

ระบบตรวจวัดชีพจร และ อุณหภูมิแบบ Real Time

REAL TIME PULSE & TEMPERATURE MONITORING SYSTEM

Sensors and Transducers



หัวข้อ 1

ปัญหาที่กำาหนดขึ้น และ แนวทางแก้ปัญา

ปัญหาที่พบเจอ



- ยุ่งยาก
- ใช้เวลานานในการวัด
- ใช้เวลานานในการตรวจสอบ
- ตรวจสอบยาก
- ข้อมูลไม่ออนไลน์
- สามารถวัดได้ค่าเดียว
- เกิดภาวะโลกร้อน
- สิ้นเปลืองกระดาษ

การแก้ไขปัญหา



- ใช้งานได้ง่าย
- ได้ค่าผลลัพธ์ที่รวดเร็ว
- ตรวจสอบผ่านทางออนไลน์
- ผู้ดูแลสามารถตรวจสอบได้
- แจ้งเตือน ผ่านทางอุปกรณ์ และออนไลน์ เมื่อถึงขีดอันตราย
- ข้อมูลเป็นระบบออนไลน์
- สามารถวัดค่าได้หลายค่า
- ไม่มีการเปลี่ยนแปลงกระดาศ
- ลดมลพิษ



หัวข้อ 2

อธิบายหลักการพิจารณา เลือกใช้ SENSORS ที่ เหมาะสมกับปัญหา

หลักการพิจารณาเลือกใช้



มองหาปัญหา



ทำการค้นหาข้อมูล



กำหนดแนวทางแก้ไข

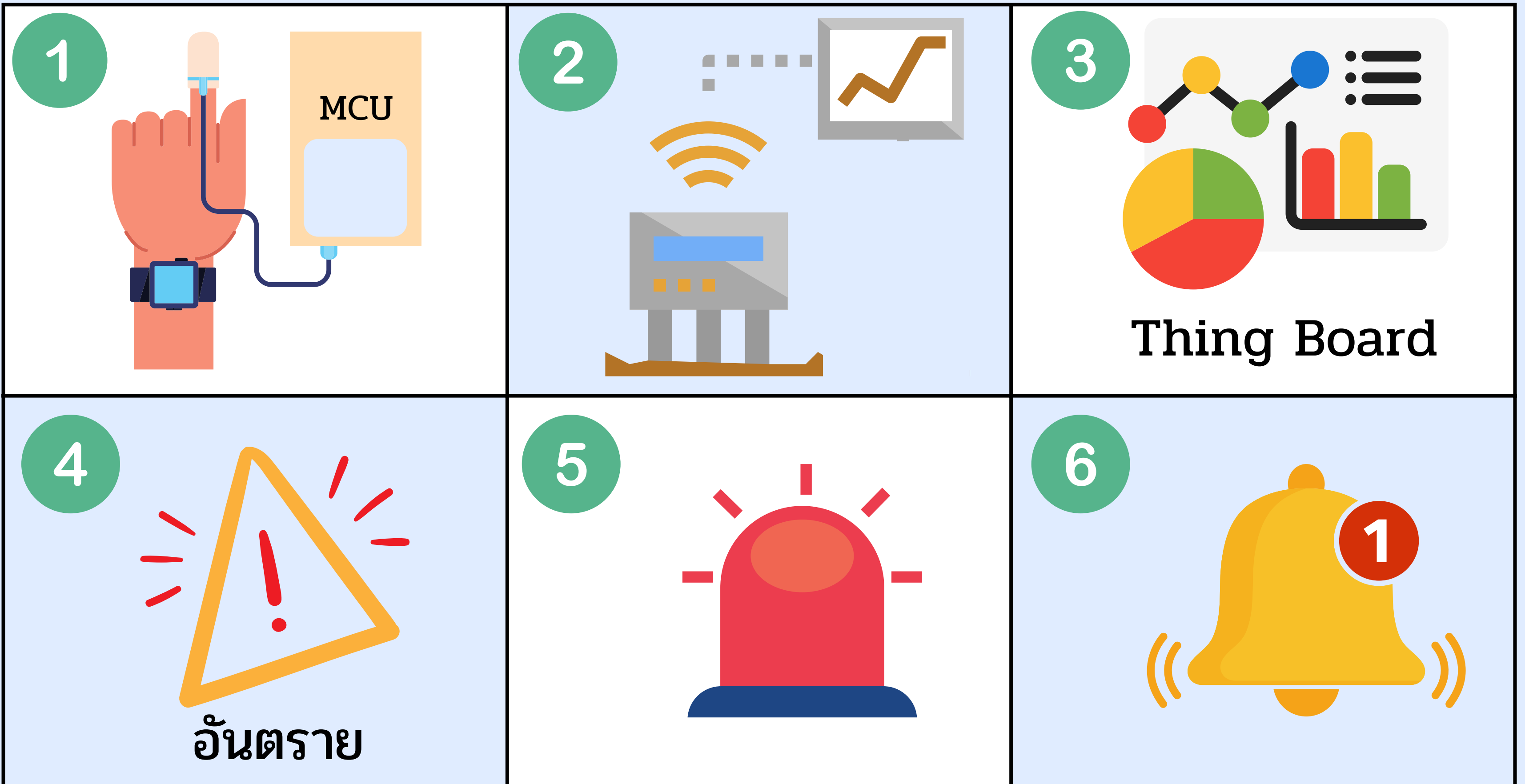


เลือกอุปกรณ์ที่ตรงกับแนวทาง



หัวข้อ 3

ภาพรวมการทำงาน





หัวข้อ 4

อธิบายการทำงาน ส่วนย่อยๆ

IMPORT ไลบารี

```
#include <Wire.h>
#include "DHT.h"
#include <WiFi.h>
#include <ThingsBoard.h>

#include "MAX30105.h"
#include "heartRate.h"
#include <Adafruit_MLX90614.h>
```

กำหนดขาต่างๆ และ WIFI , THINGBOARD



```
//Set pins for I2C2
#define SDA_2 33
#define SCL_2 32
#define LED 23
#define LED2 19
#define WIFI_AP "Thoth"
#define WIFI_PASSWORD "top123456789"
#define TOKEN "ESP32_DEMO_TOKEN"
// DHT
#define DHTPIN 2
#define DHTTYPE DHT22
char thingsboardServer[] = "34.222.203.238";
WiFiClient wifiClient;
// Initialize DHT sensor.
DHT dht(DHTPIN, DHTTYPE);
ThingsBoard tb(wifiClient);
```

กำหนดตัวแปร

```
int status = WL_IDLE_STATUS;
unsigned long lastSend;

const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred

float beatsPerMinute;
int beatAvg;
float temperature;
```

กำหนดช่อง I2C



```
//set I2C default
if (particleSensor.begin(Wire, I2C_SPEED_FAST) == false) //Use default I2C port, 400kHz speed
{
    Serial.println("MAX30102 was not found. Please check wiring/power. ");
    while (1);
}
particleSensor.setup(); //Configure sensor with default settings
particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running
particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
//set I2C 2 use SDA 33 SCL 32 addr 0x5A
bool status2 = mlx.begin(0x5A,&Wire1);
if (!status2) {
    Serial.println("Could not find a valid MLX90614 sensor, check wiring!");
    while (1);
}
```

รับค่า และคำนวณ BPM

```
long irValue = particleSensor.getIR();
temperature = mlx.readObjectTempC();//อ่านค่าอุณหภูมิ
//BPM
if (checkForBeat(irValue) == true)
{
    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();

    beatsPerMinute = 60 / (delta / 1000.0);

    if (beatsPerMinute < 255 && beatsPerMinute > 20)
    {
        rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
        rateSpot %= RATE_SIZE; //Wrap variable

        //Take average of readings
        beatAvg = 0;
        for (byte x = 0 ; x < RATE_SIZE ; x++)
            beatAvg += rates[x];
        beatAvg /= RATE_SIZE;
    }
}
```

แจ้งเตือนผ่าน LED

```
//แจ้งเตือน
if(temperature >= 37.5){
    digitalWrite(LED, HIGH); // Turn the LED on
}else{
    digitalWrite(LED, LOW);
}

if(beatsPerMinute >= 180){
    digitalWrite(LED2, HIGH); // Turn the LED on
}else{
    digitalWrite(LED2, LOW);
}
```

แสดงค่าทาง SERIAL MONITOR

```
//แสดงผลทาง Serial Monitor
Serial.print("Object = ");
Serial.println(temperature);
Serial.print(" IR=");
Serial.print(irValue);
Serial.print(", BPM=");
Serial.print(beatsPerMinute);
Serial.print(", Avg=");
Serial.print(beatAvg);
if (irValue < 50000){
    Serial.print(" No finger?");
    beatsPerMinute =0;
    beatAvg =0;
}
Serial.println();
```


ใช้งานฟังก์ชัน



```
if ( !tb.connected() ) {  
    reconnect();  
}  
if ( millis() - lastSend > 1000 ) { // Update and send only after 1 seconds  
    getAndSendTemperatureAndHumidityData();  
    lastSend = millis();  
}  
tb.loop();
```

ฟังก์ชันส่งข้อมูล



```
void getAndSendTemperatureAndHumidityData()
{
    Serial.println("Collecting temperature data.");
    // Check if any reads failed and exit early (to try again).
    if (isnan(temperature) || isnan(beatsPerMinute) || isnan(beatAvg)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    Serial.println("Sending data to ThingsBoard:");
    tb.sendTelemetryFloat("temperature", temperature);
    tb.sendTelemetryFloat("beatsPerMinute", beatsPerMinute);
    tb.sendTelemetryFloat("beatAvg", beatAvg);
}
```

ฟังก์ชันเชื่อมต่อ WIFI

```
void InitWiFi()
{
    Serial.println("Connecting to AP ...");
    // attempt to connect to WiFi network

    WiFi.begin(WIFI_AP, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("Connected to AP");
}
```

ฟังก์ชันเชื่อมต่อ THING BOARD

```
void reconnect() {
  // Loop until we're reconnected
  while (!tb.connected()) {
    status = WiFi.status();
    if ( status != WL_CONNECTED) {
      WiFi.begin(WIFI_AP, WIFI_PASSWORD);
      while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
      }
      Serial.println("Connected to AP");
    }
    Serial.print("Connecting to ThingsBoard node ...");
    if ( tb.connect(thingsboardServer, TOKEN) ) {
      Serial.println( "[DONE]" );
    } else {
      Serial.print( "[FAILED]" );
      Serial.println( " : retrying in 5 seconds" );
      // Wait 5 seconds before retrying
      delay( 5000 );
    }
  }
}
```



สรุปผล



ตรงตามจุดประสงค์



มีความแม่นยำ



ทำงานได้อย่างถูกต้อง

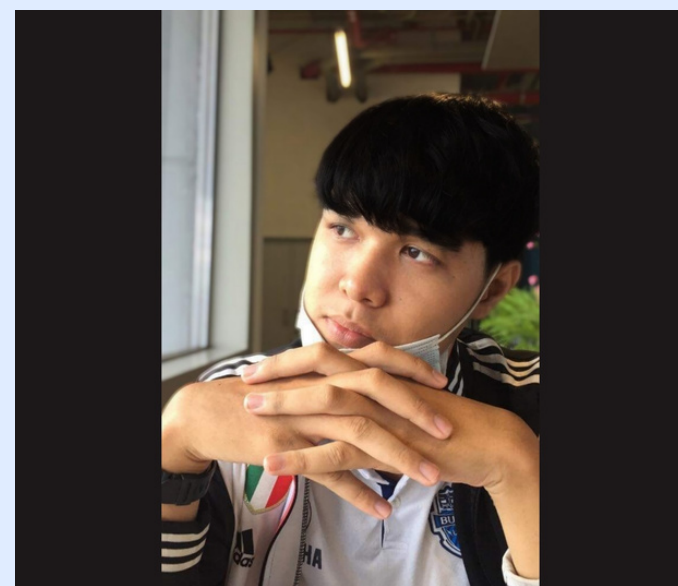


ได้ผลลัพธ์ตามที่ต้องการ



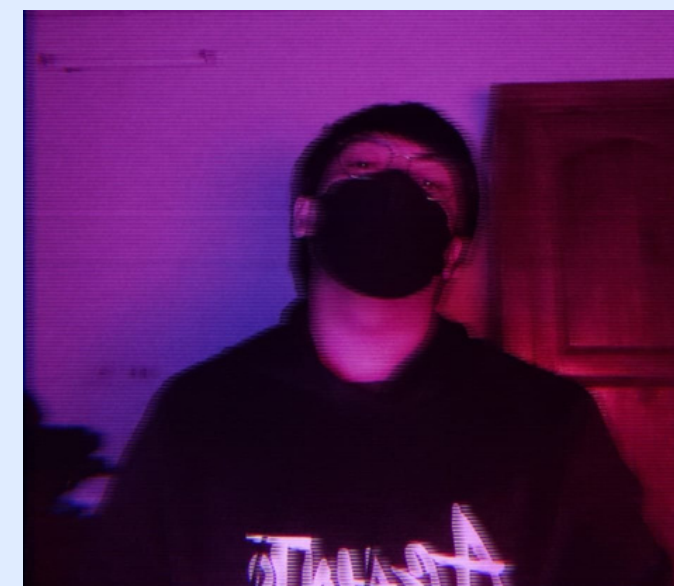
จัดทำโดย

Real time pulse and temperature
monitoring system



ธนพล ชำนาญหาญ

163404140004



คณินศักดิ์ สุจารีย์

163404140001

THANK YOU

ขอบคุณครับ