

```

#include "Wire.h"
#include <CapSense.h>

#define HRMI_I2C_ADDR    127
#define HRMI_HR_ALG      1 // 1= average sample, 0 = raw sample

////////////////////////////////////

CapSense  cs_4_2 = CapSense(4,2); // 10 megohm resistor between pins 4 & 2,
pin 2 is sensor pin, add wire, foil
CapSense  cs_4_3 = CapSense(4,3);
CapSense  cs_4_5 = CapSense(4,5);
CapSense  cs_4_6 = CapSense(4,6);

int ledPin = 11;
int ledTouch = 12;
int smooth;
long time = 0;
int ledStatePIN = LOW;
//int ledStateTOUCH = LOW;
long previousMillis = 0;

////////////////////////////////////

void setup(){

  setupHeartMonitor(HRMI_HR_ALG);

  cs_4_2.set_CS_Autocal_Millis(0xFFFFFFFF); // turn off autocalibrate on
channel 1 - just as an example

  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
  pinMode(ledTouch, OUTPUT);
}

////////////////////////////////////

void loop(){

  int heartRate = getHeartRate();

  long start = millis();
  long total1 = cs_4_2.capSense(10);
  long total2 = cs_4_3.capSense(10);
  long total3 = cs_4_5.capSense(10);
  long total4 = cs_4_6.capSense(10);

```

```

long valeur1 = total1 + 1;
String touch1 = (String) valeur1;

long valeur2 = total2 + 1;
String touch2 = (String) valeur2;

long valeur3 = total3 + 1;
String touch3 = (String) valeur3;

long valeur4 = total4 + 1;
String touch4 = (String) valeur4;

unsigned long currentMillis = millis();

// Aucun signal cardiaque
if(heartRate == 0){
  analogWrite(ledPin, 255);
  Serial.print("777");
  Serial.print(",");
}

// 1er interval cardiaque
if(heartRate>=55 && heartRate<=72){
  int periode = 1500;
  time = millis();
  smooth = 128+127*cos(2*PI/periode*time);
  analogWrite(ledPin, smooth);
  Serial.print(heartRate);
  Serial.print(",");

  /*long interval = 800;
  if(currentMillis - previousMillis > interval) {
    previousMillis = currentMillis;
    if (ledStatePIN == LOW)
      ledStatePIN = HIGH;
    else
      ledStatePIN = LOW;
    digitalWrite(ledPin, ledStatePIN);
  }*/
}

// 2eme interval cardiaque
if(heartRate>72 && heartRate<=89){
  int periode = 1250;
  time = millis();
  smooth = 128+127*cos(2*PI/periode*time);
  analogWrite(ledPin, smooth);
  Serial.print(heartRate);
  Serial.print(",");
}

```

```

    /*long interval = 600;
if(currentMillis - previousMillis > interval) {
    previousMillis = currentMillis;
    if (ledStatePIN == LOW)
        ledStatePIN = HIGH;
    else
        ledStatePIN = LOW;
    digitalWrite(ledPin, ledStatePIN);
}*/
}

//3eme interval cardiaque
if(heartRate>89 && heartRate<=106){
    int periode = 1000;
    time = millis();
    smooth = 128+127*cos(2*PI/periode*time);
    analogWrite(ledPin, smooth);
    Serial.print(heartRate);
    Serial.print(",");

    /*long interval = 400;
if(currentMillis - previousMillis > interval) {
    previousMillis = currentMillis;
    if (ledStatePIN == LOW)
        ledStatePIN = HIGH;
    else
        ledStatePIN = LOW;
    digitalWrite(ledPin, ledStatePIN);
}*/
}

//4eme interval cardiaque
if(heartRate>106 && heartRate<=125){
    int periode = 750;
    time = millis();
    smooth = 128+127*cos(2*PI/periode*time);
    analogWrite(ledPin, smooth);
    Serial.print(heartRate);
    Serial.print(",");

    /*long interval = 200;
if(currentMillis - previousMillis > interval) {
    previousMillis = currentMillis;
    if (ledStatePIN == LOW)
        ledStatePIN = HIGH;
    else
        ledStatePIN = LOW;
    digitalWrite(ledPin, ledStatePIN);
}*/
}

```

```

}

//5eme interval cardiaque
if(heartRate>125 && heartRate<=140){
  int periode = 500;
  time = millis();
  smooth = 128+127*cos(2*PI/periode*time);
  analogWrite(ledPin, smooth);
  Serial.print(heartRate);
  Serial.print(",");

  /*long interval = 100;
  if(currentMillis - previousMillis > interval) {
    previousMillis = currentMillis;
    if (ledStatePIN == LOW)
      ledStatePIN = HIGH;
    else
      ledStatePIN = LOW;
    digitalWrite(ledPin, ledStatePIN);
  }*/
}

if(total1>300){
  digitalWrite(ledTouch, HIGH);
  //Serial.println("TOUCH");
}else{
  digitalWrite(ledTouch, LOW);
}

Serial.print(touch1);
Serial.print(",");
Serial.print(touch2);
Serial.print(",");
Serial.print(touch3);
Serial.print(",");
Serial.print(touch4);
Serial.print(",");
Serial.println("TEST");

delay(100);
}

////////////////////

void setupHeartMonitor(int type){
  //setup the heartrate monitor
  Wire.begin();
  writeRegister(HRMI_I2C_ADDR, 0x53, type); // Configure the HRMI with the

```

```
requested algorithm mode
}
```

```
////////////////////////////////
```

```
int getHeartRate(){
    //get and return heart rate
    //returns 0 if we couldnt get the heart rate
    byte i2cRspArray[3]; // I2C response array
    i2cRspArray[2] = 0;
```

```
    writeRegister(HRMI_I2C_ADDR, 0x47, 0x1); // Request a set of heart rate values
```

```
    if (hrmiGetData(127, 3, i2cRspArray)) {
        return i2cRspArray[2];
    }
    else{
        return 0;
    }
}
```

```
////////////////////////////////
```

```
void writeRegister(int deviceAddress, byte address, byte val) {
    //I2C command to send data to a specific address on the device
    Wire.beginTransmission(deviceAddress); // start transmission to device
    Wire.write(address); // send register address
    Wire.write(val); // send value to write
    Wire.endTransmission(); // end transmission
}
```

```
////////////////////////////////
```

```
boolean hrmiGetData(byte addr, byte numBytes, byte* dataArray){
    //Get data from heart rate monitor and fill dataArray byte with response
    //Returns true if it was able to get it, false if not
    Wire.requestFrom(addr, numBytes);
    if (Wire.available()) {
        for (int i=0; i<numBytes; i++){
            dataArray[i] = Wire.read();
        }
        return true;
    }
    else{
        return false;
    }
}
```