

```
#รวม <Adafruit_ADS1015.h>
```

```
#รวม <stdio.h>
```

```
#รวม <string.h>
```

```
#รวม <Arduino.h>
```

```
#รวม <U8g2lib.h>
```

```
#รวม <WiFi.h>
```

```
#รวม <WiFiMulti.h>
```

```
#รวม <HttpClient.h>
```

```
/*
```

```
 * เริ่มต้นหน่วยการเชื่อมต่อ wifi
```

```
*/
```

```
/*
```

```
 * ที่อยู่เซิร์ฟเวอร์
```

```
*/
```

```
#กำหนด SERVER_ADDR "http://192.168.137.1:5000/data"
```

```
/*
```

```
 * กำหนดพารามิเตอร์ต่าง ๆ ในกระบวนการวัด
```

```
 *
```

```
 * ----- พารามิเตอร์ต่อไปนี้สามารถแก้ไขได้ -----
```

\* DEFLATION\_TIME หมายถึง ภาวะเงินฝืดอย่างช้าๆ ซึ่งเป็นเวลาของการวัด

\* TARGET\_PRESSURE ระดับความดันเงินเพื่อเป้าหมายเมื่อพองตัว

\* QUICK\_DEF\_TIME ระดับหน่วยเวลาของภาวะเงินฝืดอย่างรวดเร็วในหน่วย ms

\*

\* ----- ไม่แนะนำให้ใช้พารามิเตอร์ต่อไปนี้ -----

\* \_SAMPLING\_RATE อัตราการสุ่มตัวอย่างต่อวินาที

\* \_STEP\_TIME หน่วย ms สำหรับแต่ละครั้งที่พองตัว

\*/

#กำหนด DEFLATION\_TIME 5

#กำหนด TARGET\_PRESSURE 200

#กำหนด \_SAMPLING\_RATE 100

#กำหนด \_STEP\_TIME 100

#กำหนด QUICK\_DEF\_TIME 20,000

// กำหนดจำนวนจุดข้อมูลที่จะรวบรวม

#กำหนด DATA\_ARRAY\_SIZE (DEFLATION\_TIME\*\_SAMPLING\_RATE)

// ความล่าช้าหลังจากการสุ่มตัวอย่างแต่ละครั้งในรูป

#กำหนด SAMPLING\_DELAY (1000/\_SAMPLING\_RATE)

// เปิดสองอาร์เรย์เพื่อเก็บข้อมูล

int wave\_data[DATA\_ARRAY\_SIZE];

float pressure\_data[DATA\_ARRAY\_SIZE];

/\*

\* คำจำกัดความของจอแสดงผล OLED

\* ตั้งค่าความละเอียดไว้ที่ 128\*64

\* พอร์ตนาฬิกาคือ 17 พอร์ตข้อมูลคือ 16

\* ใช้ I2C

\*/

U8G2\_SSD1306\_128X64\_NONAME\_F\_HW\_I2C u8g2(U8G2\_R3, /\* รีเซ็ต=\*/ U8X8\_PIN\_NONE, /\* นาฬิกา=\*/

17, /\* data=\*/ 16); // ESP32 Thing, HW I2C พร้อมปรับแบบ

/\*

\* กำหนด ADC

\* ใช้ I2C

\* 16 บิต

\*/

โฆษณา Adafruit\_ADS1115 ;

/\*

\* กำหนดพอร์ตอินพุตของ ADC

\* พอร์ต ADC 0 เป็นรูปคลื่น

\* พอร์ต ADC 3 คือแรงดัน

\*/

const int WAVE\_SIG = 0;

const int PRES\_SIG = 3;

```
/*
```

```
 * ความหมายของบอร์ดขับเคลื่อนเตอร์
```

```
 * สี่พอร์ตอินพุตของบอร์ดไมโครคอนโทรลเลอร์
```

```
*/
```

```
int drive_input1 = 0;
```

```
int drive_input2 = 2;
```

```
int drive_input3 = 15;
```

```
int drive_input4 = 13;
```

```
เป็นโหม้ OLED_display_default(){
```

```
    u8g2.clearBuffer();          // ล้างหน่วยความจำภายใน
```

```
    drawDateTime(0, 0, 0, 0);
```

```
    วาดBP(0,0);
```

```
    u8g2.sendBuffer();          // ถ่ายโอนหน่วยความจำภายในไปยังจอแสดงผล
```

```
    ลำค่า (1,000);
```

```
}
```

```
/*
```

```
 * ฟังก์ชันการแสดงผล OLED
```

```
 * ป้อนพารามิเตอร์ int 6 ตัว ได้แก่ เดือน วัน ชั่วโมง นาที
```

```
 * นอกจากนี้ systolic และ diastolic
```

```
*/
```

```
เป็นโหม้ OLED_display (int _mon, int _day, int _hour, int _min,
```

```
        int systolic, int diastolic){

    u8g2.clearBuffer();          // ล้างหน่วยความจำภายใน

    drawDateTime(_mon, _day, _hour, _min);

    drawBP(systolic, diastolic);

    u8g2.sendBuffer();          // ถ่ายโอนหน่วยความจำภายในไปยังจอแสดงผล

    ล่าช้า (2543);

}

/*

* เดือน วันที่ ชั่วโมง นาทีที่ผ่านเป็น int

* แสดงอยู่ด้านบนของจอแสดงผล

* สกปรกมาก ห้ามดัดแปลง

*/

ถือเป็นโมฆะ drawDateTime (int _mon, int _day, int _hour, int _min){

    int xPosition = 0;

    สตริง tempStr = "";

    u8g2.setFont(u8g2_font_ncenB08_tr); // เลือกแบบอักษรที่เหมาะสม

    tempStr = tempStr + _mon + "-" + _day;

    xPosition = 1;

    u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()-2,tempStr.c_str()); // เขียนบางอย่างไปที่
หน่วยความจำภายใน

    tempStr = "";

    tempStr = tempStr + _hour + ":" + _min;
```

```
xPosition = 63-u8g2.getStrWidth(tempStr.c_str());

u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()-2,tempStr.c_str()); // เขียนบางอย่างไปที่
หน่วยความจำภายใน

u8g2.setDrawColor(2);

u8g2.drawBox(0,0,128,u8g2.getMaxCharHeight()-1);

}

/*

* ฟังก์ชันแสดงความดันโลหิตผ่านสอง int (ความดัน systolic, ความดัน diastolic)

* สกปรกมาก ห้ามดัดแปลง

*/

ถือเป็นโมฆะ drawBP (int h, int l){

    int ชดเชย = 9;

    int xPosition = 0;

    สตริง tempStr = "";

    u8g2.setFont(u8g2_font_helvR08_tr); // เลือกแบบอักษรที่เหมาะสม

    tempStr = "ซิสโตลิก";

    xPosition = 0;

    u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไปที่
หน่วยความจำภายใน

    ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต-2;

    u8g2.setFont(u8g2_font_helvR24_tn ); // เลือกแบบอักษรที่เหมาะสม
```

```
tempStr = "ชั่วโมง";
```

```
xPosition = 63-u8g2.getStrWidth(tempStr.c_str());
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไปที่  
หน่วยความจำภายใน
```

```
ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต;
```

```
u8g2.setFont(u8g2_font_ncenB08_tr); // เลือกแบบอักษรที่เหมาะสม
```

```
tempStr = "mmHg";
```

```
xPosition = 63-u8g2.getStrWidth(tempStr.c_str());
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไปที่  
หน่วยความจำภายใน
```

```
ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต+5;
```

```
u8g2.setFont(u8g2_font_helvR08_tr); // เลือกแบบอักษรที่เหมาะสม
```

```
tempStr = "ไดแอตโกลิก";
```

```
xPosition = 0;
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไปที่  
หน่วยความจำภายใน
```

```
ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต-2;
```

```
u8g2.setFont(u8g2_font_helvR24_tn ); // เลือกแบบอักษรที่เหมาะสม
```

```
tempStr = "I";
```

```
xPosition = 63-u8g2.getStrWidth(tempStr.c_str());
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไปที่  
หน่วยความจำภายใน
```

```
    ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต;

    u8g2.setFont(u8g2_font_ncenB08_tr); // เลือกแบบอักษรที่เหมาะสม

    tempStr = "mmHg";

    xPosition = 63-u8g2.getStrWidth(tempStr.c_str());

    u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางออกไปที่
หน่วยความจำภายใน

    u8g2.drawBox(0,125,20,3);

    u8g2.drawBox(21,125,20,3);

    u8g2.drawBox(42,125,20,3);

    u8g2.setDrawColor(2);

    u8g2.drawLine(1,126,18);

    u8g2.drawLine(43,126,18);

}

/*

* แสดงอัตราการเต้นของหัวใจ

* สกปรกมาก ห้ามดัดแปลง

*/

ถือเป็นโหมด drawHR(){

    int ชดเชย = 9;

    int xPosition = 0;

    สตริง tempStr = "";
```



```
u8g2.setFont(u8g2_font_helvR08_tr); // เลือกแบบอักษรที่เหมาะสม
```

```
tempStr = "อัตราการเต้นของหัวใจ";
```

```
xPosition = 0;
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไม่ที่  
หน่วยความจำภายใน
```

```
ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต-2;
```

```
u8g2.setFont(u8g2_font_helvR24_tn); // เลือกแบบอักษรที่เหมาะสม
```

```
tempStr = "83";
```

```
xPosition = 63-u8g2.getStrWidth(tempStr.c_str());
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไม่ที่  
หน่วยความจำภายใน
```

```
ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต;
```

```
u8g2.setFont(u8g2_font_ncenB08_tr); // เลือกแบบอักษรที่เหมาะสม
```

```
tempStr = "/นาฬิกา";
```

```
xPosition = 63-u8g2.getStrWidth(tempStr.c_str());
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางอย่างไม่ที่  
หน่วยความจำภายใน
```

```
u8g2.drawBox(0,125,20,3);
```

```
u8g2.drawBox(21,125,20,3);
```

```
u8g2.drawBox(42,125,20,3);
```

```
u8g2.setDrawColor(2);
```

```
u8g2.drawLine(1,126,18);
```

```
u8g2.drawLine(22,126,18);

}

/*

*สกปรกมาก ห้ามแก้ไข

*/

เป็นไหมะ drawHistory () {

    int ชดเชย = 9;

    int xPosition = 0;

    สตริง tempStr = "";

    u8g2.setFont(u8g2_font_helvR08_tr); // เลือกแบบอักษรที่เหมาะสม

    tempStr = "ประวัติ";

    xPosition = 0;

    u8g2.drawString(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนนางอย่างไปที่
หน่วยความจำภายใน

    ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต-2;

    u8g2.drawLine(0,offset+5,64);

    u8g2.setFont(u8g2_font_5x7_tf); // เลือกแบบอักษรที่เหมาะสม

    ออฟเซต = ออฟเซต +7;

    สำหรับ (int i=0;i<5;i++){

        tempStr = "06-25 19:44";

        xPosition = 0;
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางออกไปที่
```

หน่วยความจำภายใน

```
ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต+1;
```

```
tempStr = "188/88 120";
```

```
xPosition = 63-u8g2.getStrWidth(tempStr.c_str());
```

```
u8g2.drawStr(xPosition,u8g2.getMaxCharHeight()+offset,tempStr.c_str()); // เขียนบางออกไปที่
```

หน่วยความจำภายใน

```
ออฟเซต = u8g2.getMaxCharHeight()+ออฟเซต-2;
```

```
u8g2.drawLine(0,offset+5,64);
```

```
ออฟเซต = ออฟเซต +7;
```

```
}
```

```
u8g2.drawBox(0,125,20,3);
```

```
u8g2.drawBox(22,125,20,3);
```

```
u8g2.drawBox(44,125,20,3);
```

```
u8g2.setDrawColor(2);
```

```
u8g2.drawLine(45,126,18);
```

```
u8g2.drawLine(23,126,18);
```

```
}
```

```
/*
```

\* ป้อนค่าความดันและรูปคลื่น

\* เอาต์พุตอนุกรม "ความดัน รูปคลื่น\n"

\*/

เป็นโมฆะ serial\_display\_pressure\_and\_wave (float pres, int wave){

Serial.print("\n");

อนุกรม. พิมพ์(wps);

Serial.print(",");

อนุกรม พิมพ์ (คลื่น);

}

/\*

\* ฟังก์ชันนี้ใช้เพื่อจัดการกับปัญหาหลักของสัญญาณรูปคลื่น และเพื่อซูม คงที่ประมาณ 100-150

\* ปัญหา: สองสามวินาทีแรกจะมีขนาดใหญ่มากและล้น

\* ลบสองสามวินาทีแรกและค่า \_MAGIC\_k แรก

\* ชุมออก \_MAGIC\_NUM ครั้ง

\*/

int pure\_magic (คลื่น int, ลำดับ int){

int \_MAGIC\_NUM = 50;

อุณหภูมิ int = 60;

อุณหภูมิ = คลื่น/\_MAGIC\_NUM;

ถ้า (อุณหภูมิ>200 && คำสั่ง<150){

กลับ 60;

}

อื่น {

ถ้า (อุณหภูมิ == 1310){

```
        อุณหภูมิ = 46;

    }

    อุณหภูมิกลับ

}

}

/*

* ฟังก์ชันพอง

* อัตราเงินเฟ้อเพื่อเวลาบิลลีนาที้

*/

โน้ต: inflate_for_x_ms (เวลา int){

    openM1_closeM2();

    เวลาล่าช้า);

}

/*

* ฟังก์ชันยุบซ้ำ

*/

เป็นโน้ต:ยุบ (){

    ปิดM1_closeM2();

}

/*

* ฟังก์ชันยุบอย่างรวดเร็ว

* ระยะเวลาคงค้างออกภาวะเย็นผิดปกติอย่างรวดเร็วเป็นบิลลีนาที้
```

```
*/
```

```
ถือเป็นโมฆะ quick_deflate_for_x_ms (เวลา int){
```

```
เปิดM1_openM2();
```

```
เวลาล่าช้า);
```

```
}
```

```
/*
```

```
* เปิดมอเตอร์ M1 ระหว่าง OUT1 และ OUT2 ในโครงการนี้มอเตอร์เฟือง
```

```
* ปิดมอเตอร์ M2 ระหว่าง OUT3 และ OUT4 ในโครงการนี้ว่าส่วไล่ม
```

```
*/
```

```
ถือเป็นโมฆะ openM1_closeM2(){
```

```
digitalWrite(drive_input1,สูง);
```

```
digitalWrite (drive_input2, ต่ำ);
```

```
digitalWrite (drive_input3, ต่ำ);
```

```
digitalWrite (drive_input4, ต่ำ);
```

```
}
```

```
/*
```

```
* ปิดมอเตอร์ M1 ระหว่าง OUT1 และ OUT2 ในโครงการนี้มอเตอร์เฟือง
```

```
* ปิดมอเตอร์ M2 ระหว่าง OUT3 และ OUT4 ในโครงการนี้ว่าส่วไล่ม
```

```
*/
```

```
เป็นโมฆะ closeM1_closeM2(){
```

```
digitalWrite (drive_input1, ต่ำ);
```

```
digitalWrite (drive_input2, ต่ำ);

digitalWrite (drive_input3, ต่ำ);

digitalWrite (drive_input4, ต่ำ);

}

/*

* ปิดมอเตอร์ M1 ระหว่าง OUT1 และ OUT2 ในโครงการนี้มอเตอร์เข้าลบ

* เปิดมอเตอร์ M2 ระหว่าง OUT3 และ OUT4 ในโครงการนี้ว่าสวิตช์ล้น

*/

เป็นโหม้: closeM1_openM2(){

digitalWrite (drive_input1, ต่ำ);

digitalWrite (drive_input2, ต่ำ);

digitalWrite(drive_input3,สูง);

digitalWrite (drive_input4, ต่ำ);

}

/*

* แปลงอินพุต int ของ ADC เป็นสัญญาณแรงดันไฟฟ้า

* อินพุต: int

* เอาต์พุต: ลอย

*/

ลอย transfer_adc2v (int n){

ลอย t = (n * 0.1875)/1,000;
```

```
        กลับ t;

    }

/*

* แปลงสัญญาณแรงดันของเซ็นเซอร์แรงดันเป็นสัญญาณแรงดันในหน่วย mmHg

* อินพุต: ลอย

* เอาต์พุต: ลอย

*/

float transfer_v2p (float v){

    float v1 = v/5.0;

    ลอย v2 = v1-0.04;

    kpa ลอย = v2/0.018;

    float mmhg = 7.5*kpa;

    กลับ mmhg;

}

/*

* รับอินพุตสองอาร์เรย์: อาร์เรย์รูปคลื่นและอาร์เรย์แรงดัน

* แรงดันเฉลี่ยขาออก

*/

ค่าเฉลี่ยความดันลอย (ค่าลอยตัว [], คลื่น int [], ขนาด size_t){

    int max_wave = 0;

    ลอย avg_pres = 0;

    สำหรับ (ตัวนับ int = 0; ตัวนับ <ขนาด; ตัวนับ ++){
```



```
    ถ้า (คลื่น [เคาน์เตอร์]>max_wave){  
  
        max_wave = คลื่น [เคาน์เตอร์];  
  
        avg_pres = ส่วนหน้า [เคาน์เตอร์];  
  
    }  
  
}  
  
Serial.printf("THE average_pressure IS -----%f", avg_pres);  
  
    กลับ avg_pres;  
  
}
```

```
ลอย systolic_pressure (ลอย pres [], คลื่น int [], size_t arr_size){
```

```
    int wave_peak[50];
```

```
    ลอย pres_when_wave_peak[50];
```

```
    int peak_num = 0;
```

```
    // เริ่มต้นเป็น 0
```

```
    สำหรับ (int i = 0;i<50;i++) wave_peak[0] =0;
```

```
    สำหรับ (int i = 0;i<50;i++) pres_when_wave_peak[0] =0;
```

```
    int before2=0, before1=0, mid=0, after1=0, after2=0;
```

```
    // บันทึกจุดสูงสุดใน wave_peak
```

```
    สำหรับ (int i=5;i<(arr_size-5);i++){
```

```
        before2 = คลื่น [i-2];
```

```
        before1 = คลื่น [i-1];
```

```
        กลาง = คลื่น [i];
```

```
        after1 = คลื่น[i+1];
```

```
        after2 = คลื่น[i+2];
```

```
    // เพิ่มขึ้นที่ด้านหน้าและลดลงที่ด้านหลัง
```

```
        ถ้า((ก่อน1>ก่อน2)&&(กลาง>ก่อน1)&&
            (หลัง 1<กลาง)&& (หลัง 2 <หลัง 1)
            &&peak_num<50 && กลาง>60){

                wave_peak[peak_num] = กลาง;

                pres_when_wave_peak[peak_num] = ล่วงหน้า[i];

                peak_num++;

            }
    }

    // หาจุดพีคสูงสุด

    int peak_max = 0;

    อุณหภูมิ int = 0;

    สำหรับ (int i=0;i<50;i++){

        อุณหภูมิ = wave_peak [i];

        ถ้า (อุณหภูมิ>peak_max)

            peak_max = อุณหภูมิ;

    }

    int calced_sys_wave = peak_max*0.7;

    int calced_dia_wave = peak_max*0.55;

    กลับ 0;

}
```

```
/*

    ลอย send_data_array (int sys, int dia){

        WiFiโมดูลไฟฟ้า;

        // การตั้งค่า wifi

        สำหรับ (uint8_t t = 4; t > 0; t--) {

            Serial.printf("[ตั้งค่า WIFI] รอบ %d...\n", t);

            อุ่นระบบ. flush();

            ล่าช้า (1,000);

        }

        wifiMulti.addAP("ฮอตสปอต", "Whatapwd");

        ในขณะที่(wifiMulti.run() != WL_CONNECTED){

            Serial.println("กำลังพยายามเชื่อมต่อ");

        }

        Serial.println("เชื่อมต่อแล้ว");

        Serial.println("เริ่มส่งอาร์เรย์ทั้งหมด");

        HTTPClient http;

        http.begin(SERVER_ADDR);

        http.POST(String(sys)+"", "+String(dia));

        http.end();

    }

*/
```

```
การตั้งค่าเป็นโมดูล (เป็นโมดูล) {

    // การตั้งค่าการแสดงผล

    u8g2.begin(0,23,2,3,4,5);
```

```
// การตั้งค่าซีเรียล

Serial.begin(115200);

// การตั้งค่า ADC

ads.begin();

// การตั้งค่าของบอร์ดไมโครคอนโทรลเลอร์

โหมดพิน (drive_input1, OUTPUT);

โหมดพิน (drive_input2, OUTPUT);

โหมดพิน (drive_input3, OUTPUT);

โหมดพิน (drive_input4, OUTPUT);

}

วนเป็นโมฆะ (โมฆะ) {

// ส่งอินพุตของ ADC

int16_t adc_wave, adc_pressure;

// อ่านสัญญาณแรงดันจาก adc

adc_pressure = โฆษณา readADC_SingleEnded (PRES_SIG);

/*

* กำหนด pres เป็นค่าความดันในขณะนี้

* ข้อมูลที่แสดงคือ

* "ค่าความดัน,0\๓"

*/

ปล่อย pres = transfer_v2p (transfer_adc2v (adc_pressure));

Serial.println("ทดสอบที่นี้");
```

$$/^{*}$$

\* ฟังก์ชันพหุคูณ

\* หยุดเมื่อพบ TARGET\_PRESSURE

\* \_STEP\_TIME ควบคุมเวลาในการสูบน้ำแต่ละครั้ง หน่วยเป็น ms

\*/

```
ในขณะที (เพรส<TARGET_PRESSURE){
```

```
// อ่านสัญญาณแรงดันจาก adc
```

```
adc_pressure = ໙໙໙ readADC_SingleEnded (PRES_SIG);
```

 $\text{I}^*$ 

\* กำหนด pres เป็นค่าความดันในขณะนี้

\* ข้อมูลที่แสดงคือ

\* "ค่าความดัน, 0\ก"

\* /

คำสั่ง = transfer\_v2p(transfer\_adc2v(adc\_pressure));

```
//Serial.println("ทดสอบ ที่นี้2");
```

```
serial_display_pressure_and_wave(ก่อนหน้า,60);
```

```
inflate_for_x_ms(_STEP_TIME);
```

}

$$/^{*}$$

\* ปิ๊งขึ้นอย่างสมบูรณ์

```
* และทำฟังก์ชันการแสดงผลและฟังก์ชันการบันทึกให้สมบูรณ์หลังจากลูปน้ำเสร็จ

*/

เปิดM1_closeM2();

// รับข้อมูล DATA_ARRAY_SIZE ครั้ง

int magic_adc_wave;

int _adc_wave;

สำหรับ (ตัวนับ int = 0; ตัวนับ < DATA_ARRAY_SIZE; ตัวนับ ++){

    _adc_wave = โหมด readADC_SingleEnded(WAVE_SIG);

    adc_pressure = โหมด readADC_SingleEnded (PRES_SIG);

    คำสั่ง = transfer_v2p(transfer_adc2v(adc_pressure));

    // ประมวลผลสัญญาณคลื่นพัลส์เดิมเพื่อให้ดูดีขึ้น

    magic_adc_wave = pure_magic(_adc_wave,เคาน์เตอร์);

    // เก็บสัญญาณในสองอาร์เรย์

    wave_data[เคาน์เตอร์] = magic_adc_wave;

    pressure_data[เคาน์เตอร์] = แรงกด;

    //แสดงความดันและคลื่น คำนวณด้วยเครื่องหมายจุลภาค แสดงเส้นโค้งสองเส้น

    serial_display_pressure_and_wave(ก่อน,magic_adc_wave);

    ล่าช้า (SAMPLING_DELAY);

}

// รับแรงดันเฉลี่ย

float avg_pres = average_pressure (ความดัน_ข้อมูล, wave_data, DATA_ARRAY_SIZE);

float fake_sys_pres = avg_pres*1.3;

float fake_dia_pres = avg_pres*0.63;

int sys_pres = (int)fake_sys_pres;
```

```
int dia_pres = (int)fake_dia_pres;

Serial.println("sys ความดันคือ      , ความดัน dia คือ \n"+String(fake_sys_pres)+String(fake_dia_pres));

Serial.printf("ความดัน sys คือ %d, ความดัน dia คือ %d\n",sys_pres,dia_pres);

OLED_display(6,30,0,4,sys_pres,dia_pres);

quick_deflate_for_x_ms(QUICK_DEF_TIME);

//send_data_array(sys_pres,dia_pres);

}
```