

ภาคผนวก ก.

โปรแกรมควบคุม

## โปรแกรมควบคุม

### โปรแกรมของส่วนเปลี่ยนจ่ายน้ำยาล้างไต (Notification)

```
//-----CHECK-LED-----D3-D8 //สัญญาณไฟแอลอีดี (LED)
int Checkold = 3; //เตือนทางถุงน้ำยาเก่า
int Checknew = 4; //เตือนทางถุงน้ำยาใหม่
int Checkhuman = 5; //เตือนทางช่องท้องผู้ป่วย
int Checkse = 6; //เตือนความพร้อม
int Checkwater = 7; //เตือนติดเชื้อ
int Checkmemory = 8; //เตือนใช้ที่ซึ่งน้ำหนัก
//-----
//-----Reedswicth-----D9-D11
int Reedoldopen = 9; //สวิตซ์ทางถุงน้ำยาเก่า
int Reednewopen = 10; //สวิตซ์ทางถุงน้ำยาใหม่
int Reedhumanopen = 11; //สวิตซ์ทางช่องท้องผู้ป่วย
//-----
//-----Sensor-----A0
int senhelp = A0; //เซ็นเซอร์ตรวจเชื้อ
//-----
//-----Start-----D2
int start = 2; //สวิตซ์ความพร้อม
//-----
//-----Stage-----
boolean bReedoldopen = 0;
boolean bReednewopen = 0;
boolean bReedhumanopen = 0;
boolean bsenhelp = 0;
boolean bstart = 0;
//-----
//-----
int a = 0;
int b = 0;
int c = 0;
int d = 0;
int e = 0;
int f = 0;
//-----
void setup()
```

```

{
  //----INPUT-----
  pinMode(Reedoldopen,INPUT);
  pinMode(Reednewopen,INPUT);
  pinMode(Reedhumanopen,INPUT);

  pinMode(senhelpp,INPUT);

  pinMode(start,INPUT);
  //-----
  //----OUTPUT-----
  pinMode(Checkold,OUTPUT);
  pinMode(Checknew,OUTPUT);
  pinMode(Checkhuman,OUTPUT);
  pinMode(Checkse,OUTPUT);
  pinMode(Checkwater,OUTPUT);
  pinMode(Checkmemory,OUTPUT);
  //-----
}

void loop()
{
  //----int-----//---SW-LOW-กด---reed-LOW-กด
  bReedoldopen = digitalRead(Reedoldopen);
  bReednewopen = digitalRead(Reednewopen);
  bReedhumanopen = digitalRead(Reedhumanopen);
  bsenhelp = analogRead(senhelpp);
  bstart = digitalRead(start);
  //-----
  //=====StepI=====
  if(bReedoldopen == HIGH && bReednewopen == HIGH && bReedhumanopen ==
HIGH && bstart == HIGH)
  {
    digitalWrite(Checkold,LOW);
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checkhuman,HIGH); //start
  }
}

```

```

}
if(bReedoldopen == HIGH && bReednewopen == HIGH && bReedhumanopen ==
LOW && bstart == HIGH)
{
    digitalWrite(Checkhuman,LOW);
    digitalWrite(Checkold,LOW);

    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);

    digitalWrite(Checknew,HIGH);        //open line

}
if(bReedoldopen == HIGH && bReednewopen == LOW && bReedhumanopen ==
LOW && bstart == HIGH)
{
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkold,LOW);
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checkhuman,HIGH);    //open line waternew
}
if(bReedoldopen == HIGH && bReednewopen == LOW && bReedhumanopen ==
HIGH && bstart == HIGH)
{
    digitalWrite(Checkhuman,LOW);
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkold,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checkse,HIGH);        //
}
//=====OK
//=====StepII=====
//----test-----

```

```

if(bReedoldopen == HIGH && bReednewopen == LOW && bReedhumanopen ==
HIGH && bstart == LOW)
{
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkold,LOW);

    f++;          //waterold out 15 min(1000*60*15),checkwater,count number,35%
    d++;
    if(f<900)
    {
        a++;
    }
    bsenhelp     = digitalRead(senhelp);
    if(bsenhelp == LOW)
    {
        b++;
    }
    delay(1000);
    for(d==900;d>899;d++)
    {
        //----end-----
        digitalWrite(Checkhuman,HIGH);
        delay(1000);
        d=0;
        //-----
    }
}
//-----
/**
if(bReedoldopen == LOW && bReednewopen == LOW && bReedhumanopen ==
HIGH && bstart == LOW)
{
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);

```

```

    digitalWrite(Checkold,LOW);
    digitalWrite(Checkhuman,LOW);
    digitalWrite(Checknew,HIGH);          //
}
/**OK
if(bReedoldopen == HIGH && bReednewopen == LOW && bReedhumanopen ==
LOW && bstart == LOW)
{
    digitalWrite(Checkhuman,LOW);
    digitalWrite(Checkold,LOW);
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checknew,HIGH);        //
}
if(bReedoldopen == HIGH && bReednewopen == HIGH && bReedhumanopen ==
LOW && bstart == LOW)
{
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkhuman,LOW);
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checkold,HIGH);        //
}
if(bReedoldopen == LOW && bReednewopen == HIGH && bReedhumanopen ==
LOW && bstart == LOW)
{
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkold,LOW);        //selece v.water
    delay(500);
    digitalWrite(Checkold,HIGH);
    delay(500);
    digitalWrite(Checkmemory,HIGH);
    digitalWrite(Checkhuman,HIGH);
}

```

```

if(bReedoldopen == LOW && bReednewopen == HIGH && bReedhumanopen ==
HIGH && bstart == LOW)
{
    digitalWrite(Checkhuman,LOW);
    digitalWrite(Checkold,LOW);
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checkwater,LOW);
    e++;                //waternew in wait 15 min(60000*15)
    delay(1000);
    for(e==900;e>899;e++)
    {
        digitalWrite(Checkse,HIGH);
        delay(1000);
        e=0;
    }
}
//=====OK
//=====StepIII=====
if(bReedoldopen == LOW && bReednewopen == HIGH && bReedhumanopen ==
HIGH && bstart == HIGH)
{

    digitalWrite(Checkold,LOW);
    digitalWrite(Checknew,LOW);
    digitalWrite(Checkse,LOW);
    digitalWrite(Checkwater,LOW);
    digitalWrite(Checkmemory,LOW);
    digitalWrite(Checkhuman,HIGH);    //
}
if(bReedoldopen == LOW && bReednewopen == HIGH && bReedhumanopen ==
LOW && bstart == HIGH)
{
    digitalWrite(Checkhuman,LOW);
    digitalWrite(Checkold,LOW);
    digitalWrite(Checknew,LOW);
    c = (a*35)/100;                //
    if(b>c)

```

```

{
    digitalWrite(Checkwater,HIGH);
}
digitalWrite(Checkmemory,HIGH);
//-----THE-END-----
digitalWrite(Checkse,LOW);
delay(500);
digitalWrite(Checkse,HIGH);
delay(500);
//-----
}
//=====OK
}

```

### โปรแกรมของส่วนที่ชั่งน้ำหนัก (Weight)

```

#include <HX711.h>
#include <EEPROM.h>
#include "LedControlMS.h"
//-----D12,D11,D10,D1
LedControl lc=LedControl(12,11,10,1);    //แอลอีดี 7 ส่วน
//-----
//-----A0-A1
HX711 scale(A1, A0);                    //โหลดเซลล์
//-----
//-----D2-D5
const int buttonPin1 = 2;                //สวิทช์เริ่มต้นผู้ป่วยใหม่
const int buttonPin2 = 3;                //สวิทช์เก็บค่าน้ำยาใหม่
const int buttonPin3 = 4;                //สวิทช์เก็บค่าน้ำยาเก่า
const int buttonPin4 = 5;                //สวิทช์แสดงปริมาณน้ำต่อวัน
//-----
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
int lastButtonState1 = 0;
int lastButtonState2 = 0;
int lastButtonState3 = 0;
int lastButtonState4 = 0;

```

```

int v = 0;
int w = 0;
int y;
int x;
int z;

int w_round;
int water_in;
int water_out;
int gain_loss;
int total;
int water_day;
int ledState = LOW;
unsigned long previousMillis = 0;
const long interval = 300;
int led_pinb=16;
int last_total;
int weight;
void setup()
{
  Serial.begin(9600);
  //----LED-----D6-D8
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  //-----
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
  pinMode(buttonPin4, INPUT);
  lc.shutdown(0,false);
  lc.setIntensity(0,8);
  lc.clearDisplay(0);
  scale.set_scale(418.45f);           //ปรับค่านิเบท loadcell
  scale.tare();

  open_set();
  //----Round-----

```

```

Serial.println(eepread());
if(EEPROM.read(6)>4)
{
  EEPROM.write(6,1);
}
//-----
led_pinb=6;
}
void loop()
{
  buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);
  buttonState3 = digitalRead(buttonPin3);
  buttonState4 = digitalRead(buttonPin4);
  weight=scale.get_units(5), 1;
  y=ceil(weight/10)*10;          //ความละเอียด 10 g
  //=====Button1=====เก็บค่ามาตรฐาน 2200
  if (buttonState1 != lastButtonState1)
  {
    if (buttonState1 == LOW)
    {
      EEPROM.write(6,0);
      EEPROM.write(16,0);
      EEPROM.write(15,0);
      EEPROM.write(5000,0);
      v = 1;
      eepwrite(2200);
      water_in=2200;
      EEPROM.write(6,1);
    }
    lastButtonState1 = buttonState1;
  }
  //=====
  //=====Button2=====เก็บค่าน้ำหนักใหม่
  if (buttonState2 != lastButtonState2)
  {
    if (buttonState2 == LOW)
    {

```

```

if(v == 1)
{
    digitalWrite(6,LOW);
    led_pinb=7;
    water_in=2200;
    eepwrite(y);
}
else
{
    digitalWrite(6,LOW);
    led_pinb=7;
    water_in=eepread();
    eepwrite(y);
}
}
lastButtonState2 = buttonState2;
}
//=====
//=====Button3=====เก็บค่าน้ำหนักเก่า
if (buttonState3 != lastButtonState3)
{
    if (buttonState3 == LOW)
    {
        digitalWrite(6,LOW);
        digitalWrite(7,LOW);
        led_pinb=16;
        water_out=y-(2200-eepread()); //Cal_water_out
        gain_loss=water_in-water_out; //Cal_gain_loss
        if(EEPROM.read(16)==0)
        {
            last_total=tread();
        }
        else
        {
            last_total=tread()-(tread()*2);
        }
        if(EEPROM.read(6)==1)
        {

```

```

        total=gain_loss;
    }
    else
    {
        total=last_total+(gain_loss);    //Cal_total
    }
    if(total>0)
    {
        EEPROM.write(16,0);
        twrite(abs(total));
    }
    else
    {
        EEPROM.write(16,1);
        twrite(abs(total));
    }
    if(EEPROM.read(6)<=3)
    {
        water_day=800;
        x=water_day/100;
        EEPROM.write(5000,x);
        w = 1;
    }
    else
    {
        water_day=total+800;            //Cal_water_day
        x=water_day/100;
        EEPROM.write(5000,x);
        w = 1;
    }
    Serial.println("<" +String(EEPROM.read(6))+" "+String(water_in)+" "+String(water_out)+" "+
String(gain_loss)+" "+String(total)+" "+String(water_day)+" "+String(w,1,>));
    digitalWrite(8,HIGH);
    EEPROM.write(6,EEPROM.read(6)+1);
}
lastButtonState3 = buttonState3;
}
//=====

```

```

if(y>4000)
{
  set_char();
}
else
{
  //=====Button4=====
  if (buttonState4 == LOW)
  {
    if(w == 1)
    {
      print_digita(water_day);
    }
    else
    {
      z=EEPROM.read(5000)*100;
      print_digita(z);
    }
  }
  //=====
  else
  {
    print_digita(y);
  }
}
led_blink(led_pinb);
}
void led_blink(int ledPin)
{
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis >= interval)
  {
    previousMillis = currentMillis;
    if (ledState == LOW)
      ledState = HIGH;
    else
      ledState = LOW;
    digitalWrite(ledPin, ledState);
  }
}

```

```

    }
}
//----Overload-----
void set_char()
{
  lc.setChar(0,4,'-',false);
  lc.setChar(0,5,'-',false);
  lc.setChar(0,6,'-',false);
  lc.setChar(0,7,'-',false);
}
//-----
int tread()
{
  int nl=EEPROM.read(15);
  String weight;
  for(int i=10;i<=nl+10;i++)
  {
    weight+=String(EEPROM.read(i));
  }
  return weight.toInt();
}
void twrite(int in)
{
  int nl=String(in).length();
  for(int i=10;i<=nl+10;i++)
  {
    EEPROM.write(i,String(in).substring(i-10,(i-10)+1).toInt());
  }
  EEPROM.write(15,nl-1);
}
int eepread()
{
  int nl=EEPROM.read(5);
  String weight;
  for(int i=0;i<=nl;i++)
  {
    weight+=String(EEPROM.read(i));
  }
}

```

```

    return weight.toInt();
}
void eepwrite(int in)
{
    int nl=String(in).length();
    for(int i=0;i<=nl;i++)
    {
        EEPROM.write(i,String(in).substring(i,i+1).toInt());
    }
    EEPROM.write(5,nl-1);
}
void print_digita(int num)
{
    lc.clearDisplay(0);
    int nl=String(num).length();
    for(int i=0;i<=nl-1;i++)
    {
        lc.setDigit(0,i+4,String(num).substring(nl-i, (nl-1)-i).toInt(),false);
    }
}
void print_digitb(int num)
{
    int nl=String(num).length();
    for(int i=0;i<=nl-1;i++)
    {
        lc.setDigit(0,i,String(num).substring(i, i+1).toInt(),false);
    }
}
//----Open-----
void set_open1()
{
    lc.setChar(0,4,'-',false);
    lc.setChar(0,5,'0',false);
    lc.setChar(0,6,'0',false);
    lc.setChar(0,7,'-',false);
}
void set_open2()
{

```

```

lc.setChar(0,4,'0',false);
lc.setChar(0,5,'-',false);
lc.setChar(0,6,'-',false);
lc.setChar(0,7,'0',false);
}
//-----
void open_1()
{
  set_open1();
  delay(500);
}
void open_2()
{
  set_open2();
  delay(500);
}
void open_close()
{
  open_1();
  open_2();
}
void open_set()
{
  open_close();
  open_close();
  open_close();
}

```

### โปรแกรมของส่วนที่ซั้งน้ำหนั (Memory)

```

#include <Wire.h>
#include <Time.h>
#include <DS3232RTC.h>
const int chipSelect = 10;
#include <SdFat.h>
SdFat sd;
SdFile myFile;
//----SW-----
const int buttonPin1 = 2;

```

```

int buttonState1 = 0;
int lastButtonState1 = 0;
//-----
String command = "";
String command2="";
char commandStart = '<';
char commandEnd = '>';
String w_round;
String water_in;
String water_out;
String gain_loss;
String total;
String water_day;
String type;
int a = 0;
void setup()
{
  Serial.begin(9600);
  //----LED-----D6-D8
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  //-----
  //----CheckSD-----
  if (!Sd.init(SPI_HALF_SPEED, chipSelect))
  {
    Serial.println("Failed");
    digitalWrite(6,LOW);
  }
  else
  {
    Serial.println("Done");
    digitalWrite(6,HIGH);
  }
  //-----
  //----CheckTime-----
  setSyncProvider(RTC.get);
  if(timeStatus() != timeSet)

```

```

{
  Serial.println("Unable to sync with the RTC");
  digitalWrite(7,LOW);
}
else
{
  Serial.println("RTC has set the system time");
  digitalWrite(7,HIGH);
}
//-----
//----CheckFile-----
if(myFile.open("PD.csv"))
{
  Serial.println("YES");
  digitalWrite(8,LOW);
  myFile.close();
}
else
{
  Serial.println("NO");
  digitalWrite(8,HIGH);
}
//-----
}
void loop()
{
  buttonState1 = digitalRead(buttonPin1);
  //=====Button1=====build table
  if (buttonState1 != lastButtonState1)
  {
    if (buttonState1 == LOW)
    {
      a++;
      if(a == 1)
      {
        wsd();
        digitalWrite(8,LOW);
      }
    }
  }
}

```

```

else
{
    digitalWrite(8,LOW);
}
}
lastButtonState1 = buttonState1;
}
//=====
while (Serial.available() > 0)
{
    int input = Serial.read();
    if (input == commandStart)
    {
        command = "";
    }
    else if (input == commandEnd)
    {
        handle(command);
        command = "";
    }
    else
    {
        command += (char)input;
    }
}
}
void handle(String command)
{
    w_round=getStringPartByNr(command, ',', 0);
    water_in=getStringPartByNr(command, ',', 1);
    water_out=getStringPartByNr(command, ',', 2);
    gain_loss=getStringPartByNr(command, ',', 3);
    total=getStringPartByNr(command, ',', 4);
    water_day=getStringPartByNr(command, ',', 5);
    type=getStringPartByNr(command, ',', 5);
    Serial.print(day());
    Serial.print("/");
    Serial.print(month());

```

```

Serial.print("/");
Serial.print(year());
Serial.print(",");
Serial.print(w_round);
Serial.print(",");
Serial.print(water_in);
Serial.print(",");
Serial.print(water_out);
Serial.print(",");
Serial.print(gain_loss);
Serial.print(",");
Serial.print(total);
Serial.print(",");
Serial.print(water_day);
Serial.println("");
wsd2();
}
void wsd2()
{
  if (!myFile.open("PD.csv", O_RDWR | O_CREAT | O_AT_END))
  {
  }
  else
  {
    myFile.print(day());
    myFile.print("/");
    myFile.print(month());
    myFile.print("/");
    myFile.print(year());
    myFile.print(",");
    myFile.print(w_round);
    myFile.print(",");
    myFile.print(water_in);
    myFile.print(",");
    myFile.print(water_out);
    myFile.print(",");
    myFile.print(gain_loss);
    myFile.print(",");
  }
}

```

```

        myFile.print(total);
        myFile.print(",");
        myFile.print(water_day);
        myFile.println("");
        myFile.close();
        Serial.println("SD write");
    }
}
void wsd()
{
    if (!myFile.open("PD.csv", O_RDWR | O_CREAT | O_AT_END))
    {
    }
    else
    {
        myFile.print("DAY");
        myFile.print(",");
        myFile.print("ROUND");
        myFile.print(",");
        myFile.print("WATER-IN");
        myFile.print(",");
        myFile.print("WATER-OUT");
        myFile.print(",");
        myFile.print("GAIN/LOSS");
        myFile.print(",");
        myFile.print("TOTAL");
        myFile.print(",");
        myFile.print("WATER/DAY");
        myFile.println("");
        myFile.close();
        Serial.println("SD write");
    }
}
String getStringPartByNr(String data, char separator, int index)
{
    int stringData = 0;
    String dataPart = "";
    for(int i = 0; i<data.length()-1; i++)

```

```
{
  if(data[i]==separator)
  {
    stringData++;
  }
  else if(stringData==index)
  {
    dataPart.concat(data[i]);
  }
  else if(stringData>index)
  {
    return dataPart;
    break;
  }
}
return dataPart;
}
```