Photoplethysmography Signal Sensing and quality checking

LakshmiNarayana Ravada[1],✉

[1] Sophomore, Electronics and Communication Engineering, Indian Institute of technology, Bhubaneswar, Odissa, India

This is report on sensing of Photoplethysmogarphy signal and checking the quality of it using zero crossing rate using arduino and ppg sensor. It is a necessary requirement of any device that measures Cardio properties based on ppg signal.

Introduction

Photoplethysmography (PPG) is an optical measurement method which is widely used to moniter Heart rates. PPG sensor measures the change in volume of blood using transmission and reflection of light. As the heart contracts and expands the blood in the arteries is pumped in and out. This pressurized outflow of blood causes a minute swell in arteries. This sensor measures the absorption rate based on Beer-Lambert's law, The PPG signal is formed by two main components DC offset, which represents constant absorption rate of light passing through vessels

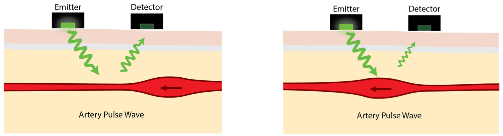

Fig. 1. Placeholder image of Iris with a long example caption to show justification setting.

Arduino is a microcontroller based open-source electronic prototyping board which can be programmed with an easy to use ardiuno IDE. The hardware features with an open-source hardware board designed around an 8-bit Atmel AVR microcontroller or a 32-bit Atmel ARM. Current models consists a USB interface, 6 analog input pins and 14 digital I/O pins that allows the user to attach various extension boards.
5pt The Arduino Uno board is a microcontroller based on the ATmega328. It has 14 digital input/output pins in which 6 can be used as PWM outputs, a 16 MHz ceramic resonator, an ICSP header, a USB connection, 6 analog inputs, a power jack and a reset button. This contains all the required support needed for microcontroller. In order to get started, they are simply connected to a computer with a USB cable or with a AC-to-DC adapter or battery. Arduino Uno Board varies from all other boards and they will not use the FTDI USB-to-serial driver chip in them. It is featured by the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Design Specifications

Ardiuno Uno

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Inout Voltage (limit)	6-20V
Digital I/O Pins	14
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20mA
DC current for 3.3V Pin	50 mA
Clock Speed	16 MHz
LED-BUILTIN	PIN 13

PPG sensor.

Visible Light Source	860nm (5nm LED)
Infrared Light Source	860nm (5nm LED)
Supply	5V
Measurement Site .	Only at peripheral sites for Visible light source.

Block Diagram

PPG sensor:

S to A0, +ve to 5v in arduino, -ve to Gnd in arduino.

HC-05 Module:

RX to 9, TX to 8, +5V to +5V, Gnd to Gnd in ardiuno.

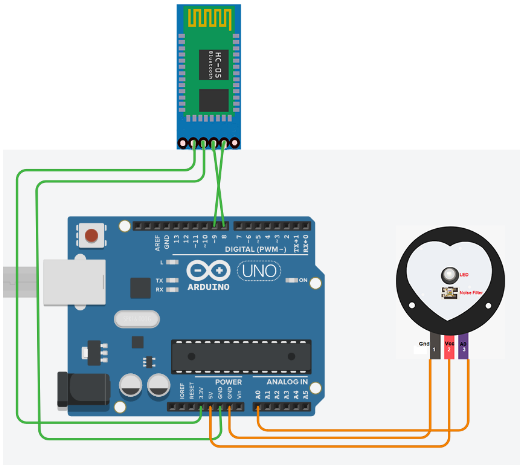

Fig. 2. Circuit diagram for .

Algorithm

The main Objective of this project is to find **zero crossing** rate of measured ppg signal using ppg sensor.

The signal is taken as input using `analogRead(A0)`, this is the raw signal the was obtained form ppg sensor.

```
    input = analogRead(A0)                    (1)
```

To obtain more accurate results form the data, The data in `i` th index is meaned over next `k` values. This brings the data down to `n-k` samples form `n` samples.

```
    for(int j = i+1; j < i+k+1; j++){
        data[i] = data[i] + data[j]           (2)
    }
```

Calculation of mean of the meaned data, `mean` is sum of all data and division by total samples in data. Initialise `mean` to `0` at the start.

```
    float mean = 0.0
    for(int i = 0; i < n-k; i++){
           mean = mean + data[i]
    }                                         (3)
```

Optimising data to be efficient in transferring via HC-05 Bluetooth module in single byte transfer. Ranging all the data in `[-200, 200]` by dividing with `maximum` and multiplying with `200`.

```
    float maximum = 0.0
    for(int i = 0; i < n-k; i++){
          if(data[i] > maximum){
              maximum = data[i]
          }
    }                                         (4)

    for(int i = 0; i < n-k; i++){
      data[i] = (data[i]/maximum)*200
    }                                         (5)
```

Calculation of zero crossing rate `zcr` **(8)** by summing positive to negative crossings **(6)** and negative to positive crossings **(7)** .

```
    int zcr = 0.0
    for(int i = 0; i < n-k; i++){
       if(a[i] > 0 && a[i+1] < 0){
              zcr = zcr + 1
          }                                   (6)
       if(a[i] < 0 && a[i+1] > 0){
              zcr = zcr + 1
          }                                   (7)
    }
    zcr = pzcr + nzcr                         (8)
```

Real Time Evaluation and Results

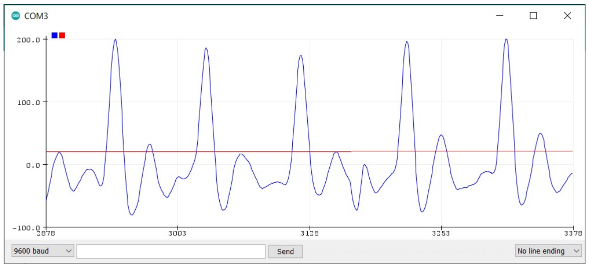

Fig. 3. High Quality PPG signal

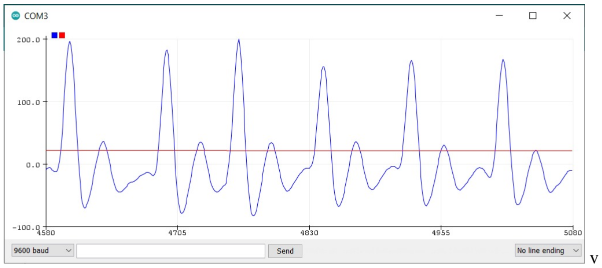

Fig. 4. Zero crossing rate for Minutely Distorted Quality PPG signal

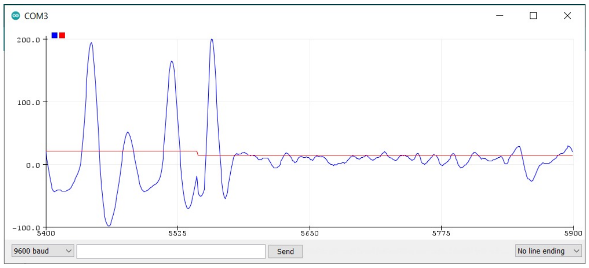

Fig. 5. Zero crossing rate for Poor Quality PPG signal

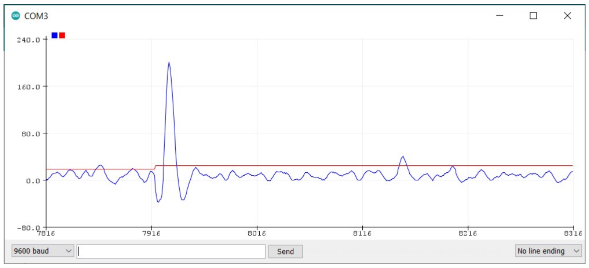

Fig. 6. Zero crossing rate for Very Poor Quality PPG signal

We can observe that zero crossing rate does not lie in the interval, either it goes up or it goes down.

Conclusion

This concludes that a quality of PPG signal can determined by calculation of zero crossing rate. Which can be found by counting total number of crossings around the x-axis in the normalised data, then compare it with values obtained from good quality signals and averge of zero crossing rates of large samples. This tells of the signal is good enough for extracting necessary data or mostly noise.

Source Code

https://github.com/NarayanaRavada/PPG-Signal-Quality-Checking-using-ardiuno