**ภาคผนวก ข**

**โปรแกรมการทำงาน**

#include <SPI.h>

#include <DS3231.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Keypad.h>

#include <SD.h>

#define IRQ (2)

#define RESET (3)

File myFile;

LiquidCrystal\_I2C lcd(0x27,20,4);

DS3231 rtc(SDA, SCL);

#define DS3231\_I2C\_ADDRESS 104

byte seconds, minutes, hours, day, date, month, year;

const int chipSelect=10;

boolean present=0;

float New =0;

float old =0;

int buttonPin = 0;

int count;

int Gr = 0;

const int proc\_pin = 13;

unsigned long time\_old, time\_new;

const byte ROWS= 4;

const byte COLS= 4;

unsigned long previousTime = 0;

int setTime1;

int state = 0;

int Top = 0;

int Sw = 0;

int Up = 0;

int Dw = 0;

int start = 0;

int Num=0;

String P1;

char customKey;

int p = 0;

int A = 0;

char data[16];

char weekDay[4];

byte tMSB, tLSB;

float my\_temp;

char my\_array[100];

char customKey111;

int hourupg;

int minupg;

int yearupg;

int monthupg;

int dayupg;

int dateupg;

int menu = 0;

int secupg;

bool blinking = true;

Time t;

char keys[ROWS][COLS]=

{

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'.', '0', '#', 'D'},

};

byte rowPins[ROWS] = {9,8,7,6};

byte colPins[COLS] = {5,4,3,2};

Keypad keypad= Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

String sta = "";

int Station;

int Sn;int Sn2;int Sn3;int Sn4;int Sn5;int Sn6;int Sn7;int Sn8;int Sn9;int Sn10;

int Sn11;int Sn12;int Sn13;int Sn14;int Sn15;int Sn16;int Sn17;int Sn18;int Sn19;

int Sn20;

String rrd = "";

float River\_depth;

float Rd;float Rd2;float Rd3;float Rd4;float Rd5;float Rd6;float Rd7;float Rd8;float Rd9;float Rd10;

float Rd11;float Rd12;float Rd13;float Rd14;float Rd15;float Rd16;float Rd17;float Rd18;

float Rd19;float Rd20;float Rd21;float Rd22;float Rd23;float Rd24;float Rd25;float Rd26;

float Rd27;float Rd28;float Rd29;float Rd30;

String dep = "";

float Depth;float Dh;float Dh2;float Dh3;float Dh4;float Dh5;float Dh6;float Dh7;

float Dh8;float Dh9;float Dh10;float Dh11;float Dh12;float Dh13;float Dh14;float Dh15;

float Dh16;float Dh17;float Dh18;float Dh19;float Dh20;float Dh21;float Dh22;float Dh23;

float Dh24;float Dh25;float Dh26;float Dh27;float Dh28;float Dh29;float Dh30;float Dh31;

float Dh32;float Dh33;float Dh34;float Dh35;float Dh36;float Dh37;float Dh38;float Dh39;

float Dh40;float Dh41;float Dh42;float Dh43;float Dh44;float Dh45;float Dh46;float Dh47;

float Dh48;float Dh49;float Dh50;float Dh51;float Dh52;float Dh53;float Dh54;

float Dh55;float Dh56;float Dh57;float Dh58;float Dh59;float Dh60;float Dh61;

float Dh62;float Dh63;float Dh64;float Dh65;float Dh66;float Dh67;float Dh68;

float Dh69;float Dh70;float Dh71;float Dh72;float Dh73;float Dh74;float Dh75;

float Dh76;float Dh77;float Dh78;float Dh79;float Dh80;float Dh81;float Dh82;

float Dh83;float Dh84;float Dh85;float Dh86;float Dh87;float Dh88;float Dh89;

float Dh90;float Dh91;float Dh92;float Dh93;float Dh94;float Dh95;float Dh96;

float Dh97;float Dh98;float Dh99;float Dh100;float Dh101;float Dh102;float Dh103;

float Dh104;float Dh105;float Dh106;float Dh107;float Dh108;float Dh109;float Dh110;

String str = "";

float flo;

String are = "";

float Area;

float Aa;float Aa2;float Aa3;float Aa4;float Aa5;float Aa6;float Aa7;

float Aa8;float Aa9;float Aa10;float Aa11;

float Q;float Q2;float Q3;float Q4;float Q5;float Q6;float Q7;float Q8;

float Q9;float Q10;float Q11;float Q12;float Q13;float Q14;float Q15;

float Q16;float Q17;float Q18;float Q19;float Q20;float Q21;float Q22;

float Q23;float Q24;float Q25;float Q26;float Q27;float Q28;float Q29;

float T1;float T2;

float T3;float T4;float T5;float T6;float T7;float T8;

float T9;float T10;float T11;float T12;float T13;

float T14;float T15;float T16;float T17;float T18;float T19;float T20;float T21;

float T22;float T23;float T24;float T25;float T26;float T27;float T28;float T29;

float T30;float T31;float T32;float T33;float T34;float T35;float T36;float T37;

float T38;float T39;float T40;float T41;float T42;

float co;float co2;float co3;float co4;float co5;

float co6;float co7;float co8;float co9;

float co10;float co11;float co12;float co13;float co14;float co15;float co16;float co17;

float co18;float co19;float co20;float co21;float co22;float co23;float co24;float co25;

float co26;float co27;float co28;float co29;float co30;float co31;float co32;float co33;

float co34;float co35;float co36;float co37;float co38;float co39;float co40;float co41;

float co42;

float ans;float ans1;float ans2;float ans3;float ans4;float ans5;float ans6;float ans7;

float ans8;float ans9;float ans10;float ans11;float ans12;float ans13;float ans14;float ans15;

float ans16;float ans17;float ans18;float ans19;float ans20;float ans21;float ans22;float ans23;

float ans24;float ans25;float ans26;float ans27;float ans28;float ans29;float ans30;float ans31;

float ans32;float ans33;float ans34;float ans35;float ans36;float ans37;float ans38;float ans39;

float ans40;float ans41;float ans42;

float V;float V2;float V3;float V4;float V5;float V6;float V7;float V8;

float V9;float V10;float V11;float V12;float V13;float V14;float V15;float V16;

float V17;float V18;float V19;float V20;float V21;float V22;float V23;float V24;

float V25;float V26;float V27;float V28;float V29;float V30;float V31;float V32;

float V33;float V34;float V35;float V36;

float V37;float V38;float V39;float V40;float V41;float V42;

void setup()

{

Serial.begin(9600);

pinMode(52, INPUT);

pinMode(51, INPUT);

Wire.begin(); //begin I2C

lcd.init();

lcd.backlight();

checkSD();

rtc.begin();

t = rtc.getTime();

if(present==1)

{

lcd.clear();

lcd.setCursor(7,0);

lcd.print("RMUTR");

lcd.setCursor(0,1);

lcd.print("Flowrate PROJECT :)");

}

}

void loop()

{

switch (Sw)

{

case 1: state = 1; break; case 2: state = 2; break;

case 3: state = 3; break; case 4: state = 4; break;

case 5: state = 5; break; case 6: state = 6; break;

case 7: state = 7; break; case 8: state = 8; break;

case 9: state = 9; break; case 10: state = 10; break;

case 11: state = 11; break; case 12: state = 12; break;

case 13: state = 13; break; case 14: state = 14; break;

case 15: state = 15; break; case 16: state = 16; break;

case 17: state = 17; break; case 18: state = 18; break;

case 19: state = 19; break; case 20: state = 20; break;

case 21: state = 21; break; case 22: state = 22; break;

case 23: state = 23; break; case 24: state = 24; break;

case 25: state = 25; break; case 26: state = 26; break;

case 27: state = 27; break; case 28: state = 28; break;

case 29: state = 29; break; case 30: state = 30; break;

case 31: state = 31; break; case 32: state = 32; break;

case 33: state = 33; break; case 34: state = 34; break;

case 35: state = 35; break; case 36: state = 36; break;

case 37: state = 37; break; case 38: state = 38; break;

case 39: state = 39; break; case 40: state = 40; break;

case 41: state = 41; break; case 42: state = 42; break;

case 43: state = 43; break; case 44: state = 44; break;

case 45: state = 45; break; case 46: state = 46; break;

case 47: state = 47; break; case 48: state = 48; break;

case 49: state = 49; break; case 50: state = 50; break;

case 51: state = 51; break; case 52: state = 52; break;

case 53: state = 53; break; case 54: state = 54; break;

case 55: state = 55; break; case 56: state = 56; break;

case 57: state = 57; break; case 58: state = 58; break;

case 59: state = 59; break; case 60: state = 60; break;

case 61: state = 61; break; case 62: state = 62; break;

case 63: state = 63; break; case 64: state = 64; break;

case 65: state = 65; break;

break;

}

char key = keypad.getKey();

unsigned long showTime = millis();

if(state==1)

{

PROVINCE();

}

if(state==2)

{ int Credit,Num;

if(key != NO\_KEY ){

if (key == '.')

{

menu=menu+1;

if(menu==2){p=1;}

if(menu==1){p=0;}

}

}

if (menu==0)

{

t = rtc.getTime();

lcd.setCursor(2,1);

display\_position(t.hour);

lcd.print(":");

display\_position(t.min);

lcd.print(":");

display\_position(t.sec);

lcd.setCursor(2,2);

lcd.print(rtc.getDOWStr(1));

lcd.setCursor(8,2);

display\_position(t.date);

lcd.print("/");

display\_position(t.mon);

lcd.print("/");

display\_position(t.year);

}

if (menu==1)

{

DisplaySetHour(); // set hour

}

if (menu==2)

{

DisplaySetMinute(); // set minute

}

if (menu==3)

{

DisplaySetDay(); // set day

}

if (menu==4)

{

DisplaySetDate(); // set date

}

if (menu==5)

{

DisplaySetMonth(); // set Month

}

if (menu==6)

{

DisplaySetYear(); // set year

if(minupg <= 0 && hourupg <= 0 && dayupg <= 0 && dateupg <= 0 && monthupg <= 0 && yearupg <= 0)

{

menu = 0;

lcd.clear();

lcd.setCursor(0,1);

lcd.print(" ERROR ");

delay(500);

}

}

if (menu==7)

{

StoreAgg(); //setup time date to DS3231

Serial.print(minupg);

delay(500);

menu=0; // return to main menu

}

delay(100);

}

if(state==3)

{

lcd.setCursor(5,0);

lcd.print("Station");

lcd.setCursor(0,2);

lcd.print("Name:");

lcd.setCursor(7,2);

lcd.print(sta);

Sn=Station;

}

if(state==4)

{

lcd.setCursor(2,0);

lcd.print("River depth");

lcd.setCursor(0,2);

lcd.print("Depth:");

lcd.setCursor(7,2);

lcd.print(rrd);

Rd=River\_depth;

}

if(state==5)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh=Depth;

}

if(state==6)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh);

lcd.setCursor(8,0);

lcd.print(Sn);

lcd.setCursor(7,2);

lcd.print(str);

co2=co;

T2=T1;

ans2=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==7)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn);

lcd.setCursor(8,1);

lcd.print(Dh);

lcd.setCursor(6,2);

lcd.print(ans2,4);

lcd.setCursor(4,3);

V =(ans2\*(0.2594-0.0086));

lcd.print(V,4);

}

if(state==8)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh2=Depth;

}

if(state==9)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh2);

lcd.setCursor(8,0);

lcd.print(Sn);

lcd.setCursor(7,2);

lcd.print(str);

co3=co;

T3=T1;

ans3=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==10)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn);

lcd.setCursor(8,1);

lcd.print(Dh2);

lcd.setCursor(6,2);

lcd.print(ans3,4);

lcd.setCursor(4,3);

V2 =(ans3\*(0.2594-0.0086));

lcd.print(V2,4);

}

if(state==11)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh3=Depth;

}

if(state==12)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh3);

lcd.setCursor(8,0);

lcd.print(Sn);

lcd.setCursor(7,2);

lcd.print(str);

co4=co;

T4=T1;

ans4=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==13)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn);

lcd.setCursor(8,1);

lcd.print(Dh3);

lcd.setCursor(6,2);

lcd.print(ans4,4);

lcd.setCursor(4,3);

V3 =(ans4\*(0.2594-0.0086));

lcd.print(V3,4);

}

if(state==14)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh4=Depth;

}

if(state==15)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh4);

lcd.setCursor(8,0);

lcd.print(Sn);

lcd.setCursor(7,2);

lcd.print(str);

co5=co;

T5=T1;

ans5=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==16)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn);

lcd.setCursor(8,1);

lcd.print(Dh4);

lcd.setCursor(6,2);

lcd.print(ans5,4);

lcd.setCursor(4,3);

V4 =(ans5\*(0.2594-0.0086));

lcd.print(V4,4);

}

if(state==17)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(10,0);

lcd.print(Sn);

lcd.setCursor(0,2);

lcd.print("A =");

lcd.setCursor(7,2);

lcd.print(are);

Aa=Area;

}

if(state==18)

{

lcd.setCursor(0,0);

lcd.print("Q =");

Q=(Aa\*(V+V2+V3+V4)/2);

lcd.setCursor(6,0);

lcd.print(Q,4);

lcd.setCursor(13,0);

lcd.print("m^3/s");

lcd.setCursor(3,1);

Q2=(Q\*60);

lcd.print(Q2,4);

lcd.setCursor(13,1);

lcd.print("m^3/min");

lcd.setCursor(3,2);

Q3=(Q\*3600);

lcd.print(Q3,4);

lcd.setCursor(13,2);

lcd.print("m^3/hr");

lcd.setCursor(0,3);

lcd.print("River\_depth =");

lcd.setCursor(14,3);

lcd.print(Rd);

}

if(state==19)

{

lcd.setCursor(5,0);

lcd.print("Station");

lcd.setCursor(0,2);

lcd.print("Name:");

lcd.setCursor(7,2);

lcd.print(sta);

Sn2=Station;

}

if(state==20)

{

lcd.setCursor(2,0);

lcd.print("River depth");

lcd.setCursor(0,2);

lcd.print("Depth:");

lcd.setCursor(7,2);

lcd.print(rrd);

Rd2=River\_depth;

}

if(state==21)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh5=Depth;

}

if(state==22)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh5);

lcd.setCursor(8,0);

lcd.print(Sn2);

lcd.setCursor(7,2);

lcd.print(str);

co6=co;

T6=T1;

ans6=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==23)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn2);

lcd.setCursor(8,1);

lcd.print(Dh5);

lcd.setCursor(6,2);

lcd.print(ans6,4);

lcd.setCursor(4,3);

V5 =(ans6\*(0.2594-0.0086));

lcd.print(V5,4);

}

if(state==24)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh6=Depth;

}

if(state==25)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh6);

lcd.setCursor(8,0);

lcd.print(Sn2);

lcd.setCursor(7,2);

lcd.print(str);

co7=co;

T7=T1;

ans7=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==26)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn2);

lcd.setCursor(8,1);

lcd.print(Dh6);

lcd.setCursor(6,2);

lcd.print(ans7,4);

lcd.setCursor(4,3);

V6 =(ans7\*(0.2594-0.0086));

lcd.print(V6,4);

}

if(state==27)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh7=Depth;

}

if(state==28)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh7);

lcd.setCursor(8,0);

lcd.print(Sn2);

lcd.setCursor(7,2);

lcd.print(str);

co8=co;

T8=T1;

ans8=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==29)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn2);

lcd.setCursor(8,1);

lcd.print(Dh7);

lcd.setCursor(6,2);

lcd.print(ans8,4);

lcd.setCursor(4,3);

V7 =(ans8\*(0.2594-0.0086));

lcd.print(V7,4);

}

if(state==30)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh8=Depth;

}

if(state==31)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh8);

lcd.setCursor(8,0);

lcd.print(Sn2);

lcd.setCursor(7,2);

lcd.print(str);

co9=co;

T9=T1;

ans9=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==32)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn2);

lcd.setCursor(8,1);

lcd.print(Dh8);

lcd.setCursor(6,2);

lcd.print(ans9,4);

lcd.setCursor(4,3);

V8 =(ans9\*(0.2594-0.0086));

lcd.print(V8,4);

}

if(state==33)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(10,0);

lcd.print(Sn2);

lcd.setCursor(0,2);

lcd.print("A =");

lcd.setCursor(7,2);

lcd.print(are);

Aa2=Area;

}

if(state==34)

{

lcd.setCursor(0,0);

lcd.print("Q =");

Q4=(Aa2\*(V5+V6+V7+V8)/2);

lcd.setCursor(6,0);

lcd.print(Q4,4);

lcd.setCursor(13,0);

lcd.print("m^3/s");

lcd.setCursor(3,1);

Q5=(Q4\*60);

lcd.print(Q5,4);

lcd.setCursor(13,1);

lcd.print("m^3/min");

lcd.setCursor(3,2);

Q6=(Q4\*3600);

lcd.print(Q6,4);

lcd.setCursor(13,2);

lcd.print("m^3/hr");

lcd.setCursor(0,3);

lcd.print("River\_depth =");

lcd.setCursor(14,3);

lcd.print(Rd2);

}

if(state==35)

{

lcd.setCursor(5,0);

lcd.print("Station");

lcd.setCursor(0,2);

lcd.print("Name:");

lcd.setCursor(7,2);

lcd.print(sta);

Sn3=Station;

}

if(state==36)

{

lcd.setCursor(2,0);

lcd.print("River depth");

lcd.setCursor(0,2);

lcd.print("Depth:");

lcd.setCursor(7,2);

lcd.print(rrd);

Rd3=River\_depth;

}

if(state==37)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh9=Depth;

}

if(state==38)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh9);

lcd.setCursor(8,0);

lcd.print(Sn3);

lcd.setCursor(7,2);

lcd.print(str);

co10=co;

T10=T1;

ans10=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==39)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn3);

lcd.setCursor(8,1);

lcd.print(Dh9);

lcd.setCursor(6,2);

lcd.print(ans10,4);

lcd.setCursor(4,3);

V9 =(ans10\*(0.2594-0.0086));

lcd.print(V9,4);

}

if(state==40)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh10=Depth;

}

if(state==41)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh10);

lcd.setCursor(8,0);

lcd.print(Sn3);

lcd.setCursor(7,2);

lcd.print(str);

co11=co;

T11=T1;

ans11=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==42)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn3);

lcd.setCursor(8,1);

lcd.print(Dh10);

lcd.setCursor(6,2);

lcd.print(ans11,4);

lcd.setCursor(4,3);

V10 =(ans11\*(0.2594-0.0086));

lcd.print(V10,4);

}

if(state==43)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh11=Depth;

}

if(state==44)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh11);

lcd.setCursor(8,0);

lcd.print(Sn3);

lcd.setCursor(7,2);

lcd.print(str);

co12=co;

T12=T1;

ans12=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==45)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn3);

lcd.setCursor(8,1);

lcd.print(Dh11);

lcd.setCursor(6,2);

lcd.print(ans12,4);

lcd.setCursor(4,3);

V11 =(ans12\*(0.2594-0.0086));

lcd.print(V11,4);

}

if(state==46)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh12=Depth;

}

if(state==47)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh12);

lcd.setCursor(8,0);

lcd.print(Sn3);

lcd.setCursor(7,2);

lcd.print(str);

co13=co;

T13=T1;

ans13=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==48)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn3);

lcd.setCursor(8,1);

lcd.print(Dh12);

lcd.setCursor(6,2);

lcd.print(ans13,4);

lcd.setCursor(4,3);

V12 =(ans13\*(0.2594-0.0086));

lcd.print(V12,4);

}

if(state==49)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(10,0);

lcd.print(Sn3);

lcd.setCursor(0,2);

lcd.print("A =");

lcd.setCursor(7,2);

lcd.print(are);

Aa3=Area;

}

if(state==50)

{

lcd.setCursor(0,0);

lcd.print("Q =");

Q7=(Aa3\*(V9+V10+V11+V12)/2);

lcd.setCursor(6,0);

lcd.print(Q7,4);

lcd.setCursor(13,0);

lcd.print("m^3/s");

lcd.setCursor(3,1);

Q8=(Q7\*60);

lcd.print(Q8,4);

lcd.setCursor(13,1);

lcd.print("m^3/min");

lcd.setCursor(3,2);

Q9=(Q7\*3600);

lcd.print(Q6,4);

lcd.setCursor(13,2);

lcd.print("m^3/hr");

lcd.setCursor(0,3);

lcd.print("River\_depth =");

lcd.setCursor(14,3);

lcd.print(Rd3);

}

if(state==51)

{

lcd.setCursor(5,0);

lcd.print("Station");

lcd.setCursor(0,2);

lcd.print("Name:");

lcd.setCursor(7,2);

lcd.print(sta);

Sn4=Station;

}

if(state==52)

{

lcd.setCursor(2,0);

lcd.print("River depth");

lcd.setCursor(0,2);

lcd.print("Depth:");

lcd.setCursor(7,2);

lcd.print(rrd);

Rd4=River\_depth;

}

if(state==53)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh13=Depth;

}

if(state==54)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh13);

lcd.setCursor(8,0);

lcd.print(Sn4);

lcd.setCursor(7,2);

lcd.print(str);

co14=co;

T14=T1;

ans14=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==55)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn4);

lcd.setCursor(8,1);

lcd.print(Dh13);

lcd.setCursor(6,2);

lcd.print(ans14,4);

lcd.setCursor(4,3);

V13 =(ans14\*(0.2594-0.0086));

lcd.print(V13,4);

}

if(state==56)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh14=Depth;

}

if(state==57)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh14);

lcd.setCursor(8,0);

lcd.print(Sn4);

lcd.setCursor(7,2);

lcd.print(str);

co15=co;

T15=T1;

ans15=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==58)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn4);

lcd.setCursor(8,1);

lcd.print(Dh14,4);

lcd.setCursor(6,2);

lcd.print(ans15,4);

lcd.setCursor(4,3);

V14 =(ans15\*(0.2594-0.0086));

lcd.print(V14,4);

}

if(state==59)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh15=Depth;

}

if(state==60)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh15);

lcd.setCursor(8,0);

lcd.print(Sn4);

lcd.setCursor(7,2);

lcd.print(str);

co16=co;

T16=T1;

ans16=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==61)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn4);

lcd.setCursor(8,1);

lcd.print(Dh15);

lcd.setCursor(6,2);

lcd.print(ans16,4);

lcd.setCursor(4,3);

V15 =(ans16\*(0.2594-0.0086));

lcd.print(V15,4);

}

if(state==62)

{

lcd.setCursor(0, 0);

lcd.print("Level of depth");

lcd.setCursor(0,2);

lcd.print("Level:");

lcd.setCursor(7,2);

lcd.print(dep);

Dh16=Depth;

}

if(state==63)

{

setTime1=flo;

buttonPin = digitalRead(51);

Point();

lcd.setCursor(0,0);

lcd.print("Station");

lcd.setCursor(0,1);

lcd.print("count:");

lcd.setCursor(0,2);

lcd.print("Start:");

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.setCursor(12,0);

lcd.print(Dh16);

lcd.setCursor(8,0);

lcd.print(Sn4);

lcd.setCursor(7,2);

lcd.print(str);

co17=co;

T17=T1;

ans17=ans;

if(key != NO\_KEY )

{

if(key == 'B')

{

count=0;

start = 1;

previousTime = showTime;

flo = str.toFloat();

lcd.setCursor(0,3);

lcd.print(flo,3);

}

}

}

if(state==64)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(0,1);

lcd.print("Depth =");

lcd.setCursor(0,2);

lcd.print("N/s =");

lcd.setCursor(0,3);

lcd.print("V =");

lcd.setCursor(10,0);

lcd.print(Sn4);

lcd.setCursor(8,1);

lcd.print(Dh16);

lcd.setCursor(6,2);

lcd.print(ans17,4);

lcd.setCursor(4,3);

V16 =(ans17\*(0.2594-0.0086));

lcd.print(V16,4);

}

if(state==65)

{

lcd.setCursor(0,0);

lcd.print("Staton =");

lcd.setCursor(10,0);

lcd.print(Sn4);

lcd.setCursor(0,2);

lcd.print("A =");

lcd.setCursor(7,2);

lcd.print(are);

Aa4=Area;

}

if(state==66)

{

lcd.setCursor(0,0);

lcd.print("Q =");

Q10=(Aa4\*(V13+V14+V15+V16)/2);

lcd.setCursor(6,0);

lcd.print(Q10,4);

lcd.setCursor(13,0);

lcd.print("m^3/s");

lcd.setCursor(3,1);

Q11=(Q10\*60);

lcd.print(Q11,4);

lcd.setCursor(13,1);

lcd.print("m^3/min");

lcd.setCursor(3,2);

Q12=(Q10\*3600);

lcd.print(Q12,4);

lcd.setCursor(13,2);

lcd.print("m^3/hr");

lcd.setCursor(0,3);

lcd.print("River\_depth =");

lcd.setCursor(14,3);

lcd.print(Rd4);

}

if(start == 1)

{

lcd.setCursor(9,3);

lcd.print("setTime:");

lcd.print(setTime1); //แสดงค่าเวลาที่ตั้งไว้

T1 =(setTime1);

lcd.setCursor(0,3);

lcd.print("Time:");

lcd.print((showTime - previousTime) / 1000); //แสดงผลการคำนวณว่าจับเวลาไปกี่วิแล้ว

lcd.print(" ");

lcd.setCursor(8,1);

lcd.print(count);

co =(count);

if((showTime - previousTime) / 1000 >= setTime1)

{

start = 0;

lcd.setCursor(0,2);

lcd.print(" ");

ans = (co/T1);

lcd.setCursor(12,1);

lcd.print(ans,4);

lcd.print(" ");

}

}

if(key != NO\_KEY && key != 'B'&& key != 'C')

{

setTime1 = (key - 48);

if (key == 'A')

{

Station = sta.toFloat();

River\_depth = rrd.toFloat();

Depth = dep.toFloat();

flo = str.toFloat();

Area =are.toFloat();

lcd.setCursor(16,0);

lcd.print("Save");

}

else

{

sta +=key;

rrd +=key;

dep +=key;

str +=key;

are +=key;

lcd.print(key);

}

}

if(key == '#')

{

lcd.setCursor(7,2);

sta="";

lcd.print(" ");

lcd.setCursor(7,2);

rrd="";

lcd.print(" ");

lcd.setCursor(7,2);

dep="";

lcd.print(" ");

lcd.setCursor(7,2);

str="";

lcd.print(" ");

lcd.setCursor(7,2);

are="";

lcd.print(" ");

key=0;

}

if (key == 'C')

{

key == Sw;

Sw--;

key=0;

count=0;

lcd.clear();

}

if (key == 'D')

{

key == Sw;

Sw++;

sta="";

rrd="";

dep="";

str="";

are="";

Station=0;

River\_depth=0;

Depth=0;

flo=0;

Area=0;

key=0;

count=0;

co=0;

T1=0;

ans=0;

lcd.clear();

}

Gr = digitalRead(52);

if (Gr == LOW)

{

lcd.setCursor(0,0);

lcd.print("Save ");

delay(2000);

key=0;

count=0;

writeSDStation();

}

}