|  |  |
| --- | --- |
|  |  |
|  | #include "MAX30100\_PulseOximeter.h" |
|  | #include <U8g2lib.h> |
|  | #include <Wire.h> |
|  |  |
|  | #define REPORTING\_PERIOD\_MS 500 |
|  | U8G2\_SSD1306\_128X32\_UNIVISION\_F\_HW\_I2C u8g2(U8G2\_R0); |
|  |  |
|  | // PulseOximeter is the higher level interface to the sensor |
|  | // it offers: |
|  | // \* beat detection reporting |
|  | // \* heart rate calculation |
|  | // \* SpO2 (oxidation level) calculation |
|  | PulseOximeter pox; |
|  |  |
|  | const int numReadings=10; |
|  | float filterweight=0.5; |
|  | uint32\_t tsLastReport = 0; |
|  | uint32\_t last\_beat=0; |
|  | int readIndex=0; |
|  | int average\_beat=0; |
|  | int average\_SpO2=0; |
|  | bool calculation\_complete=false; |
|  | bool calculating=false; |
|  | bool initialized=false; |
|  | byte beat=0; |
|  |  |
|  | // Callback (registered below) fired when a pulse is detected |
|  | void onBeatDetected() |
|  | { |
|  | show\_beat(); |
|  | last\_beat=millis(); |
|  | } |
|  |  |
|  | void show\_beat() |
|  | { |
|  | u8g2.setFont(u8g2\_font\_cursor\_tf); |
|  | u8g2.setCursor(8,10); |
|  | if (beat==0) { |
|  | u8g2.print("\_"); |
|  | beat=1; |
|  | } |
|  | else |
|  | { |
|  | u8g2.print("^"); |
|  | beat=0; |
|  | } |
|  | u8g2.sendBuffer(); |
|  | } |
|  |  |
|  | void initial\_display() |
|  | { |
|  | if (not initialized) |
|  | { |
|  | u8g2.clearBuffer(); |
|  | show\_beat(); |
|  | u8g2.setCursor(24,12); |
|  | u8g2.setFont(u8g2\_font\_smart\_patrol\_nbp\_tf); |
|  | u8g2.print("Place finger"); |
|  | u8g2.setCursor(0,30); |
|  | u8g2.print("on the sensor"); |
|  | u8g2.sendBuffer(); |
|  | initialized=true; |
|  | } |
|  | } |
|  |  |
|  | void display\_calculating(int j) |
|  | { |
|  | if (not calculating) { |
|  | u8g2.clearBuffer(); |
|  | calculating=true; |
|  | initialized=false; |
|  | } |
|  | show\_beat(); |
|  | u8g2.setCursor(24,12); |
|  | u8g2.setFont(u8g2\_font\_smart\_patrol\_nbp\_tf); |
|  | u8g2.print("Measuring"); |
|  | u8g2.setCursor(0,30); |
|  | for (int i=0;i<=j;i++) { |
|  | u8g2.print(". "); |
|  | } |
|  | u8g2.sendBuffer(); |
|  | } |
|  | void display\_values() |
|  | { |
|  | u8g2.clearBuffer(); |
|  | u8g2.setFont(u8g2\_font\_smart\_patrol\_nbp\_tf); |
|  |  |
|  | u8g2.setCursor(65,12); |
|  | u8g2.print(average\_beat); |
|  | u8g2.print(" Bpm"); |
|  | u8g2.setCursor(0,30); |
|  | u8g2.print("SpO2 "); |
|  | u8g2.setCursor(65,30); |
|  | u8g2.print(average\_SpO2); |
|  | u8g2.print("%"); |
|  | u8g2.sendBuffer(); |
|  | } |
|  |  |
|  | void calculate\_average(int beat, int SpO2) |
|  | { |
|  | if (readIndex==numReadings) { |
|  | calculation\_complete=true; |
|  | calculating=false; |
|  | initialized=false; |
|  | readIndex=0; |
|  | display\_values(); |
|  | } |
|  |  |
|  | if (not calculation\_complete and beat>30 and beat<220 and SpO2>50) { |
|  | average\_beat = filterweight \* (beat) + (1 - filterweight ) \* average\_beat; |
|  | average\_SpO2 = filterweight \* (SpO2) + (1 - filterweight ) \* average\_SpO2; |
|  | readIndex++; |
|  | display\_calculating(readIndex); |
|  | } |
|  | } |
|  |  |
|  | void setup() |
|  | { |
|  | Serial.begin(115200); |
|  | u8g2.begin(); |
|  | pox.begin(); |
|  | pox.setOnBeatDetectedCallback(onBeatDetected); |
|  | initial\_display(); |
|  | } |
|  |  |
|  | void loop() |
|  | { |
|  | // Make sure to call update as fast as possible |
|  | pox.update(); |
|  | if ((millis() - tsLastReport > REPORTING\_PERIOD\_MS) and (not calculation\_complete)) { |
|  | calculate\_average(pox.getHeartRate(),pox.getSpO2()); |
|  | tsLastReport = millis(); |
|  | } |
|  | if ((millis()-last\_beat>10000)) { |
|  | calculation\_complete=false; |
|  | average\_beat=0; |
|  | average\_SpO2=0; |
|  | initial\_display(); |
|  | } |
|  | } |