

قياس درجة حرارة الانسان دون تلامس والنبض

non-contact human body thermometer and heart beat

احمد عادل مصطفى الشيمي
فهيم حسين سليمان
حسام عبدالعليم علي علي
عمر عادل عبدالمجيد القزاز
محمود عبده عبدالسلام
حمدي سيد عبدالجواد
شريفة قطب عبدالعال
فاطمة عبدالله القطري
ايه قدرى محمد النواجي
سارة جابر عباس

مادة القياسات

—

د.م/ محسن البنداري

مقدمة

المؤشرات الحيوية للانسان

هي إحصاءات فسيولوجية مختلفة يتم قياسها واستخدامها في المجال الطبي من أجل تقييم وظائف الجسم الهامة. في العلامات الحيوية يتم عادة تسجيل درجة حرارة الجسم، معدل النبض (معدل ضربات القلب) ، ضغط الدم ، ومعدل التنفس ويمكن أن تشمل قياسات أخرى على حسب متطلبات التشخيص والمرض المتوقع وتختلف القياسات غالباً باختلاف العمر. وتستخدم العلامات الحيوية لقياس الوظائف الأساسية في الجسم. يتم أخذ هذه القياسات للمساعدة في تقييم الصحة البدنية العامة للشخص، وإعطاء أدلة على الأمراض المحتملة، وتظهر التقدم نحو الانتعاش . النطاقات الطبيعية للعلامات الحيوية الشخص تختلف مع تقدم العمر والوزن والجنس، والصحة العامة.

حساس قياس معدل ضربات القلب

تُعتبر مراقبة معدل ضربات القلب أمراً مهماً للغاية بالنسبة للمرضى والرياضيين، لأنه يشخص حالة القلب الصحية. ويوجد العديد من الطرق لقياس معدل نبضات القلب وأكثرها دقةً هو استخدام جهاز تخطيط القلب الكهربائي (ECG) Electrocardiography، ولكن الطريقة الأسهل هي استخدام حساس نبضات القلب والذي يعتمد على تقنية مخطط التمثيل الضوئي Photoplethysmogram (PPG) هو تقنية بصرية بسيطة تستخدم لاكتشاف التغيرات الحجمية في الشرايين يستخدم PPG ضوء الأشعة تحت الحمراء منخفض الكثافة (IR). عندما ينتقل الضوء عبر الأنسجة البيولوجية ، يتم امتصاصه عن طريق العظام وأصابع الجلد والدم الوريدي والشرياني. نظراً لأن الدم يمتص الضوء بقوة أكبر من الأنسجة المحيطة ، يمكن اكتشاف التغيرات في تدفق الدم بواسطة مستشعرات PPG كتغيرات في شدة الضوء. تتناسب إشارة الجهد من PPG مع كمية الدم المتدفقة عبر الأوعية الدموية. حتى التغيرات الصغيرة في حجم الدم يمكن اكتشافها باستخدام هذه الطريقة ، على الرغم من أنه لا يمكن استخدامها لتحديد كمية الدم. تحتوي إشارة PPG على العديد من المكونات بما في ذلك التغيرات الحجمية في الدم الشرياني المرتبط بالنشاط القلبي ، والتغيرات في حجم الدم الوريدي الذي يعدل إشارة PPG ، ومكون آخر الذي يُظهر الخاصية البصرية للأنسجة ، وتغيرات طفيفة في الطاقة في الجسم. بعض العوامل الرئيسية التي تؤثر على التسجيلات من PPG هي موقع القياس وقوة الاتصال بين الموقع وجهاز الاستشعار. تحدث تغيرات تدفق الدم في الغالب في الشرايين وليس في الأوردة.

مقدمة

قياس درجة حرارة الانسان بدون تلامس

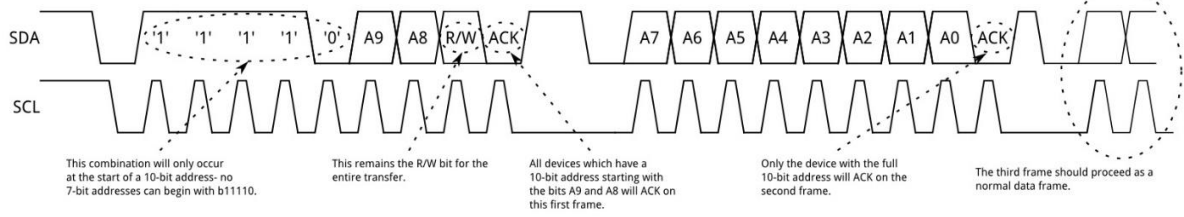
تعتمد فكره درجة الحرارة على المزدوج الحراري وكما نعلم فان المزدوج الحراري يقوم بتحويل كمية الحرارة الواقعه علي وصلتي المزدوج الي جهد كهربى ولكن الامر يختلف عندما يكون مع الانسان حيث يتم تركيز طاقة الاشعه تحت الحمراء المنبعثة من الجسم على اجهزه الكشف الضوئية وتقوم هذه الاجهزه بتحويل الطاقة الناتجة من الأشعة تحت الحمراء إلى إشارة كهربية ونظرا لأن طاقة الأشعة تحت الحمراء المنبعثة لأي جسم تتناسب مع درجة حرارته فإن الإشارة الكهربائية توفر قراءة دقيقة لدرجة حرارة الجسم الموجه إليه ويوفر مستشعر الأشعة تحت الحمراء تفاصيل اكبر أثناء القياس مقارنة بأجهزة الاتصال وذلك عن طريق توجيهه إلى نقاط مختلفة على الكائن الحي الذي تتم قراءته، ونظرا لان المزدوجات الحرارية لا يكون لها دخل فان اشارة الخرج تكون صغيره جدا حيث يتم استخدام مكبرات لتكبير جهد خرج المزدوج الحراري حيث احيانا يصل معامل التكبير الخاص بالجهد الي 1000 نظرا لصغر اشارة خرج المزدوج الحراري.

البروتوكول المستخدم في عملية التواصل بين الحساسات والمتحكم

تعتمد فكره المشروع علي استخدام العديد من الاجهزه المختلفه منها شاشه للعرض و حساس لقياس درجة الحرارة و وحساس لقياس النبض وعند اختيارنا لجميع هذه الاجهزه حرصنا علي جعل هذه الاجهزه كلها ان تستخدم نفس بروتوكول الاتصال لكي يسهل علينا عملية البرمجة والاقتصاديه في اختيار نوع المتحكم. قمنا باستخدام بروتوكول I2C وما يميز ذلك البروتوكول عن غيره انه يستخدم خطي فقط لربط الاجهزة مع بعضها البعض، ما يميز ذلك البروتوكول ايضا انه يدعم ربط اكثر من جهاز علي نفس الخط ويعطي لكل جهاز عنوان فزيائي لا يتكرر مع باقي الاجهزه ويمكن اضافته حوالي 127 جهاز علي نفس الخط، ويتكون هذا البروتوكول من خطين الاول وهو SCL وهو خاص بنبضه الساعه وتحديد سرعه نقل البيانات والثاني SDA وهو الخاص بنقل البيانات، وما يميز ذلك البروتوكول ايضا عن غيره يمكن للاجهزه ارسال واستقبال بيانات وليس ارسال فقط او استقبال فقط ويتميز ايضا بوجود بت خاص بـ ما يسمى ACK وهذه البت ترسل بعد ارسال كل من 8 بتات بيانات لتؤكد هل تم استلامهم بنجاح ام لاء وبذلك يمنع حدوث اخطاء او فقدان في نقل البيانات.

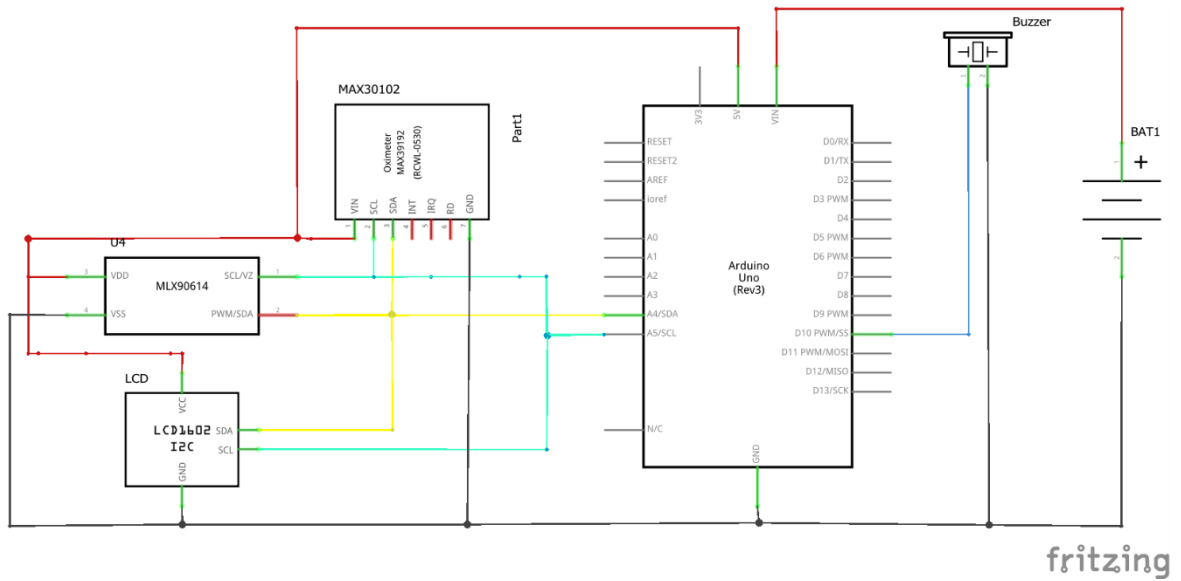
مقدمة

اما عن تحديد الجهاز المراد ارسال استقبال منه البيانات او ارسالها فاننا نقوم بارسال في المقدمة 7 بتات لتحديد العنوان الخاص بالجهاز من ثم ارسال البت الثامن لتحديد وضع الجهاز في وضع قراءة ام كتابه.



حيث توضح الصورة السابقة شكل ل حزمه البيانات الخاصه بالبروتوكول التي يتم ارسالها .

مخطط الدائرة



```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "MAX30105.h"
#include "heartRate.h"
#include <Adafruit_MLX90614.h>

#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line display
MAX30105 particleSensor;
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

const int buzzer = 10;
uint8_t heart[8] = {0x0, 0xa, 0x1f, 0x1f, 0xe, 0x4, 0x0};
int oldHeart = 0;
float oldTemp = 0;
bool rwMood = false;
int rwMoodCount = 0;

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;

int countAvg = 0;
int oldavg = 0;
int countDone = 0;

int sampleAvgComp = 5;
float maxTempWorng = 37;
int rTime = 10000;
int wrongTime = 10000;

void setup() {
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
  Serial.begin(115200);
  Serial.println("Initializing...");
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
  // Serial.println("Place your index finger on the sensor with steady pressure.");
  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

  mlx.begin();

  lcd.init();
  lcd.backlight();
  lcd.createChar(0, heart);
  lcd.setCursor(0, 0);
  lcd.print("This Project by");
  lcd.setCursor(0, 1);
  lcd.print("AE Team");
  delay(10000);

```

```

    lcd.clear();
}

void loop() {
    countDone = 0;
    long irValue = particleSensor.getIR();

    if (checkForBeat(irValue) == true)
    {
        //We sensed a beat!
        long delta = millis() - lastBeat;
        lastBeat = millis();
        beatsPerMinute = 60 / (delta / 1000.0);

        if (beatsPerMinute < 255 && beatsPerMinute > 35)
        {
            rates[ratesSpot++] = (byte)beatsPerMinute; //Store this reading in the array
            ratesSpot %= RATE_SIZE; //Wrap variable
            //Take average of readings
            beatAvg = 0;
            for (byte x = 0 ; x < RATE_SIZE ; x++)
                beatAvg += rates[x];
            beatAvg /= RATE_SIZE;
        }
    }

    Serial.println("-----");
    Serial.print("IR=");
    Serial.print(irValue);
    Serial.print(", BPM=");
    Serial.print(beatsPerMinute);
    // Serial.print("\t");
    Serial.print(", Avg BPM=");
    Serial.println(beatAvg);
    //
    Serial.print("oldavg=");
    Serial.print(oldavg);
    Serial.print(", countAvg=");
    Serial.println(countAvg);

    if (irValue < 50000) {
        Serial.print("No finger?");
        if ((oldHeart || oldTemp != 0) && countDone == 0 ) {
            lastResult(oldHeart, oldTemp);
        }
        noFing();
        rwMood = false;
        rwMoodCount = 0;
    } else {
        rwMood = true;
        rwMoodCount = 1;
        lcdWrite(beatAvg, mlx.readObjectTempC());
    }

    if (countAvg <= sampleAvgComp) {
        if ((oldHeart || oldTemp != 0) && countAvg == sampleAvgComp) {
            lastResult(oldHeart, oldTemp);
            countDone = 1;
        }
        if (oldavg != beatAvg) {
            oldavg = beatAvg;
            countAvg++;
        }
    }
    Serial.print(", countDone=");
}

```

```

    Serial.print(countDone);
    Serial.println();
}

void lcdWrite(int heart, float temp) {
    if (rwMoodCount == 1 ) {

        lcdScreenInit();
        rwMoodCount = 0;
    }

    if (oldHeart != heart) {
        lcd.setCursor(5, 0);
        lcd.print(" ");
        lcd.setCursor(5, 0);
        lcd.print(heart);
        oldHeart = heart;
    }

    if (oldTemp != temp) {
        lcd.setCursor(5, 1);
        lcd.print(" ");
        lcd.setCursor(5, 1);
        lcd.print(temp);
        oldTemp = temp;
    }
}

void noFing() {
    lcd.setCursor(0, 0);
    lcd.print("Please Put Your");
    lcd.setCursor(0, 1);
    lcd.print("Finger");
    delay(1000);
    lcd.noDisplay();
    delay(500);
    lcd.display();
    lcd.clear();
}

void lcdScreenInit() {
    lcd.setCursor(0, 0);
    lcd.print("Avg");
    lcd.printByte(0);
    lcd.setCursor(4, 0);
    lcd.print("=");
    lcd.setCursor(0, 1);
    lcd.print("Temp=");
}

void lastResalt(int heart, float temp) {
    bebbeb();
    lcd.setCursor(0, 0);
    lcd.print("Avg");
    lcd.printByte(0);
    lcd.setCursor(4, 0);
    lcd.print("=");
    lcd.setCursor(0, 1);
    lcd.print("Temp=");
    lcd.setCursor(15, 0);
    lcd.print("R");
    lcdWrite(heart, temp);

    if (temp >= maxTempWorng) {
        digitalWrite(buzzer, HIGH);
    }
}

```

```

        delay(wrongTime);
    }

    delay(rTime);
    lcd.clear();
    oldTemp = 0;
    oldHeart = 0;
    oldavg = 0;
    beatAvg = 0;
    beatsPerMinute = 0;
    countAvg = 0;
    digitalWrite(buzzer, LOW);
}

void bebbeb() {
    digitalWrite(buzzer, HIGH);
    delay(300);
    digitalWrite(buzzer, LOW);
    delay(150);
    digitalWrite(buzzer, HIGH);
    delay(300);
    digitalWrite(buzzer, LOW);
}

```