ASSUNPTION UNIVERSITY
FACULTY OF ENGINEERING

BASIC SCIENCE DEPARTMENT

EE3705 MICROPROCESSORS AND MICROCONTROLLER PROJECT

Automatic Watering System

Submitted to: A. Sunphong Thanok

Submitted by: Mr. Senee Suksawat ID: 6235107 Section: 641.
Mr. Nathapop Khankaew ID: 6310297 Section: 641.

Assumption University Thailand

Semester: 3/2022

Contents

Introduction

- Plant care and watering are important and cannot be ignored. At present, farmers are tired of reducing water in large quantities and cannot do it thoroughly and consistently. Therefore, it is our group's duty to develop a system to make farmers work easier. This system is designed to help farmers track the required fertilizer, watering, and humidity status in their respective digital farms with the help of the Blynk app.

Operations and functions

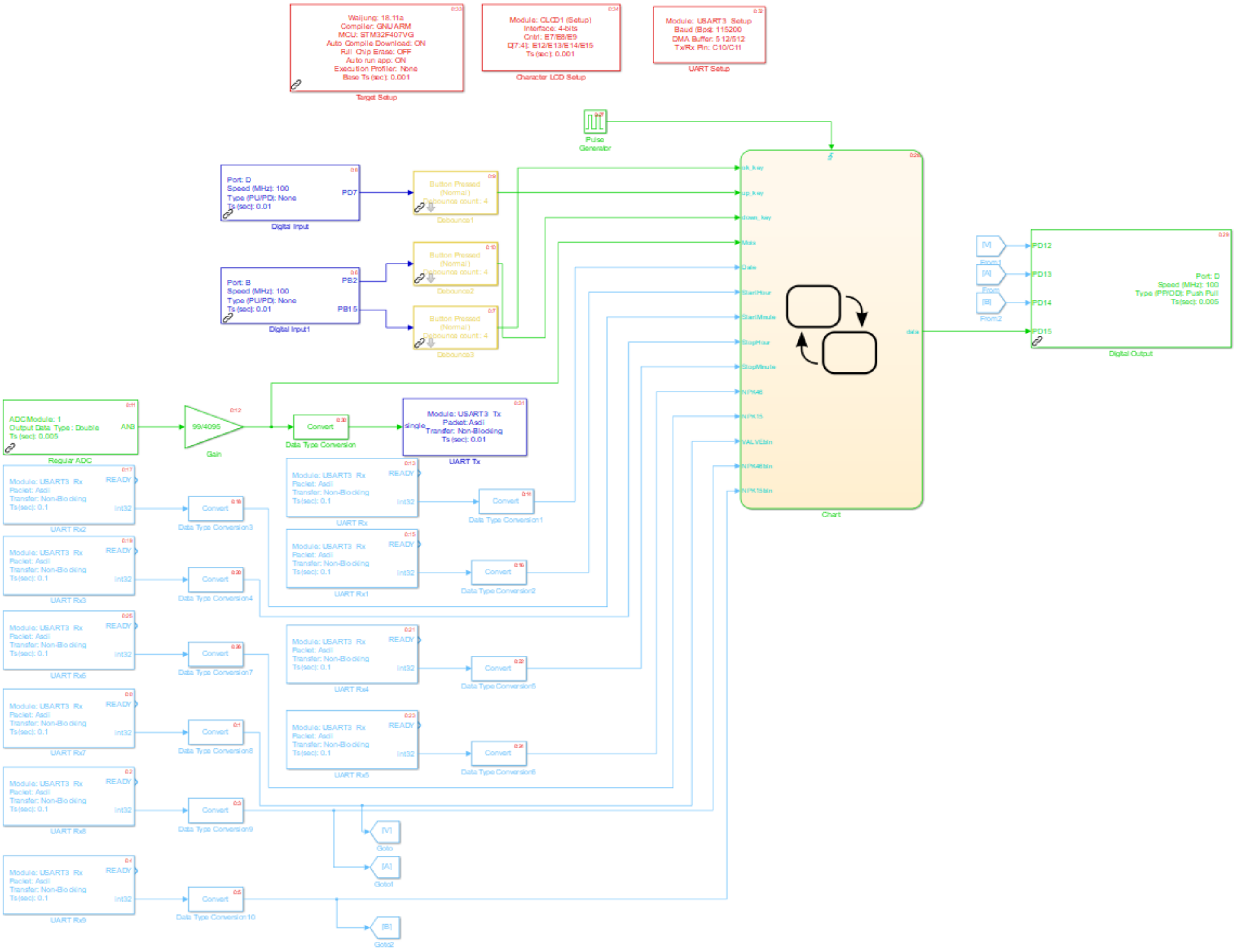
- STM32

- It is used to view the status from the Blynk application
- There are two is automatic and manual.
- It is used to view status from the Blynk app
- Show the watering date and time from Blynk app
- On the manual control status, it will show Valve being addressed to as 1 to represent that the valve will be open, and water will flow as much as the farmer wants

- Blynk

- Used for setting the duration of watering
- This project is used to display fertilizer status
- This project is used to display the humidity status

MATLAB Source code



The STM32F4 is used to display the data received from Blynk and this MATLAB module consists of three important components:

- LCD

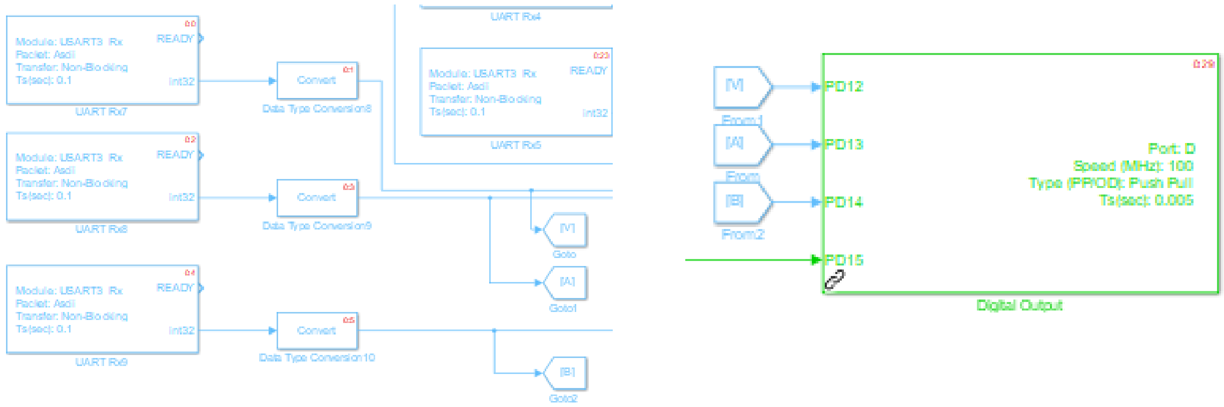
- Chart

- UART

Three components

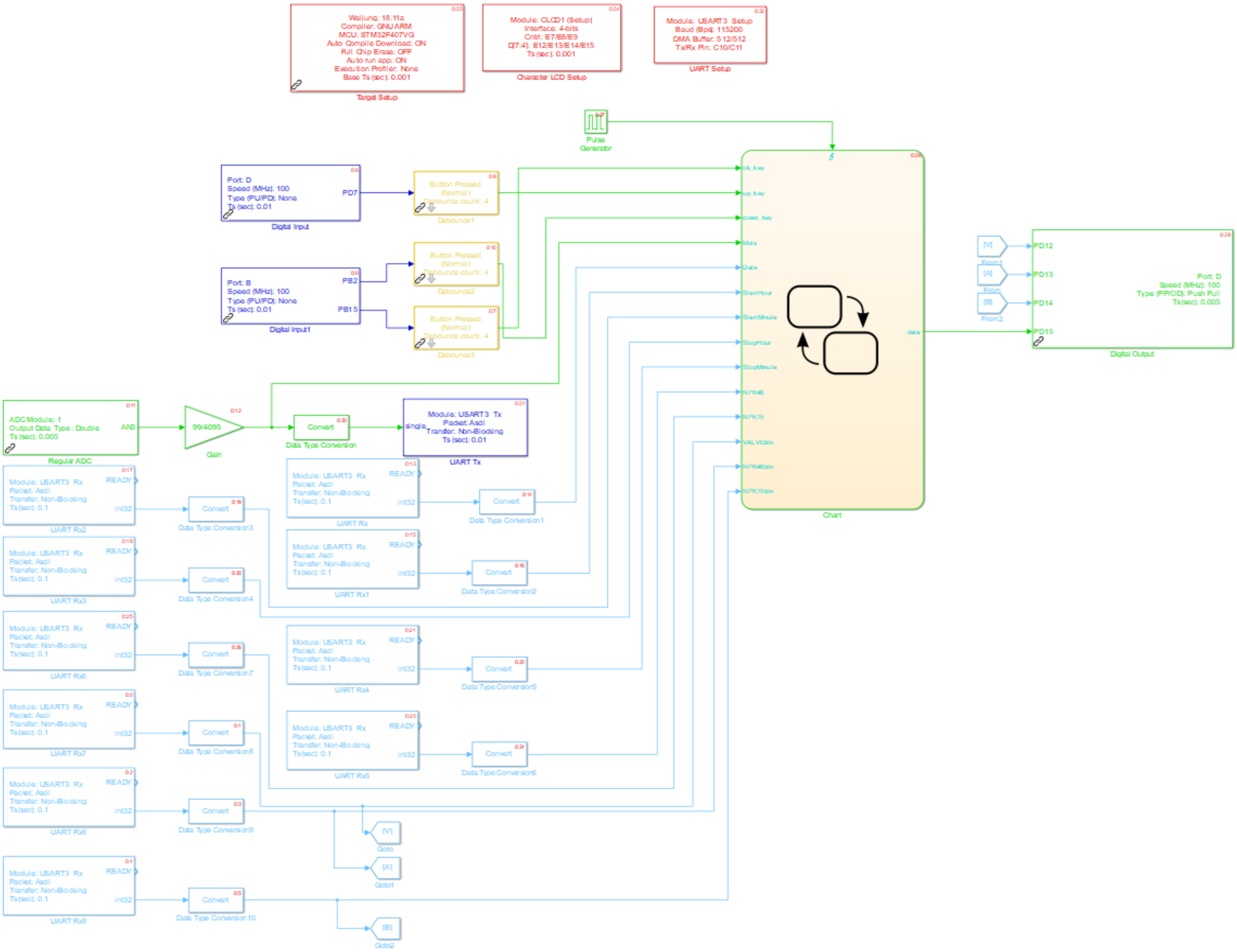
1. LED

To display texts on the STM32F4's screen, a Character LCD setup block is needed.

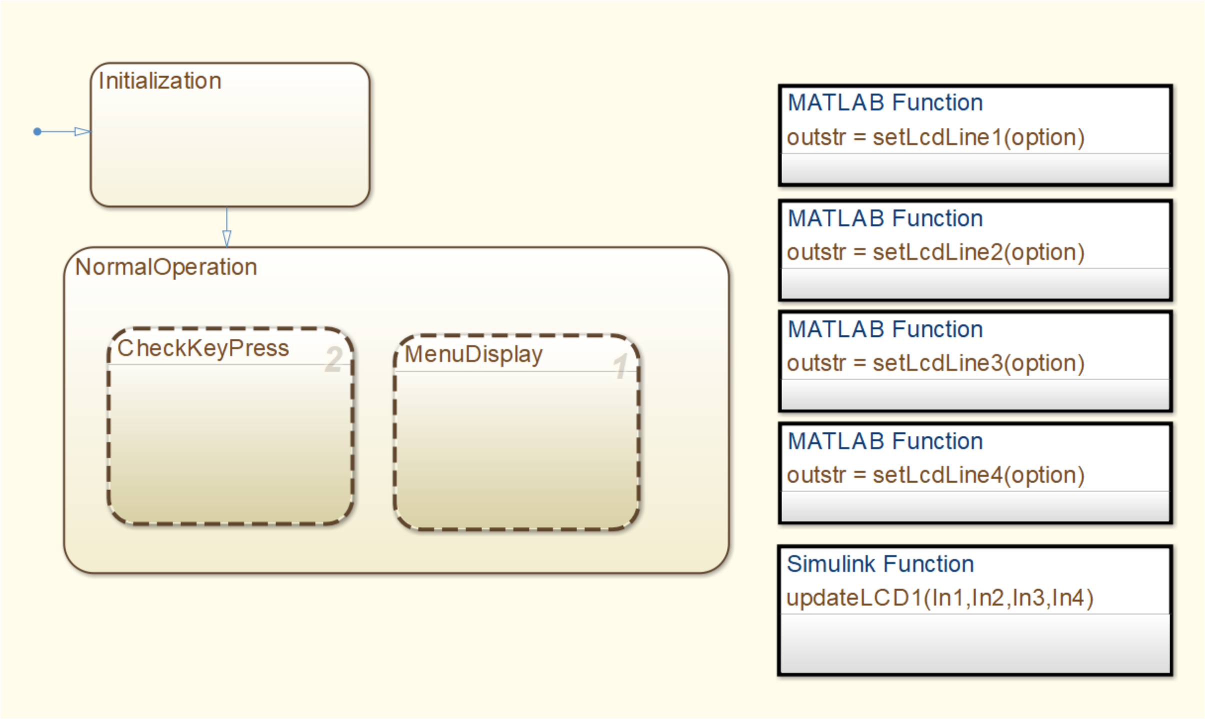


2. Chart

To create a menu design displayed on the STM32's screen, chart and state flow are needed

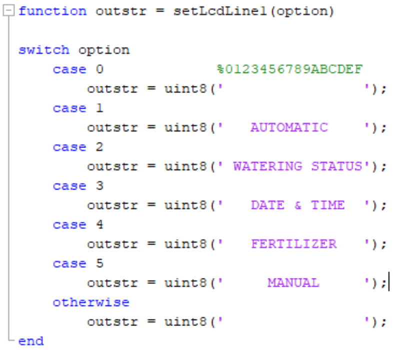


- Components inside chart


Initialization
NormalOperation
CheckKeyPress
2
MenuDisplay
1
MATLAB Function
outstr = setLcdLine1(option)
MATLAB Function
outstr = setLcdLine2(option)
MATLAB Function
outstr = setLcdLine3(option)
MATLAB Function
outstr = setLcdLine4(option)
Simulink Function
updateLCD1(In1,In2,In3,In4)

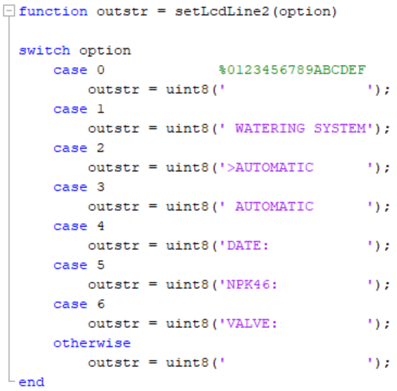
- MATLAB Functions

outstr = setLcdLine1(option)



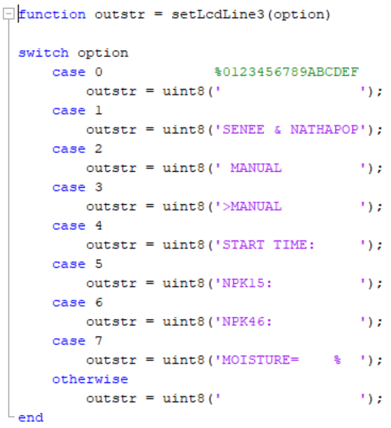
```
function outstr = setLcdLine1(option)

switch option
    case 0             %0123456789ABCDEF
        outstr = uint8('                ');
    case 1
        outstr = uint8('   AUTOMATIC    ');
    case 2
        outstr = uint8(' WATERING STATUS');
    case 3
        outstr = uint8('   DATE & TIME  ');
    case 4
        outstr = uint8('   FERTILIZER   ');
    case 5
        outstr = uint8('     MANUAL     ');
    otherwise
        outstr = uint8('                ');
end
```

outstr = setLcdLine2(option)



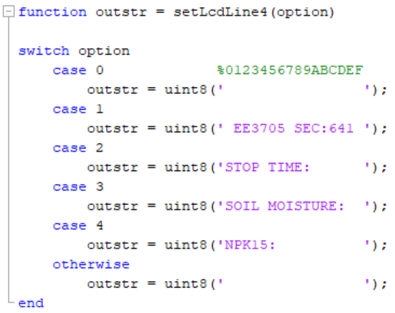
```
function outstr = setLcdLine2(option)

switch option
    case 0             %0123456789ABCDEF
        outstr = uint8('                ');
    case 1
        outstr = uint8(' WATERING SYSTEM');
    case 2
        outstr = uint8('>AUTOMATIC      ');
    case 3
        outstr = uint8(' AUTOMATIC      ');
    case 4
        outstr = uint8('DATE:           ');
    case 5
        outstr = uint8('NPK46:          ');
    case 6
        outstr = uint8('VALVE:          ');
    otherwise
        outstr = uint8('                ');
end
```

outstr = setLcdLine3(option)



```
function outstr = setLcdLine3(option)

switch option
    case 0             %0123456789ABCDEF
        outstr = uint8('                ');
    case 1
        outstr = uint8('SENEE & NATHAPOP');
    case 2
        outstr = uint8(' MANUAL         ');
    case 3
        outstr = uint8('>MANUAL         ');
    case 4
        outstr = uint8('START TIME:     ');
    case 5
        outstr = uint8('NPK15:          ');
    case 6
        outstr = uint8('NPK46:          ');
    case 7
        outstr = uint8('MOISTURE=    %  ');
    otherwise
        outstr = uint8('                ');
end
```

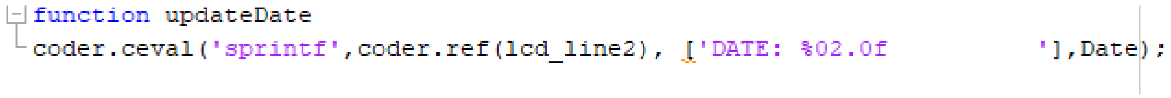
outstr = setLcdLine4(option)



```
function outstr = setLcdLine4(option)

switch option
    case 0             %0123456789ABCDEF
        outstr = uint8('                ');
    case 1
        outstr = uint8(' EE3705 SEC:641 ');
    case 2
        outstr = uint8('STOP TIME:      ');
    case 3
        outstr = uint8('SOIL MOISTURE:  ');
    case 4
        outstr = uint8('NPK15:          ');
    otherwise
        outstr = uint8('                ');
end
```

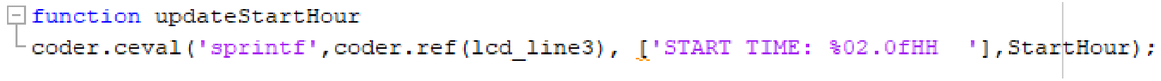
LCD is a function to display text on a specific line. For lines where data can be changed with the Blynk app, we use the function that is the updated version

-MATLAB Function updateDate



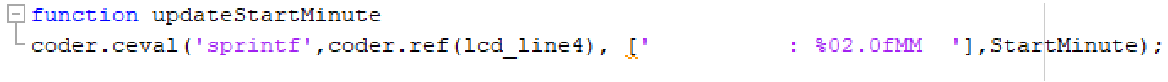
```
function updateDate
coder.ceval('sprintf',coder.ref(lcd_line2), ['DATE: %02.0f          '],Date);
```

-MATLAB Function updateStartHour



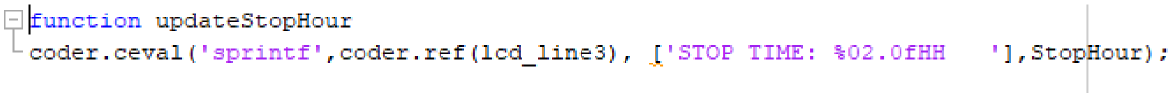
```
function updateStartHour
coder.ceval('sprintf',coder.ref(lcd_line3), ['START TIME: %02.0fHH  '],StartHour);
```

-MATLAB Function updateStartMinute



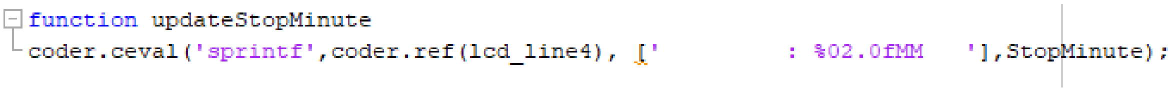
```
function updateStartMinute
coder.ceval('sprintf',coder.ref(lcd_line4), ['         : %02.0fMM  '],StartMinute);
```

-MATLAB Function updateStopHour



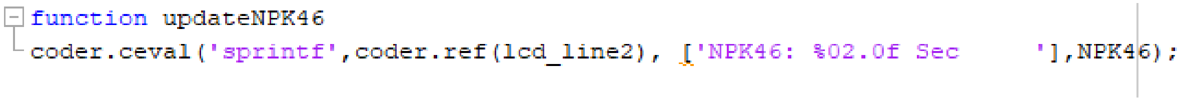
```
function updateStopHour
coder.ceval('sprintf',coder.ref(lcd_line3), ['STOP TIME: %02.0fHH    '],StopHour);
```

-MATLAB Function updateStopMinute



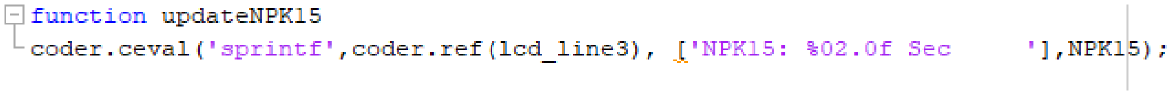
```
function updateStopMinute
coder.ceval('sprintf',coder.ref(lcd_line4), ['          : %02.0fMM   '],StopMinute);
```

-MATLAB Function updateNPK46



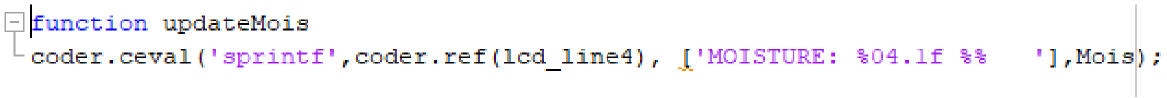
```
function updateNPK46
coder.ceval('sprintf',coder.ref(lcd_line2), ['NPK46: %02.0f Sec      '],NPK46);
```

-MATLAB Function updateNPK15



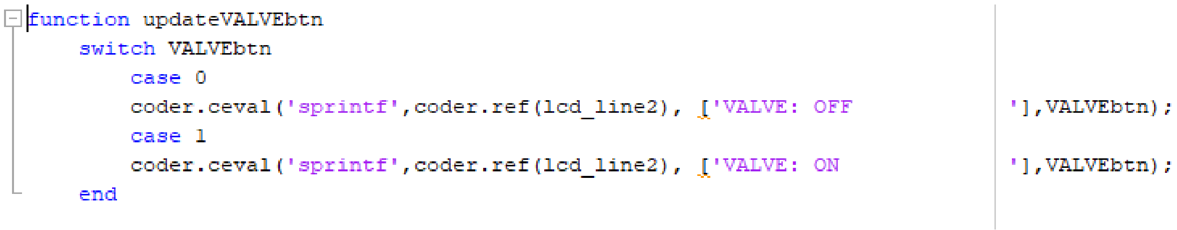
```
function updateNPK15
coder.ceval('sprintf',coder.ref(lcd_line3), ['NPK15: %02.0f Sec      '],NPK15);
```

-MATLAB Function updateMois



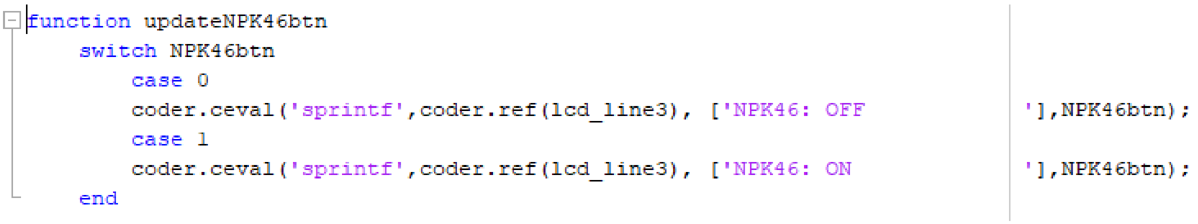
```
function updateMois
coder.ceval('sprintf',coder.ref(lcd_line4), ['MOISTURE: %04.1f %%    '],Mois);
```

-MATLAB Function updateVALVEbtn



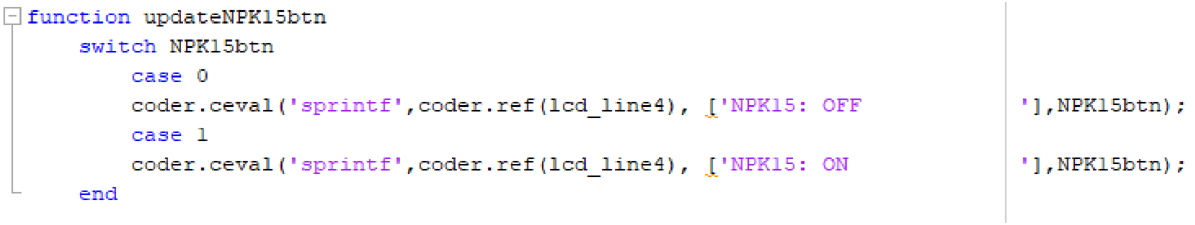
```
function updateVALVEbtn
    switch VALVEbtn
        case 0
        coder.ceval('sprintf',coder.ref(lcd_line2), ['VALVE: OFF             '],VALVEbtn);
        case 1
        coder.ceval('sprintf',coder.ref(lcd_line2), ['VALVE: ON              '],VALVEbtn);
    end
```

-MATLAB Function update NPK46btn



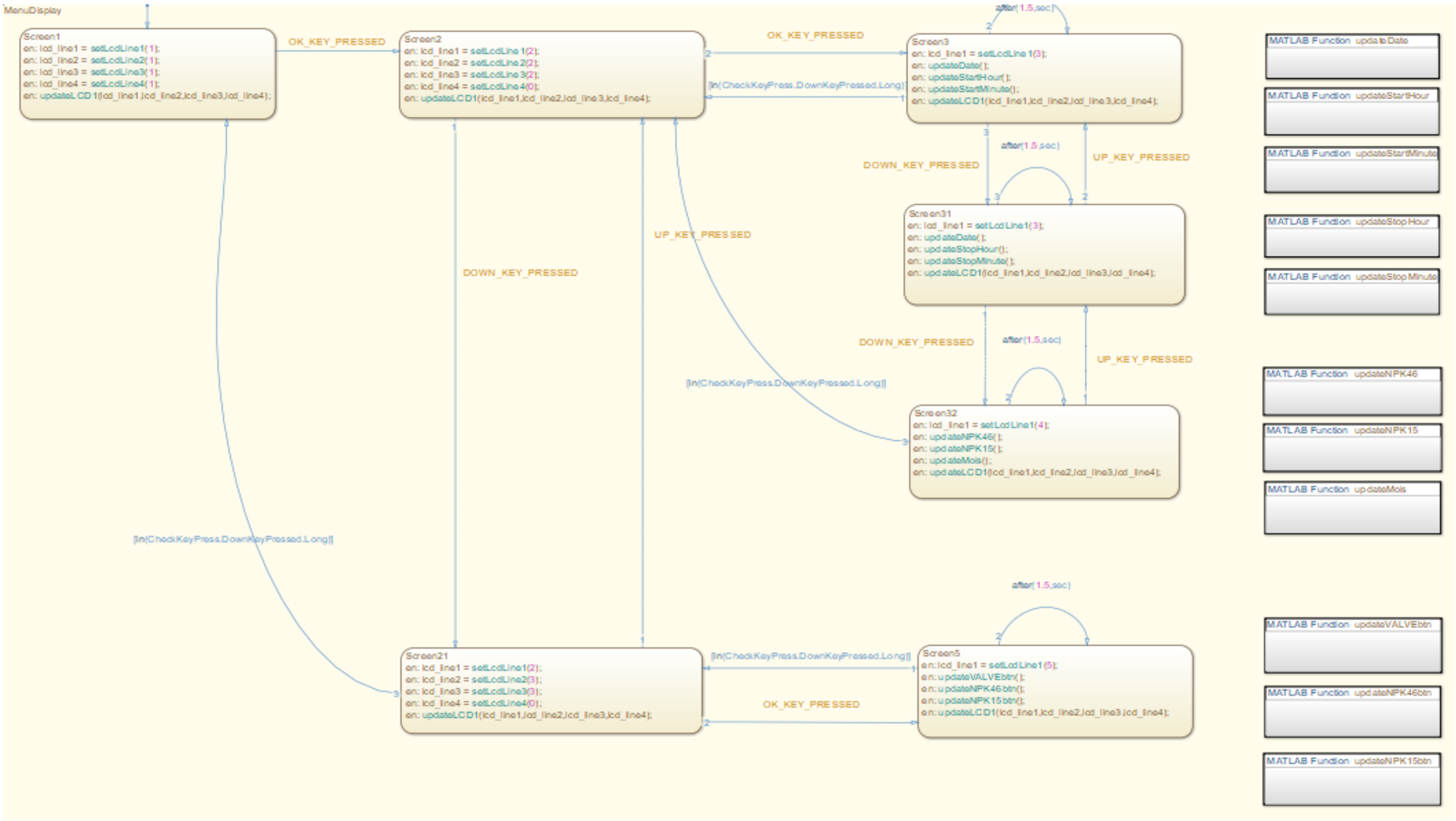
```
function updateNPK46btn
    switch NPK46btn
        case 0
        coder.ceval('sprintf',coder.ref(lcd_line3), ['NPK46: OFF             '],NPK46btn);
        case 1
        coder.ceval('sprintf',coder.ref(lcd_line3), ['NPK46: ON              '],NPK46btn);
    end
```

-MATLAB Function update NPK15Ebtn



```
function updateNPK15btn
    switch NPK15btn
        case 0
        coder.ceval('sprintf',coder.ref(lcd_line4), ['NPK15: OFF             '],NPK15btn);
        case 1
        coder.ceval('sprintf',coder.ref(lcd_line4), ['NPK15: ON              '],NPK15btn);
    end
```

Menu Display



```
Screen1
en: lcd_line1 = setLcdLine1(1);
en: lcd_line2 = setLcdLine2(1);
en: lcd_line3 = setLcdLine3(1);
en: lcd_line4 = setLcdLine4(1);
en: updateLCD1(lcd_line1,lcd_line2,lcd_line3,lcd_line4);
```

```
Screen2
en: lcd_line1 = setLcdLine1(2);
en: lcd_line2 = setLcdLine2(2);
en: lcd_line3 = setLcdLine3(2);
en: lcd_line4 = setLcdLine4(0);
en: updateLCD1(lcd_line1,lcd_line2,lcd_line3,lcd_line4);
```

```
Screen3
en: lcd_line1 = setLcdLine1(3);
en: updateDate();
en: updateStartHour();
en: updateStartMinute();
en: updateLCD1(lcd_line1,lcd_line2,lcd_line3,lcd_line4);
```

```
Screen31
en: lcd_line1 = setLcdLine1(3);
en: updateDate();
en: updateStopHour();
en: updateStopMinute();
en: updateLCD1(lcd_line1,lcd_line2,lcd_line3,lcd_line4);
```

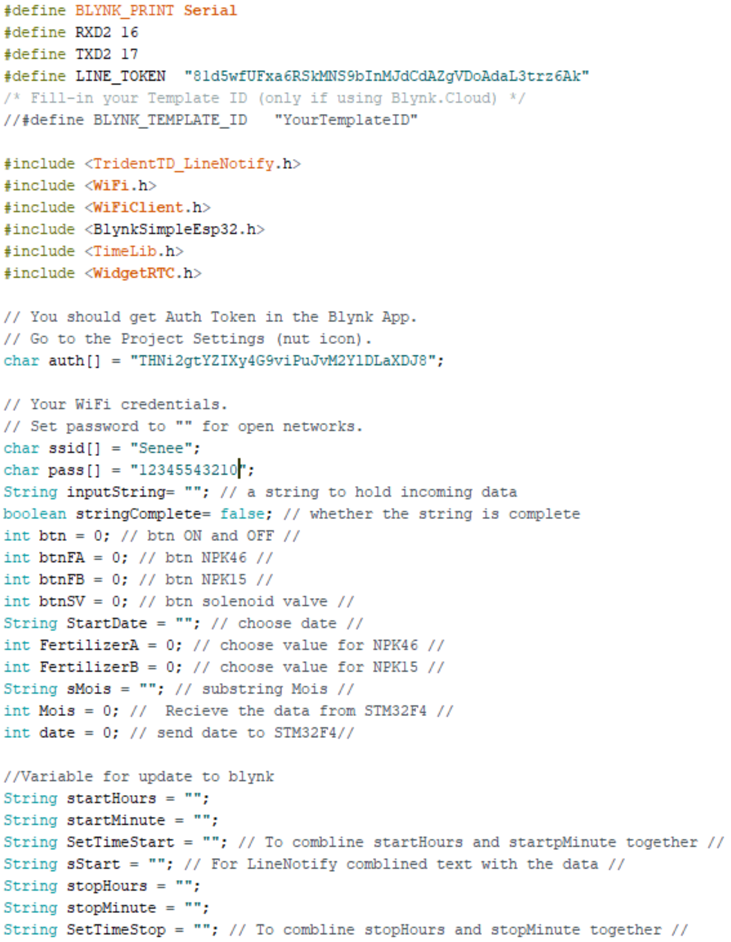
```
Screen32
en: lcd_line1 = setLcdLine1(4);
en: updateNPK46();
en: updateNPK15();
en: updateMois();
en: updateLCD1(lcd_line1,lcd_line2,lcd_line3,lcd_line4);
```

```
Screen21
en: lcd_line1 = setLcdLine1(2);
en: lcd_line2 = setLcdLine2(3);
en: lcd_line3 = setLcdLine3(3);
en: lcd_line4 = setLcdLine4(0);
en: updateLCD1(lcd_line1,lcd_line2,lcd_line3,lcd_line4);
```

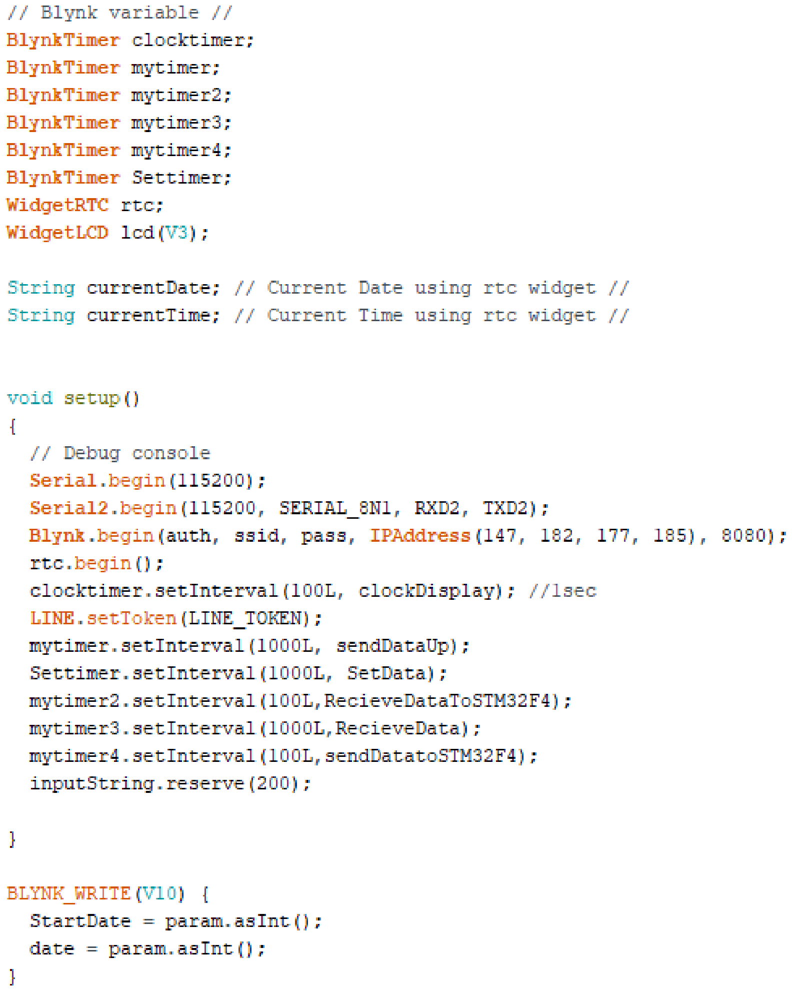
```
Screen5
en:lcd_line1 = setLcdLine1(5);
en:updateVALVEbtn();
en:updateNPK46btn();
en:updateNPK15btn();
en:updateLCD1(lcd_line1,lcd_line2,lcd_line3,lcd_line4);
```

MenuDisplay
OK_KEY_PRESSED
OK_KEY_PRESSED
after(1.5,sec)
[in(CheckKeyPress.DownKeyPressed.Long)]
after(1.5,sec)
DOWN_KEY_PRESSED
UP_KEY_PRESSED
UP_KEY_PRESSED
DOWN_KEY_PRESSED
DOWN_KEY_PRESSED
after(1.5,sec)
UP_KEY_PRESSED
[in(CheckKeyPress.DownKeyPressed.Long)]
[in(CheckKeyPress.DownKeyPressed.Long)]
after(1.5,sec)
[in(CheckKeyPress.DownKeyPressed.Long)]
OK_KEY_PRESSED
MATLAB Function updateDate
MATLAB Function updateStartHour
MATLAB Function updateStartMinute
MATLAB Function updateStopHour
MATLAB Function updateStopMinute
MATLAB Function updateNPK46
MATLAB Function updateNPK15
MATLAB Function updateMois
MATLAB Function updateVALVEbtn
MATLAB Function updateNPK46btn
MATLAB Function updateNPK15btn

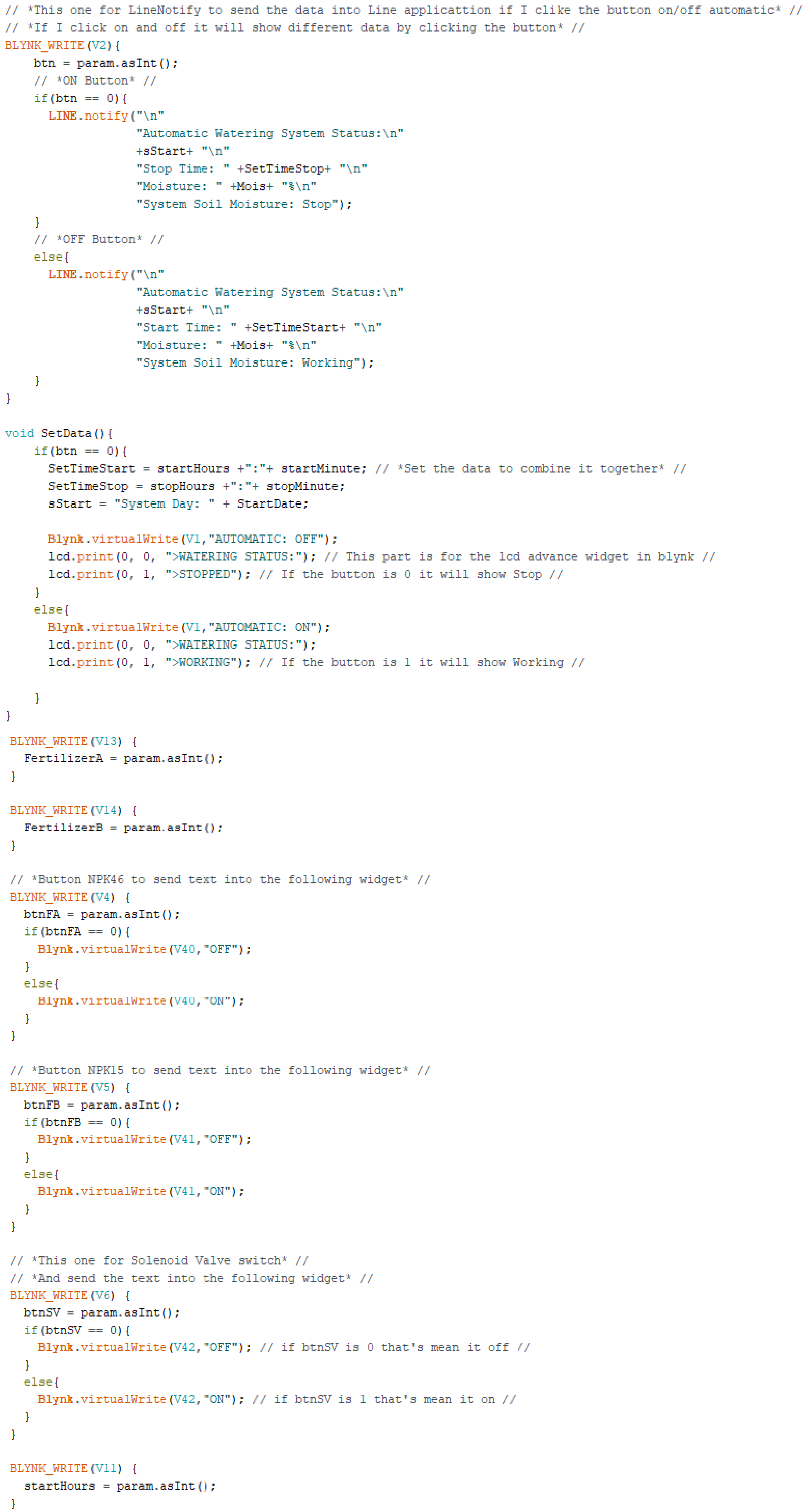
Arduino Code



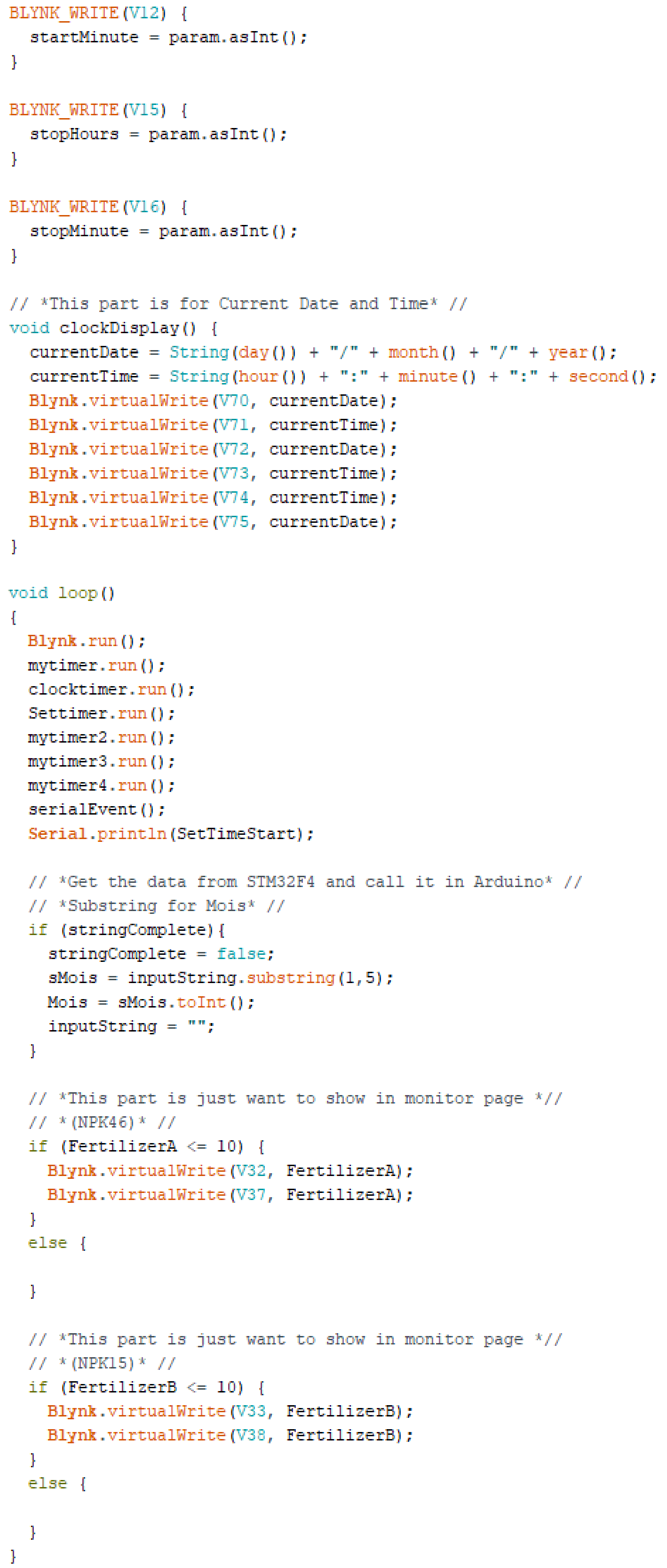
```
#define BLYNK_PRINT Serial
#define RXD2 16
#define TXD2 17
#define LINE_TOKEN  "81d5wfUFxa6RSkMNS9bInMJdCdAZgVDoAdaL3trz6Ak"
/* Fill-in your Template ID (only if using Blynk.Cloud) */
//#define BLYNK_TEMPLATE_ID   "YourTemplateID"

#include <TridentTD_LineNotify.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <TimeLib.h>
#include <WidgetRTC.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "THNi2gtYZIXy4G9viPuJvM2Y1DLaXDJ8";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Senee";
char pass[] = "12345543210";
String inputString= ""; // a string to hold incoming data
boolean stringComplete= false; // whether the string is complete
int btn = 0; // btn ON and OFF //
int btnFA = 0; // btn NPK46 //
int btnFB = 0; // btn NPK15 //
int btnSV = 0; // btn solenoid valve //
String StartDate = ""; // choose date //
int FertilizerA = 0; // choose value for NPK46 //
int FertilizerB = 0; // choose value for NPK15 //
String sMois = ""; // substring Mois //
int Mois = 0; //  Recieve the data from STM32F4 //
int date = 0; // send date to STM32F4//

//Variable for update to blynk
String startHours = "";
String startMinute = "";
String SetTimeStart = ""; // To combline startHours and startpMinute together //
String sStart = ""; // For LineNotify comblined text with the data //
String stopHours = "";
String stopMinute = "";
String SetTimeStop = ""; // To combline stopHours and stopMinute together //
```



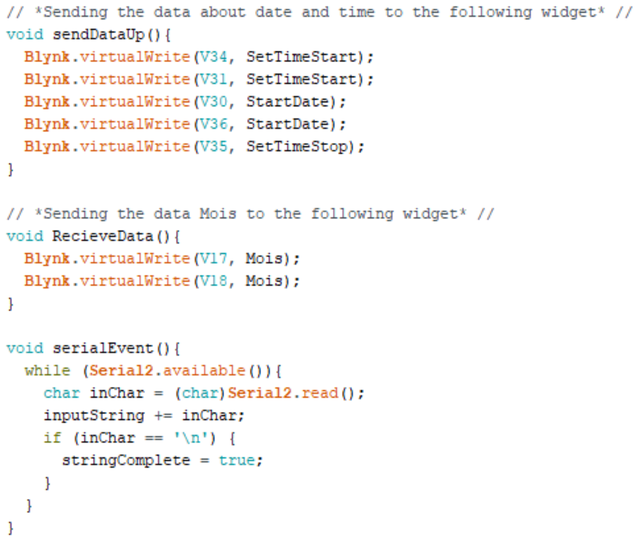
```
// Blynk variable //
BlynkTimer clocktimer;
BlynkTimer mytimer;
BlynkTimer mytimer2;
BlynkTimer mytimer3;
BlynkTimer mytimer4;
BlynkTimer Settimer;
WidgetRTC rtc;
WidgetLCD lcd(V3);

String currentDate; // Current Date using rtc widget //
String currentTime; // Current Time using rtc widget //


void setup()
{
  // Debug console
  Serial.begin(115200);
  Serial2.begin(115200, SERIAL_8N1, RXD2, TXD2);
  Blynk.begin(auth, ssid, pass, IPAddress(147, 182, 177, 185), 8080);
  rtc.begin();
  clocktimer.setInterval(100L, clockDisplay); //1sec
  LINE.setToken(LINE_TOKEN);
  mytimer.setInterval(1000L, sendDataUp);
  Settimer.setInterval(1000L, SetData);
  mytimer2.setInterval(100L,RecieveDataToSTM32F4);
  mytimer3.setInterval(1000L,RecieveData);
  mytimer4.setInterval(100L,sendDatatoSTM32F4);
  inputString.reserve(200);

}

BLYNK_WRITE(V10) {
  StartDate = param.asInt();
  date = param.asInt();
}
```



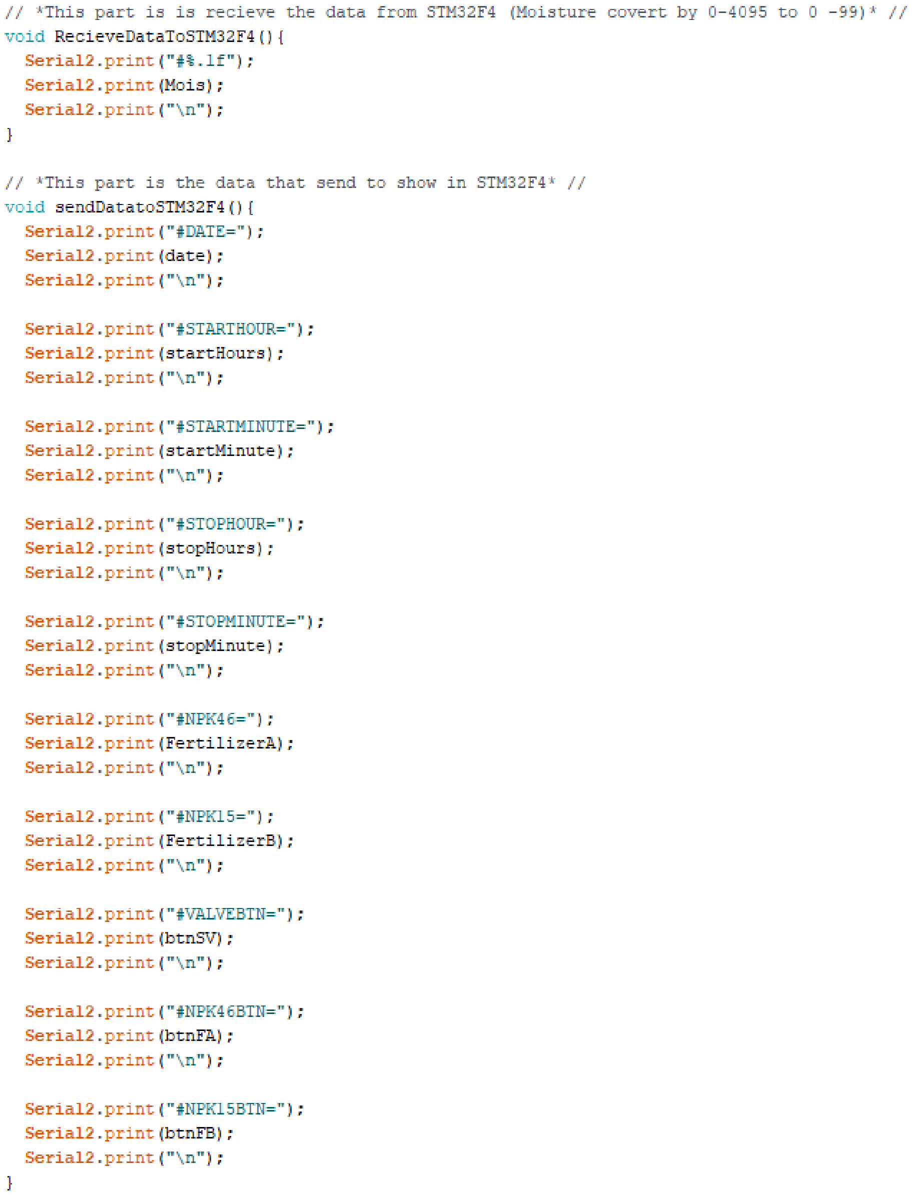
```
// *This one for LineNotify to send the data into Line applicattion if I clike the button on/off automatic* //
// *If I click on and off it will show different data by clicking the button* //
BLYNK_WRITE(V2){
    btn = param.asInt();
    // *ON Button* //
    if(btn == 0){
      LINE.notify("\n"
                  "Automatic Watering System Status:\n"
                  +sStart+ "\n"
                  "Stop Time: " +SetTimeStop+ "\n"
                  "Moisture: " +Mois+ "%\n"
                  "System Soil Moisture: Stop");
    }
    // *OFF Button* //
    else{
      LINE.notify("\n"
                  "Automatic Watering System Status:\n"
                  +sStart+ "\n"
                  "Start Time: " +SetTimeStart+ "\n"
                  "Moisture: " +Mois+ "%\n"
                  "System Soil Moisture: Working");
    }
}

void SetData(){
    if(btn == 0){
      SetTimeStart = startHours +":"+ startMinute; // *Set the data to combine it together* //
      SetTimeStop = stopHours +":"+ stopMinute;
      sStart = "System Day: " + StartDate;

      Blynk.virtualWrite(V1,"AUTOMATIC: OFF");
      lcd.print(0, 0, ">WATERING STATUS:"); // This part is for the lcd advance widget in blynk //
      lcd.print(0, 1, ">STOPPED"); // If the button is 0 it will show Stop //
    }
    else{
      Blynk.virtualWrite(V1,"AUTOMATIC: ON");
      lcd.print(0, 0, ">WATERING STATUS:");
      lcd.print(0, 1, ">WORKING"); // If the button is 1 it will show Working //

    }
}
 BLYNK_WRITE(V13) {
   FertilizerA = param.asInt();
 }

 BLYNK_WRITE(V14) {
   FertilizerB = param.asInt();
 }

 // *Button NPK46 to send text into the following widget* //
 BLYNK_WRITE(V4) {
   btnFA = param.asInt();
   if(btnFA == 0){
     Blynk.virtualWrite(V40,"OFF");
   }
   else{
     Blynk.virtualWrite(V40,"ON");
   }
 }

 // *Button NPK15 to send text into the following widget* //
 BLYNK_WRITE(V5) {
   btnFB = param.asInt();
   if(btnFB == 0){
     Blynk.virtualWrite(V41,"OFF");
   }
   else{
     Blynk.virtualWrite(V41,"ON");
   }
 }

 // *This one for Solenoid Valve switch* //
 // *And send the text into the following widget* //
 BLYNK_WRITE(V6) {
   btnSV = param.asInt();
   if(btnSV == 0){
     Blynk.virtualWrite(V42,"OFF"); // if btnSV is 0 that's mean it off //
   }
   else{
     Blynk.virtualWrite(V42,"ON"); // if btnSV is 1 that's mean it on //
   }
 }

 BLYNK_WRITE(V11) {
   startHours = param.asInt();
 }
```



```
BLYNK_WRITE(V12) {
  startMinute = param.asInt();
}

BLYNK_WRITE(V15) {
  stopHours = param.asInt();
}

BLYNK_WRITE(V16) {
  stopMinute = param.asInt();
}

// *This part is for Current Date and Time* //
void clockDisplay() {
  currentDate = String(day()) + "/" + month() + "/" + year();
  currentTime = String(hour()) + ":" + minute() + ":" + second();
  Blynk.virtualWrite(V70, currentDate);
  Blynk.virtualWrite(V71, currentTime);
  Blynk.virtualWrite(V72, currentDate);
  Blynk.virtualWrite(V73, currentTime);
  Blynk.virtualWrite(V74, currentTime);
  Blynk.virtualWrite(V75, currentDate);
}

void loop()
{
  Blynk.run();
  mytimer.run();
  clocktimer.run();
  Settimer.run();
  mytimer2.run();
  mytimer3.run();
  mytimer4.run();
  serialEvent();
  Serial.println(SetTimeStart);

  // *Get the data from STM32F4 and call it in Arduino* //
  // *Substring for Mois* //
  if (stringComplete){
    stringComplete = false;
    sMois = inputString.substring(1,5);
    Mois = sMois.toInt();
    inputString = "";
  }

  // *This part is just want to show in monitor page *//
  // *(NPK46)* //
  if (FertilizerA <= 10) {
    Blynk.virtualWrite(V32, FertilizerA);
    Blynk.virtualWrite(V37, FertilizerA);
  }
  else {

  }

  // *This part is just want to show in monitor page *//
  // *(NPK15)* //
  if (FertilizerB <= 10) {
    Blynk.virtualWrite(V33, FertilizerB);
    Blynk.virtualWrite(V38, FertilizerB);
  }
  else {

  }
}
```



```
// *Sending the data about date and time to the following widget* //
void sendDataUp(){
  Blynk.virtualWrite(V34, SetTimeStart);
  Blynk.virtualWrite(V31, SetTimeStart);
  Blynk.virtualWrite(V30, StartDate);
  Blynk.virtualWrite(V36, StartDate);
  Blynk.virtualWrite(V35, SetTimeStop);
}

// *Sending the data Mois to the following widget* //
void RecieveData(){
  Blynk.virtualWrite(V17, Mois);
  Blynk.virtualWrite(V18, Mois);
}

void serialEvent(){
  while (Serial2.available()){
    char inChar = (char)Serial2.read();
    inputString += inChar;
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
```

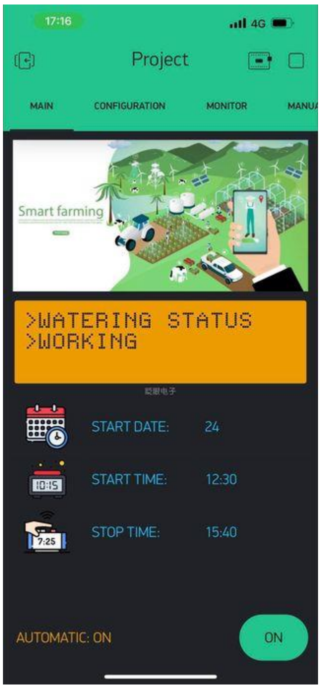


```
// *This part is is recieve the data from STM32F4 (Moisture covert by 0-4095 to 0 -99)* //
void RecieveDataToSTM32F4(){
  Serial2.print("#%.1f");
  Serial2.print(Mois);
  Serial2.print("\n");
}

// *This part is the data that send to show in STM32F4* //
void sendDatatoSTM32F4(){
  Serial2.print("#DATE=");
  Serial2.print(date);
  Serial2.print("\n");

  Serial2.print("#STARTHOUR=");
  Serial2.print(startHours);
  Serial2.print("\n");

  Serial2.print("#STARTMINUTE=");
  Serial2.print(startMinute);
  Serial2.print("\n");

  Serial2.print("#STOPHOUR=");
  Serial2.print(stopHours);
  Serial2.print("\n");

  Serial2.print("#STOPMINUTE=");
  Serial2.print(stopMinute);
  Serial2.print("\n");

  Serial2.print("#NPK46=");
  Serial2.print(FertilizerA);
  Serial2.print("\n");

  Serial2.print("#NPK15=");
  Serial2.print(FertilizerB);
  Serial2.print("\n");

  Serial2.print("#VALVEBTN=");
  Serial2.print(btnSV);
  Serial2.print("\n");

  Serial2.print("#NPK46BTN=");
  Serial2.print(btnFA);
  Serial2.print("\n");

  Serial2.print("#NPK15BTN=");
  Serial2.print(btnFB);
  Serial2.print("\n");
}
```

Result

- Blynk

-Main to show the about date and time that we select from config page.



>WATERING STATUS
>WORKING

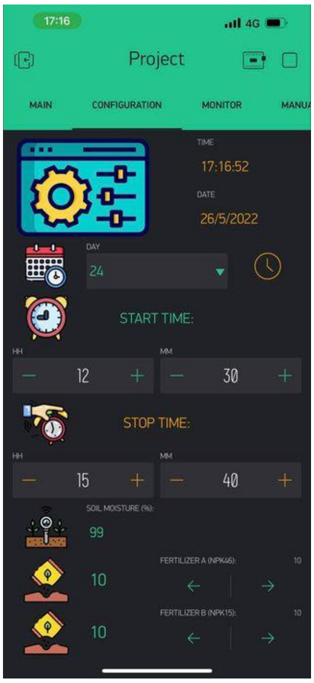
START DATE: 24

START TIME: 12:30

STOP TIME: 15:40

AUTOMATIC: ON

- Config page to select day, start time, stop time and fertilizer.



TIME 17:16:52

DATE 26/5/2022

DAY 24

START TIME:

HH 12 MM 30

STOP TIME:

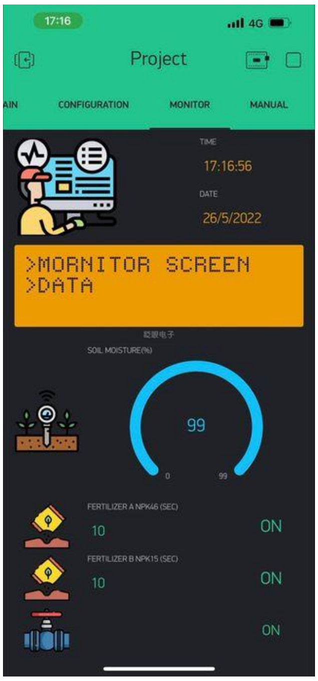
HH 15 MM 40

SOIL MOISTURE (%): 99

FERTILIZER A (NPK46): 10 10

FERTILIZER B (NPK15): 10 10

- Monitor Screen to show the status that we select from Config Screen



TIME 17:16:56

DATE 26/5/2022

>MORNITOR SCREEN
>DATA

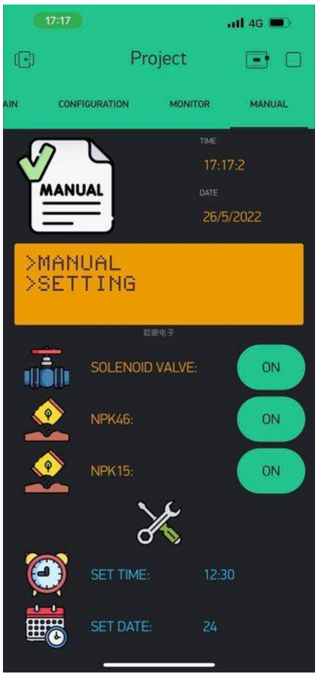
SOIL MOISTURE(%) 99

FERTILIZER A NPK46 (SEC) 10 ON

FERTILIZER B NPK15 (SEC) 10 ON

ON

- Manual to turn on and off valve, NPK46 and NPK15



TIME 17:17:2

DATE 26/5/2022

>MANUAL
>SETTING

SOLENOID VALVE: ON

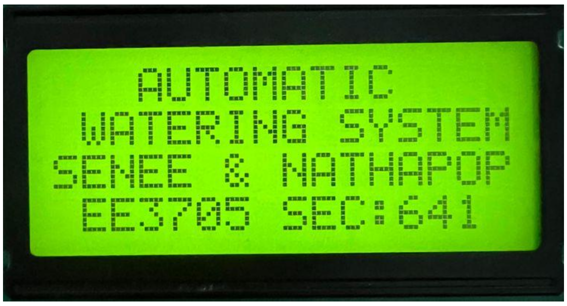
NPK46: ON

NPK15: ON

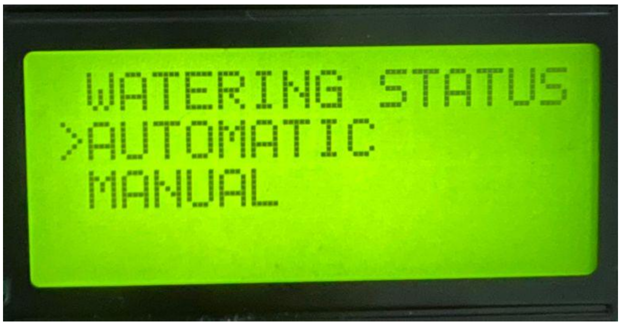
SET TIME: 12:30

SET DATE: 24

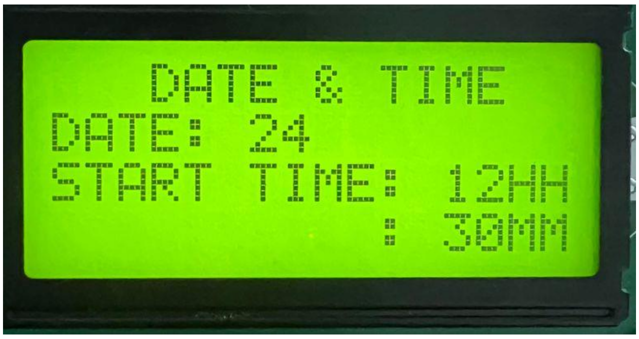
-STM32F4 (Automatic Mode)



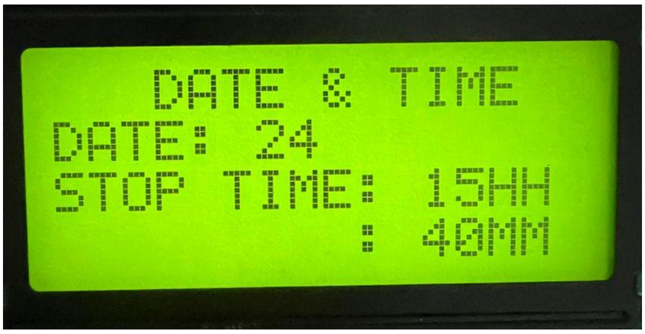
```
    AUTOMATIC
  WATERING SYSTEM
SENEE & NATHAPOP
 EE3705 SEC:641
```



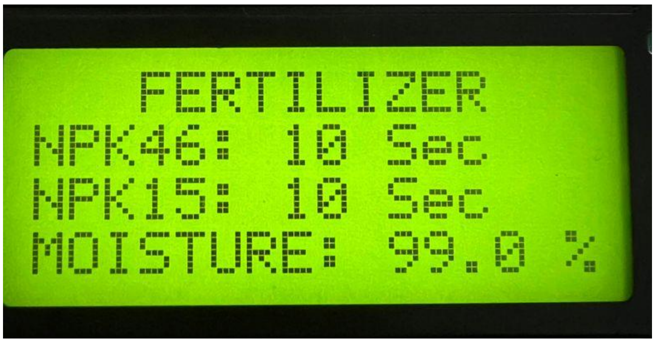
```
 WATERING STATUS
>AUTOMATIC
 MANUAL
```



```
   DATE & TIME
DATE: 24
START TIME: 12HH
          : 30MM
```

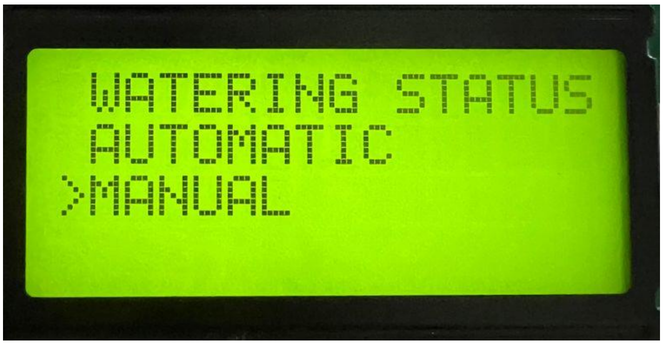


```
   DATE & TIME
DATE: 24
STOP TIME: 15HH
         : 40MM
```

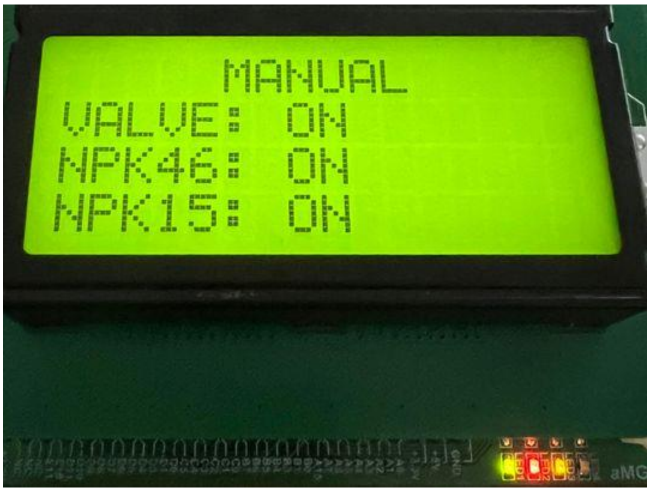


```
   FERTILIZER
NPK46: 10 Sec
NPK15: 10 Sec
MOISTURE: 99.0 %
```

-STM32F4 (Manual Mode)



```
  WATERING STATUS
  AUTOMATIC
 >MANUAL
```



```
      MANUAL
VALVE:  ON
NPK46:  ON
NPK15:  ON
```

LED

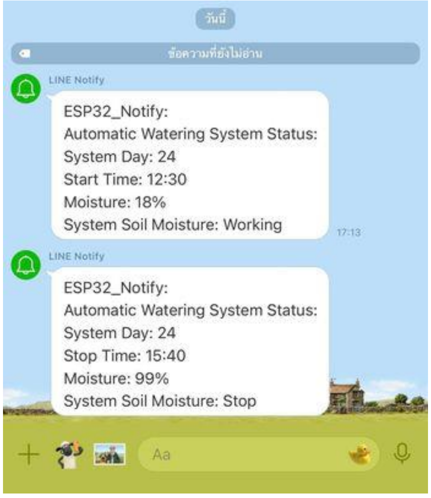
- Green is for Valve

- Red is for NPK46

- Yellow is for NPK15

(If the status on the LED will turn on too)

-Line notification



ESP32_Notify:
Automatic Watering System Status:
System Day: 24
Start Time: 12:30
Moisture: 18%
System Soil Moisture: Working

17:13

ESP32_Notify:
Automatic Watering System Status:
System Day: 24
Stop Time: 15:40
Moisture: 99%
System Soil Moisture: Stop

Video

https://drive.google.com/file/d/1Ak-moSVDb7BvbPXnJW2dl6l49sCfh3QK/view?usp=sharing

Conclusion

- From this project, I have learned about how STM32F4 and Blynk work together. Smart farm's automatic watering system is a system to help farmers make their lives easier in the covid-19 process. In addition, the system can help farmers monitor the production of water, fertilizer and moisture. The system also established a water supply cycle and a Line notification system.