seed shouldn't soak for more than 24-32 hours. If using the paper towel or cup method, after the seed grows its taproot, you can transfer it to the soil or whatever medium you're planning to grow out of. However you must use extra caution when handling the seed and taproot. At this stage the new growth is very fragile and any damage can be detrimental at this stage. Why would anyone use this method if there's a higher risk involved? Because this way you will know faster if your seeds successfully germinate or if they're duds. These simple methods work well for somewhat small scale indoor grows but when it comes to large scale grows and outdoor operations, using machinery might be more efficient.

Techniques

There are a few techniques to remedying this situation but starting with manipulating the environment to interrupt more growth is key. Optimal temperatures for powdery white mildew growth are between 65-70F (18C-21C). Increasing the temperature of the environment to 86F (30C) or higher will cut spore production in half. Temperatures between 55-60F (12-15C) will slow the growth by a couple of weeks but cooler temperatures should be used with caution around cannabis plants. Air movement and ventilation, keeping the plants dry in places like under its canopy will reduce humidity and spore production. Outdoor grows can experience this problem too. It's recommended to plant seeds or clones with sufficient space in between them in order to decrease potentially damp and shady areas. There is research that fungicides, some oils, and bicarbonate can help in these situations but inexperienced growers should research legal techniques available to them and determine the best method of eliminating the threat for their unique situation.

Chapter1

Incubator Model

Components

a. Foam box 30*30*45

b. ESPino 32 Thaieasyelec microcontroller board

c. Breadboard

d. Relay 5 V 4 channels

e. DHT22 Sensor

f. Humidity maker

g. 4 Fans 5 V

h. Power Supply 5 V 6 A

i. Pump 5 V

j. LCD I2C

k. LED grow light full spectrum

l. RTC 3231 timer

m. Servo motor

n. Ultrasonic sensor

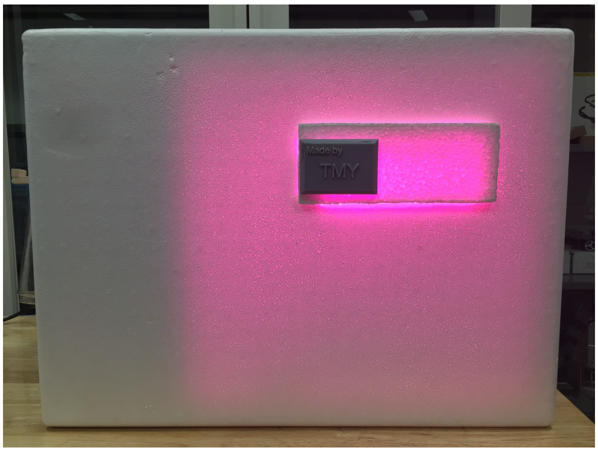
o. Wire and Jumper

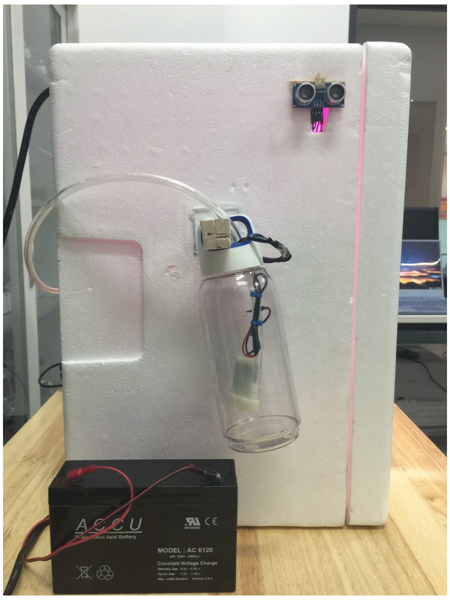
Method and Problems

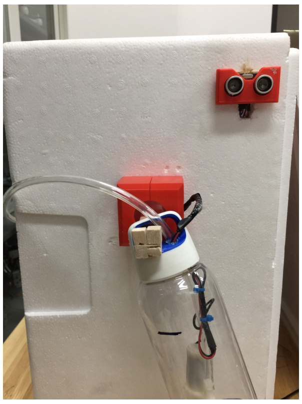
We decided to use foam because it can keep temperature and humidity, also easy to modify with our electronic devices. Importantly, we create the box to have 2 sides, which one is use to be a system box and another one is the planting space. In order to connect a lot of devices together, there are some problems that we need to fix.

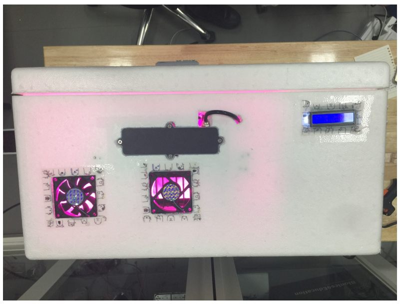
1. The water pump cannot work with LCD screen because the voltage is interrupt the microcontroller and effect to the screen. So, we make a block circuit to avoid this problem, using 4 zener diodes and 1 capacitor block the voltage coming before head to relay.
2. WFI connection is not stable, so it effect to entire system. Because the program will try to access the WIFI before running the command.
3. Relay 5 V cannot sufficiently supply voltage to fans and making the fan speed are very low. The solution is to increase the voltage source to the relay.
4. The voltage from power supply cannot supply for both microcontroller and relay because the signal from relay when it trigs will interrupt work on microcontroller. Therefore, we need to separate the voltage source for both of them.

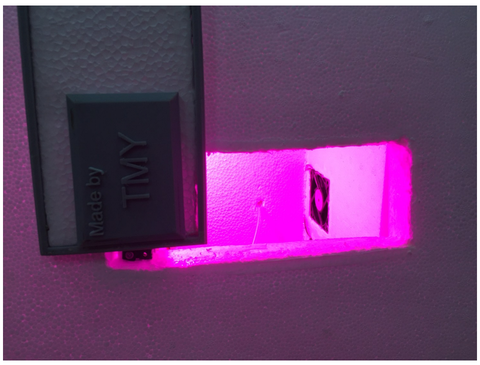
The real model of Cannabis Incubator










Made by
TMY

Chapter2

Flowchart

Temperature and Humidity Control and Observe

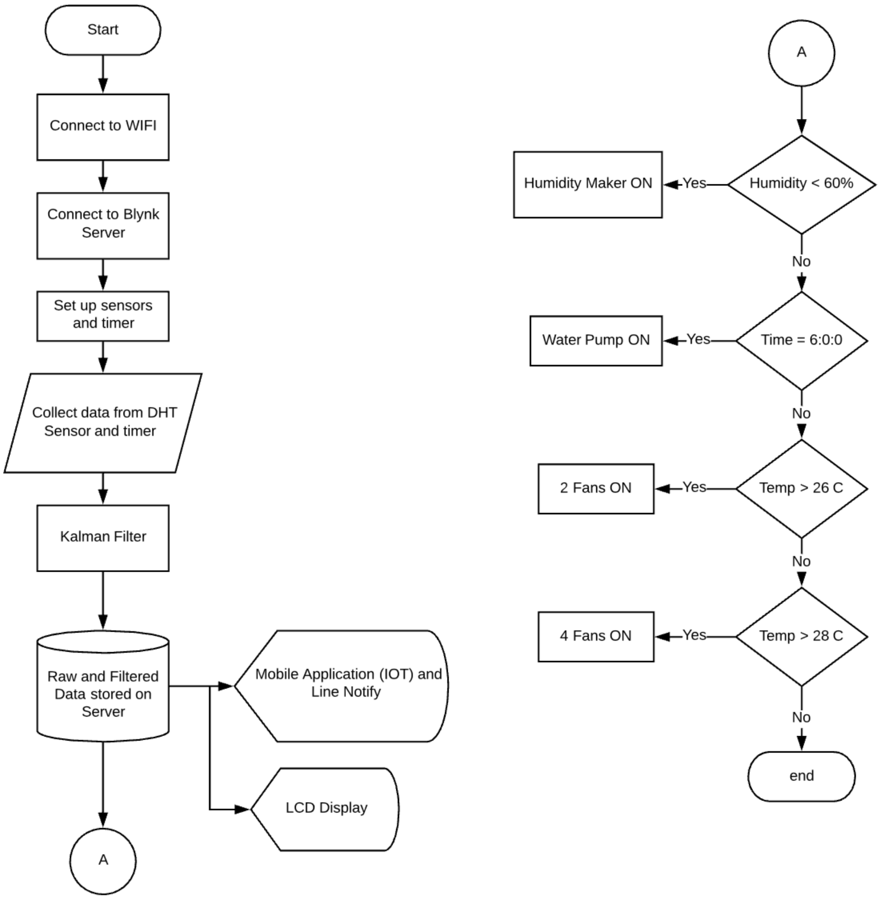


Figure 2: Sensor observe and control flowchart

Automatic Door

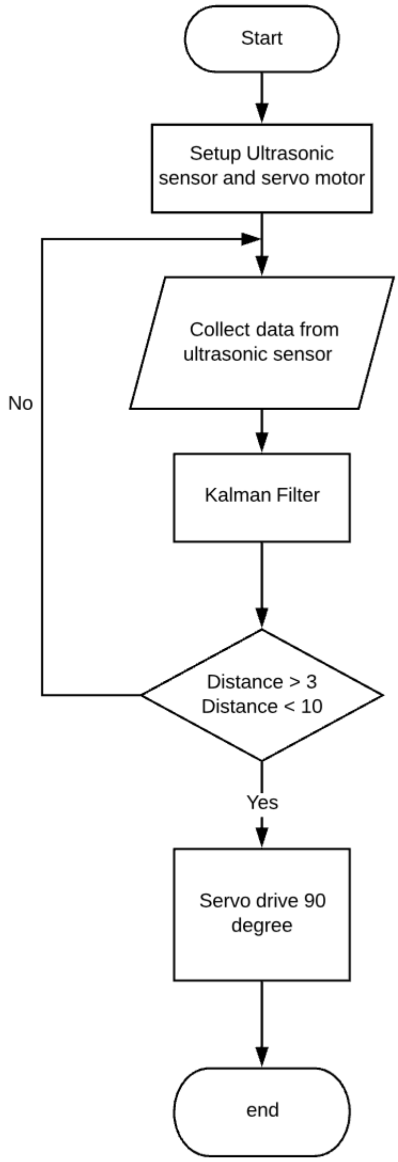


Figure 3: Automatic door algorithm

Software Development

In this project, we use microcontroller ESPino32 from Thaieasyelec and program on Arduino IDE software.

```
#include <LiquidCrystal_I2C.h> // LCD Library
#include <Adafruit_Sensor.h> // LCD Library

#include <DHT.h> // Humidity and Temperature sensor Library
#include <DHT_U.h> // Humidity and Temperature sensor Library

#include "RTClib.h" // timer library
#include <UltrasonicSensor.h>

#include <ESP32Servo.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h> // IOT app database library

#define DHTPIN A18 // RH&T sensor pin
#define DHTTYPE DHT22

#define Emea_Humid 4 // error sensor +-2%
#define Emea_Temp 1 // error sensor +-0.5 celcius
#define kalman_Loop 30

// proper humid range for cannabis
#define humid_Range1 60.00
#define humid_Range2 70.00

// proper temp range for cannabis
#define temp_Range1 26.00
#define temp_Range2 30.00

#define fan1 A15 // fan group1 pin
#define fan2 A17 // fan group2 pin
#define water_Pump A14 // water pump pin
#define RH A16  // humidity maker pin
#define servoPin A10 // servo motor pin
#define BLYNK_PRINT Serial

Servo myservo;
UltrasonicSensor ultrasonic(A12, A11);
```

The main function of system:

```
 void setup() {
  Serial.begin(9600);
  setupRelay();
  setupDHT();
  setupLCD();
  estimate_System();
  setupBlynk();
  setupUltra();
  setupServo();
}

void loop() {
  readData();
  showLCD();
  timer.run();
  // send data from kalman filter and raw data to server
  reset_Value();
  Door();
}
```

Setup all devices that we use in the setup function to begin the system:

```
void readData()
{
  void setupDHT ()
{
  dht.begin();
  delay(500);
}
void setupBlynk()
{
   Blynk.begin(auth, ssid, pass);
   timer.setInterval(2500, Sensor);
}

void setupRelay()
{
  pinMode(water_Pump , OUTPUT);
  delay(750);
  pinMode(RH, OUTPUT);
  delay(750);
  pinMode(fan1, OUTPUT);
  delay(750);
  pinMode(fan2, OUTPUT);
  delay(750);
}

void setupTimer()
{

  #ifndef ESP8266
    while (!Serial); // for Leonardo/Micro/Zero
```

```
  #endif
  delay(3000); // wait for console opening

  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }

  if (rtc.lostPower()) {
    Serial.println("RTC lost power, lets set the time!");
    // following line sets the RTC to the date & time this sketch was
        compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // This line sets the RTC with an explicit date & time, for example
        to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
  }
}

void setupLCD() //step3
{
  lcd.begin();
  delay(500);
  lcd.home();
  lcd.clear();
  delay(500);
}
void setupServo()
{
  myservo.setPeriodHertz(180); // standard 50 hz servo
  myservo.attach(servoPin, 1000, 2000);
}
void setupUltra()
{
  int temperature = 22;
  ultrasonic.setTemperature(temperature);
}
```

Read data function is the function to collect data from sensors and timer to use in further application:

```
void readData()
{
  now = rtc.now();
  humidity = dht.readHumidity();
  temp = dht.readTemperature();
}
```

Kalman Filter

Using Kalman Filter to estimate the data. Kalman Filter is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe. Therefore, the Kalman Gain is calculated, along with the observed data. The update process involves using the Kalman in conjuction with the previous estimate and new observed data to update the state variable towards a belief that's somewhere between the prediction and measurement. The process covariance is also updated based on the Kalman gain. These updates are then used for the next round of predictions.
The Kalman gain in one dimensional system

$$K = \frac{E_{est}}{E_{est} + E_{mea}}$$

The number for the Kalman gain will be somewhere 0 and 1. It is then used to update the value of the current estimate. In the equation below, x represents the estimate, K is the Kalman gain, which is multiplied (acting like a weight) by the difference between the measurement (p) and the previous estimate.

$$x_t = x_{t-1} + K[p - x_{t-1}]$$

If the Kalman Gain is close to 1, it means the measurements are accurate but the estimates are unstable. We can infer this by looking at the Kalman Gain ratio, where if the error of the measurement is small (and practically contributing nothing to E est over E st), then K will be equal to one. When the error of the measurement is small, future predictions will be strongly updated based on new input data.

However, if the Kalman Gain is small, then it the error in the estimate is large relative to the error in the estimate. Estimates are more stable and the measurements are inaccurate. As a result, any difference between new data and the prediction will have a smaller effect on the eventual update.

$$Est_t = Est_{t-1} + K[M - Est_{t-1}]$$

$$E_t = [1 - K]E_{t-1}$$

E is the error of the estimate and the Kalman Gain is multiplied by the previous estimate. The one minus K factor being multiplied with the previous error is the inverse of the size of the Kalman Gain.

The function of calculating the estimate value from previous weather condition within 200 seconds:

```
void estimate_System(){ // estimate temp & Humidity to use in kalman
    filter
  int count = 0;
  lcd.print("estimating");


  for(int i = 0 ; i < kalman_Loop ; i++ )
  {
    readData();
    est_Temp0 += temp; //estimate
    est_Humid0 += humidity;
    delay(500);

    if((i%10) == 0)
    {
      lcd.print(".");
      if(count == 3)
      {
        lcd.setCursor(11,0);
        lcd.print("  ");
        lcd.setCursor(11,0);
        count = 0;
      }
      count++;
    }
  }
  est_Temp0 /= kalman_Loop;
  est_Humid0 /= kalman_Loop; // Temp&Humid initial
  lcd.clear();
  delay(500);
  lcd.print("ready");
  delay(500);
  lcd.clear();
}
```

After calculated the estimate value put them into Kalman function to begin the estimating:

```
  void kalman()
{
    mea_Temp = temp;
    KGT = ( Eest_Temp ) / ( Eest_Temp + Emea_Temp );
    est_Temp = ( est_Temp0 ) + (KGT * ( mea_Temp - est_Temp0 ));
    Eest_Temp = ( Eest_Temp0 * (1 - KGT));

    mea_Humid = humidity;
    KGH = ( Eest_Humid ) / ( Eest_Humid + Emea_Humid );
    est_Humid =( est_Humid0 ) + (KGH * ( mea_Humid - est_Humid0 ));
    Eest_Humid = ( Eest_Humid0 * (1 - KGH));

}

void reset_Value() //reset data from kalman filter and send to next loop
     of kalman
{
  est_Temp0 = est_Temp;
  est_Humid0 = est_Humid;
  Eest_Temp0 = Eest_Temp;
  Eest_Humid0 = Eest_Humid;
}
```

showLCD function will call Kalman Filter function to estimate the nearest true value from sensors:

```
void showLCD()
{
  kalman();
  if ( est_Temp > humid_Range2 || est_Humid < humid_Range1 )
  {
    lcd.setCursor(3,0);
    lcd.write(0); lcd.createChar(0, customChar_clock);
    lcd.setCursor(4,0);
    lcd.print(now.hour(), DEC); lcd.print(":"); lcd.print(now.minute(),
        DEC); lcd.print(":"); lcd.print(now.second(), DEC); lcd.print("
        ");
    lcd.setCursor(0,1);
    lcd.write(1); lcd.createChar(1, customChar_T);
    lcd.setCursor(1,1); lcd.print(est_Temp);
    lcd.write(2); lcd.createChar(2, customChar_degree);
        lcd.setCursor(7,1); lcd.print("C ");
    lcd.write(3); lcd.createChar(3, customChar_H_bad);
        lcd.setCursor(10,1); lcd.print(est_Humid); lcd.print("%");
  }
  else
  {
    lcd.setCursor(3,0);
    lcd.write(0); lcd.createChar(0, customChar_clock);
    lcd.setCursor(4,0);
    lcd.print(now.hour(), DEC); lcd.print(":"); lcd.print(now.minute(),
```

```
        DEC); lcd.print(":"); lcd.print(now.second(), DEC);
    lcd.setCursor(0,1);
    lcd.write(1); lcd.createChar(1, customChar_T);
    lcd.setCursor(1,1); lcd.print(est_Temp);
    lcd.write(2); lcd.createChar(2, customChar_degree);
        lcd.setCursor(7,1); lcd.print("C ");
    lcd.write(3); lcd.createChar(3, customChar_H_good);
        lcd.setCursor(10,1); lcd.print(est_Humid); lcd.print("%");
  }
  enable_Fan();
  enable_Humidifier();
  delay(1000);
}
```

Sensor function is the function that send signals to Blynk application on smart phone and show the realtime data including graph plotting of RAW data and Kalman Filtered data.

```
void Sensor()
{

  Blynk.virtualWrite(V1, est_Humid);
  Blynk.virtualWrite(V2, est_Temp);
  Blynk.virtualWrite(V3, humidity);
  Blynk.virtualWrite(V4, temp);
  Blynk.virtualWrite(V5, Eest_Temp);
  Blynk.virtualWrite(V6, Eest_Humid);
}
```

This function is create to control the door opening when users show the hand on the left side of smart farming box the front door will open for small observing automatically.

```
void Door()
{
  distance = ultrasonic.distanceInCentimeters();
  delay(500);
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  if (distance >=3 && distance <= 10)
  {
   myservo.write(180);
    delay(10000);
  }
  else
  {
    myservo.write(15);
    delay(10000);
  }
}
```

Full code you can find here

Chapter3

Project Result and Conclusion

The final result based on the data using Kalman filter, the noises are absolutely reduced as the graph shows the relative humidity data in each time interval. The red line is data from Kalman Filter value and black line is RAW data from direct sensors.

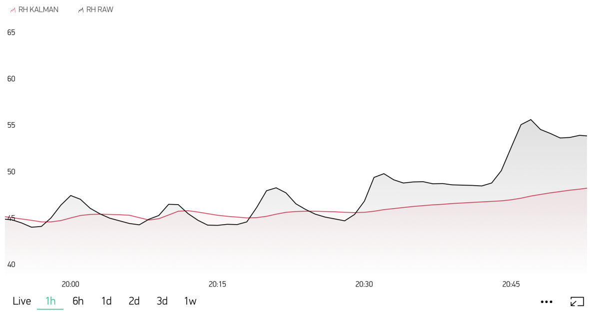


Figure 4: The result in 1 hour of opening system

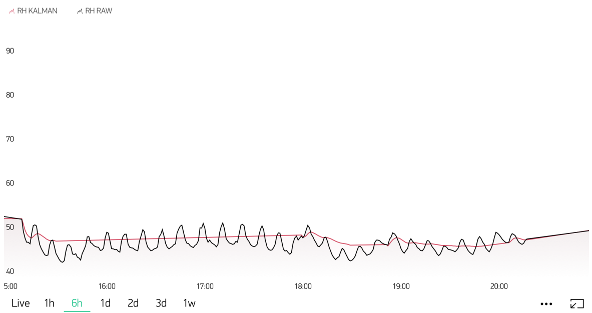


Figure 5: The result in 6 hour of opening system

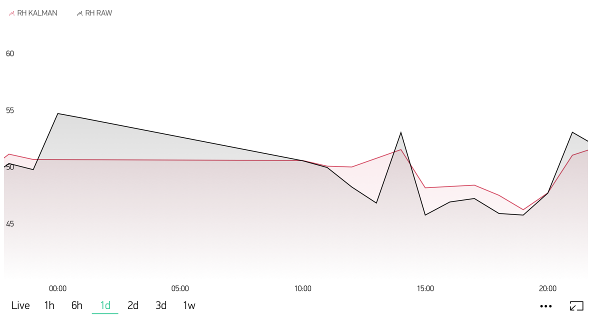


Figure 6: The result in 1 day of opening system

Cannabis growth while plant in cannabis incubator

Figure 7: Day1 in incubator

Figure 8: Day2 in incubator

Figure 9: Day3 in incubator

Figure 10: Day5 in incubator

Figure 11: Day10 in incubator

Conclusion

In conclusion, after observed the growth of Cannabis, the comparison between the 2 cannabis that live in our incubator and outdoor environment after 14 days of planting is very interesting. The cannabis tree in outdoor environment have a lean trunk because of the wind and light is not stable, including the leaves wither due to the insects. As you can see that living in incubator will avoid the unwanted flora, aphid, mold or pests. Moreover, we can set the proper weather condition to control the environment for each phase of Cannabis.

Mobile application IOT

We also develop the mobile application on Blynk platform, which provided free server to store the data. You may download our application for test here or scan this QRcode.

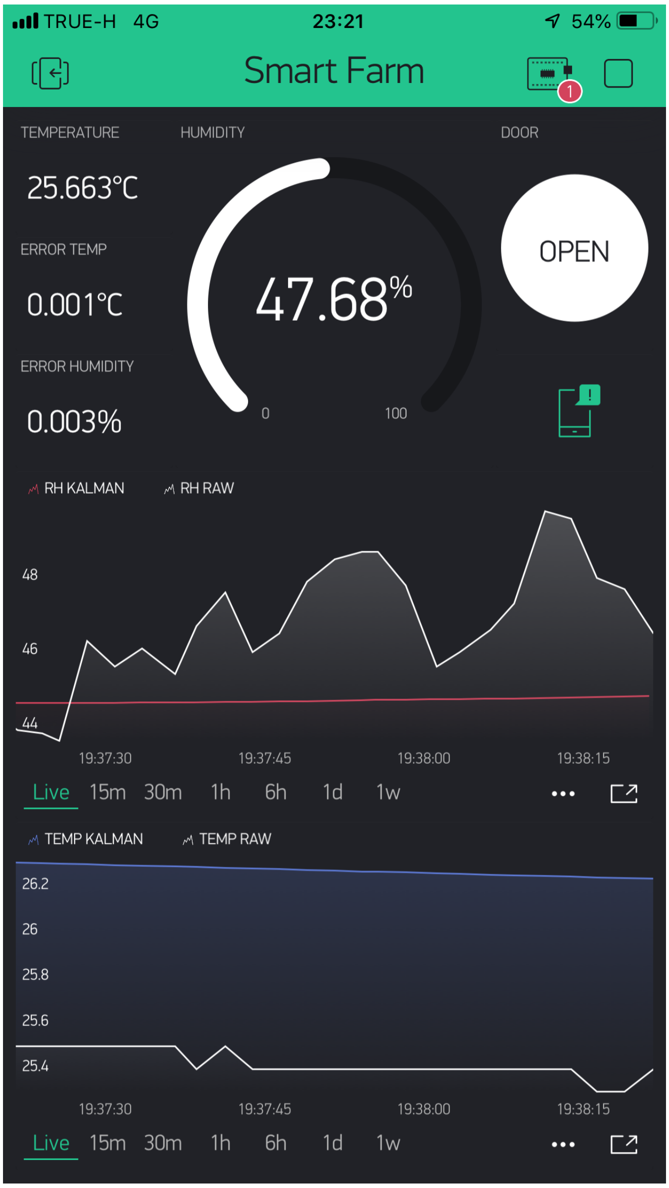


Figure 12: IOT software

Figure 13: QR code login to Blynk application

References

[1] Wikipedia. (2019) Kalman Filter. Retrieved from https://en.wikipedia.org/wiki/Kalman-filter.

[2] James Teow (2018) Understanding Kalman Filters with Python. Retrieved from https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48.

[3] Donae University. Cannabis. Retrieved from https://www.doane.edu/news/doane-to-offer-professional-cannabis-certificate-program.