Heart-Rate Monitoring System Using Photoplethysmography (PPG)

Submitted By- Kartikeya Raj and Aditya Gupta

1/7/2014

Abstract

Photoplethysmography(PPG) is a simple and low-cost optical technique that can be used to detect blood volume changes in the microvascular bed of tissues. It is used to estimate the skin blood flow using infrared light. It is of great use in medical instruments because of its advantages as non-invasive, inexpensive and convenient diagnostic tool. This project demonstrates a technique to measure the heart rate by sensing the change in blood volume in a finger artery while the heart is pumping the blood. The prototype of the portable system build in this project is comprised of infrared (IR) LED sensors, a filtering circuit, an amplifying circuit, an Arduino UNO board and an SD card module. In the prototype a simple clothes peg was augmented with infrared LED/phototransistor sensors and was able to measure the light absorbance through the skin of fingertip in order to isolate a PPG waveform. A receiver circuit was designed to amplify and filter the resulting signal. The output of the circuit was then given to the Arduino board which performed the digital analysis of the signal. The Arduino board was programmed to measure the heart rate and send the sampled input data to the SD card interfaced with it for future reference. The sampled data was also send to the PC through a serial interface and was plotted in MATLAB where the signal can be processed and analyzed further.

Table of Contents

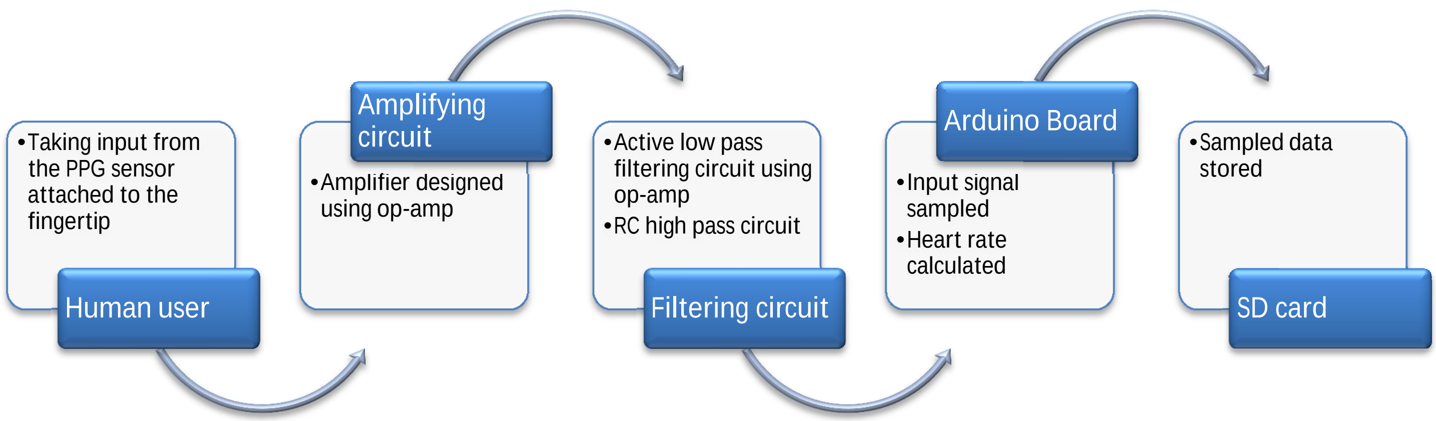
Introduction

Heart rate measurement indicates the soundess of the human cardiovascular system. The heart rate is typically expressed as beats per minute (bpm). The heart rate can vary according to the body's physical needs, including the need to absorb oxygen and excrete carbon dioxide. Activities that can provoke change include physical exercise, sleep, anxiety, stress, illness, ingesting, and drugs. The device commonly used in hospitals for heart rate measurement is the pulse oximeter. Pulse oximeters use the basic principle of photoplethysmography (PPG). Pulse oximeters flashes both red and infrared light through the finger and a photodetector measures the change of absorbance of the two different wavelengths.

Photoplethysmography is a non-invasive technique that measures relative blood volume changes in the blood vessels close to the skin. In recent years, it has developed into a popular non-invasive method for assessing mean arterial blood pressure and oxygen saturation. The measurement of blood volumetric changes in the skin perfusion by means of PPG depends on the fact that blood absorbs infrared light many times more strongly than the remaining skin tissues. PPG has several advantages. It uses simple inexpensive optical sensors that need little maintenance. The device is compact and is portable. Hence it can be used in all types of environments. The PPG sensor consists of an LED and a photo detector.

Using the concept of photoplethmosgraphy (PPG), changes of blood volume can be measured during each heart beat. It is based on the determination of optical properties of vascular tissue using a light source and a photodetector (PD). In contrast to oximetry, only one led light is used to transmit light in PPG. As the light is emitted, blood levels and tissues absorb various amounts of the light, causing different detections sensed from a photodetector. Since water absorbs light in the ultraviolet and longer IR region whereas red and near-IR light pass easily, therefore, we have used IR wavelengths as the light source in the PPG sensor. There are 2 different types of detection modes using the photodetector: transmission and reflection. Transmission mode occurs when the led source is transmitted through the skin and a detection occurs on the other side of the skin. This method can only be done through areas of the body thin enough for the photodetector to read a measurable signal. For example, possible implementations can be done through any finger, earlobes, and the toes. The second mode, reflection, occurs when both the led and photodetector are on the same side of the skin. Here we have used the photodetector in transmission mode.

Design



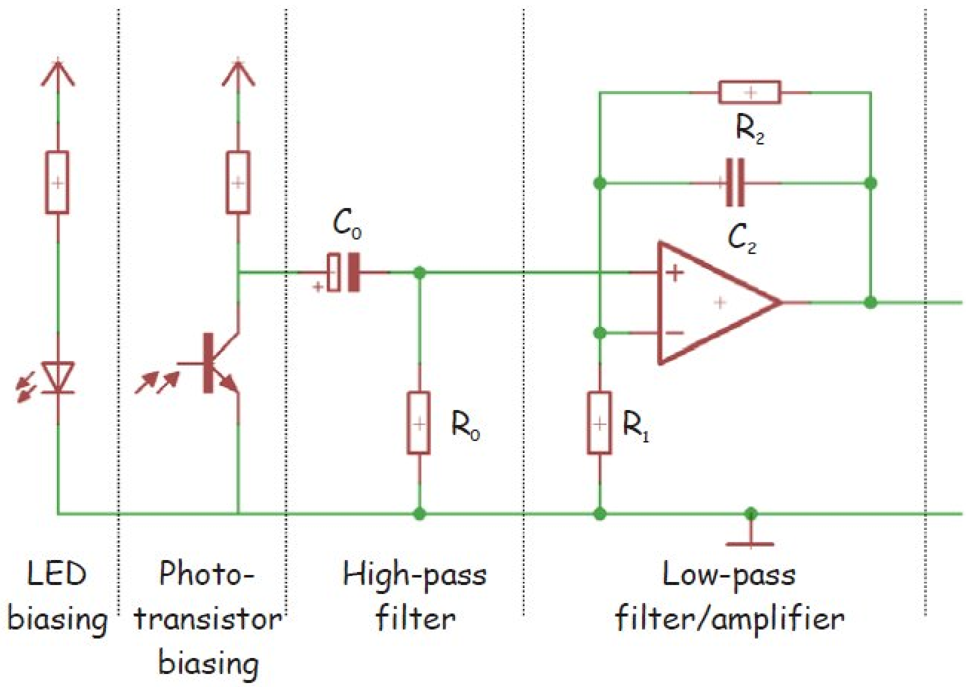
The electronic circuit

The electronic circuit has four functions:

- Biasing of LED
- Biasing of photo detector
- Amplification of the input PPG signal
- Removal of noise

The first step is biasing of LED. In our circuit, for an operational voltage of 5V a 240ohm resistor is used. The photodetector is biased next. A 100k ohm resistor is used for its biasing.

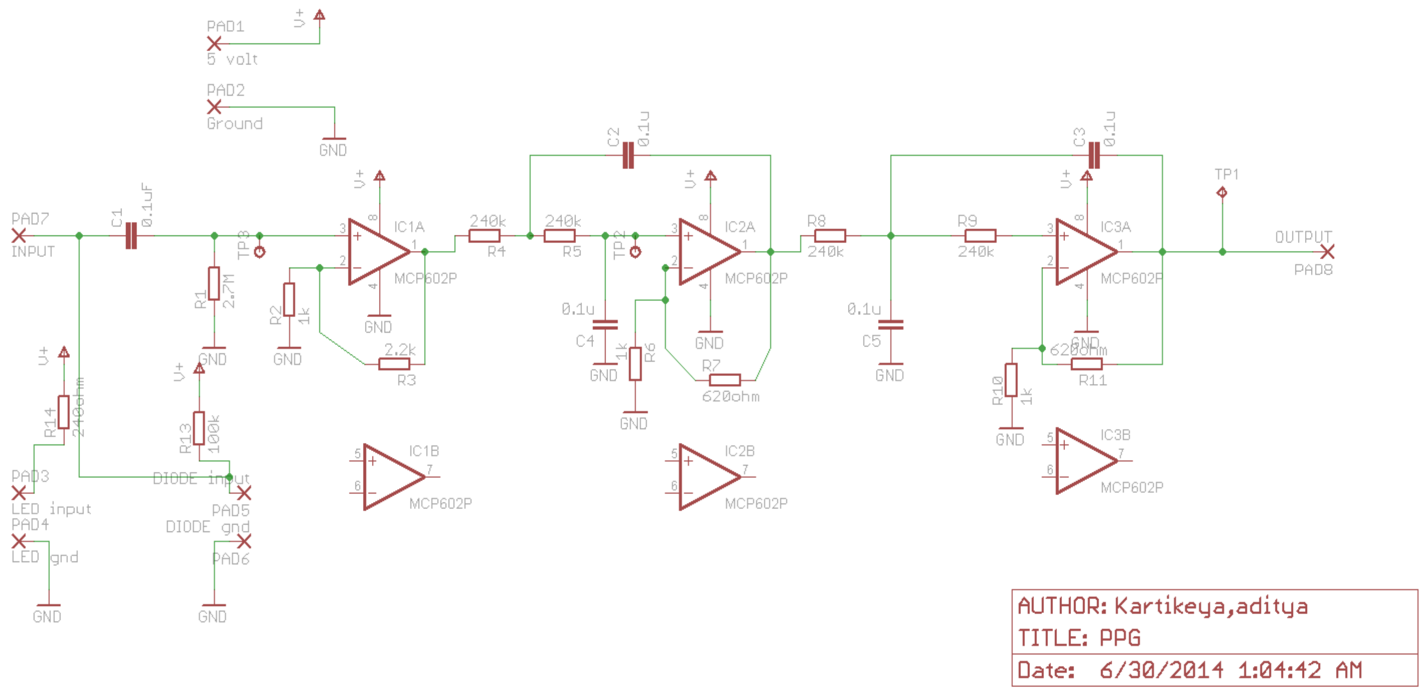
The high-pass filter is a simple first-order RC circuit with a lower frequency bound of $(2\pi R0C0)^{-1}$. The low -pass filter is implemented as an active filter to facilitate amplification as well. The high pass filter has a cut-off at $(2\pi R2C2)^{-1}$ and amplifies (R1 + R2) / R2 times. The high-pass filter frequency and the low–pass filter frequency have to be chosen such that we end -up with a band-pass filter. To make the bandpass-filter more effective, the high-pass and low-pass filters are implemented twice.



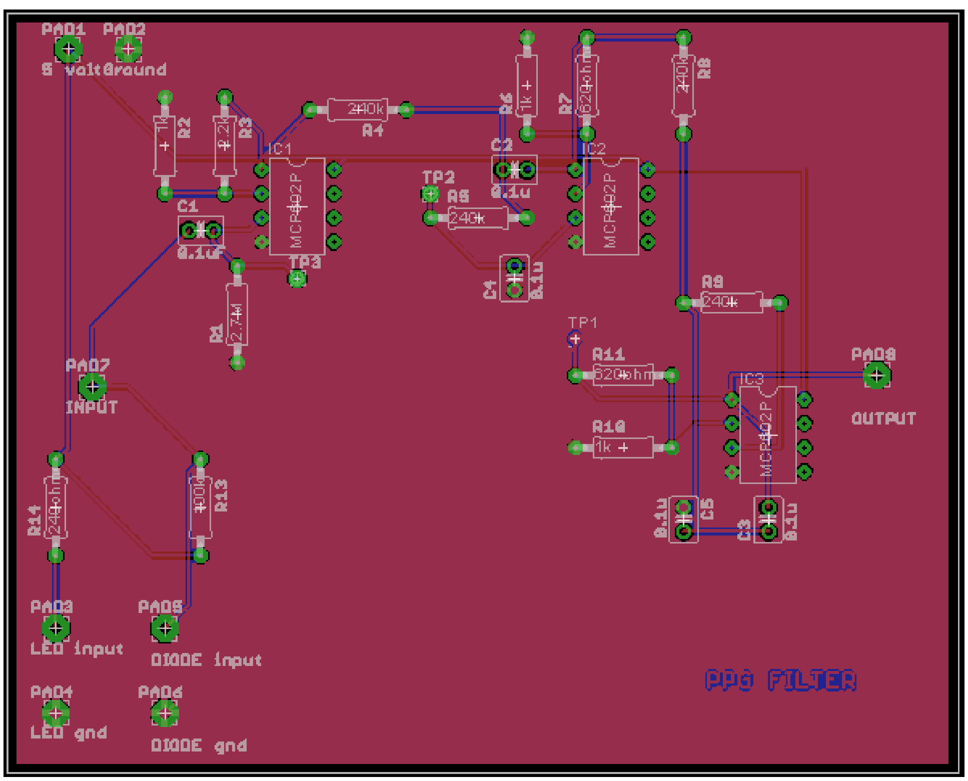
The basic circuit topology is shown above.

The output produced from the photodiode is a small analog current value. When placed through the transimpedance amplifier, the current is converted into a voltage value with a gain depending on the feedback resistor. The resistor value is chosen such that the TIA gain is as large as possible. The voltage drop caused by the DC component of the photodetector output current limits this value. The capacitor serves to bandwidth limit the signal, disregarding high frequency noises. The output voltage from the TIA is then fed into the next stage, which acts as a high pass filter and DC bias. The bandpass amplifier places bandwidth limitations onto the signal in order to acquire the desired response. The lower cutoff is set around 0.5Hz while the upper cutoff is set to around 5Hz. This range correlated to a human heart rate range that the circuit acquires be between 30 beats per minutes (BPM) and 300 BPM.

Power can be supplied to the circuit by a battery so as to make it portable. The operational amplifier IC used in the circuit is MCP602, a dual Op-amp chip from Microchip. It operates at a single power supply and provides rail-to-rail output swing.



Finally, the whole circuit is shown in figure above. The sensor is now replaced by a connector, so only the biasing resistors are visible. Note that the circuit shown in Figure 2.3 was drawn using the free version of Eagle Schematic. The PCB design of the circuit is shown below.



Arduino Sketch

There are three parts in the arduino sketch:

- Sampling data and calculating heartbeat rate.
- Sending the sampled data to the SD card interfaced with the board by using the SPI protocol.
- Interfacing MATLAB with the Arduino so that the data can be send serially to the PC and be plot using MATLAB.

Arduino code for computing the heartbeat

The goal is to find successive moments of instantaneous heart beat and measure the time between, called the Inter Beat Interval (IBI). Ideally, we wanted to find the instantaneous moment of the heart beat. This is important for accurate BPM calculation, Heart Rate Variability (HRV) studies, and Pulse Transit Time (PTT) measurement. It was found by reading papers of heart researchers that the instantaneous moment of heart beat happens at some point during that fast upward rise in the PPG waveform. In our code the IBI was measured by timing between moments when the signal crosses 50% of the wave amplitude during that fast upward rise. The BPM was derived every beat from an average of the previous 10 IBI times.

To have a regular sample rate with high enough resolution we set up Timer2, an 8 bit hardware timer on the ATmega328 (UNO), so that it throws an interrupt every other millisecond. That gives us a sample rate of 500Hz.

```
void interruptSetup(){
  TCCR2A = 0x02;
  TCCR2B = 0x06;
  OCR2A = 0x7C;
  TIMSK2 = 0x02;
  sei();
}
```

The register settings above tells Timer2 to go into CTC mode, and to count up to 124 (0x7C) over and over and over again. A prescaler of 256 was used to get the timing right so that it takes 2 milliseconds to count to 124. An interrupt flag was set every time Timer2 reaches 124, and a special function called an Interrupt Service Routine (ISR) that we wrote is run at the very next possible moment.

```
ISR(TIMER2_COMPA_vect){
  Signal = analogRead(pulsePin);
  sampleCounter += 2;
  int N = sampleCounter - lastBeatTime;
```

This function is called every 2 milliseconds. It takes the analog reading from the input and increments the variable sampleCounter.

We then kept track of the highest and lowest values of the PPG wave, to get an accurate measure of amplitude.

```
    if(Signal < thresh && N > (IBI/5)*3){
      if (Signal < T){
        T = Signal;
      }
    }
    if(Signal > thresh && Signal > P){
        P = Signal;
      }
```

Variable P and T hold peak and trough values, respectively. The thresh variable is initialized at 512 (middle of analog range) and changes during run time to track a point at 50% of amplitude as we will see later. There is a time period of 3/5 IBI that must pass before T gets updated as a way to avoid noise and false readings from the dicroic notch.

To check if we have a pulse we the following code:

```
if (N > 250){                                  // 250 millisecond minimum N places an upper limit of 240 BPM
if ( (Signal > thresh) && (Pulse == false) && (N > ((IBI/5)*3) ){
    Pulse = true;
    digitalWrite(pulsePin,HIGH);
    IBI = sampleCounter - lastBeatTime;
    lastBeatTime = sampleCounter;
```

The following code is to store the values of time interval in an array. The value is calculated only after the second beat and the first beat is ignored. The BPM is hence derived from an average of the last 10 IBI values.

```
if(secondBeat){
      secondBeat = false;
      for(int i=0; i<=9; i++){
        rate[i] = IBI;
      }
    }
               if(firstBeat){
                  firstBeat = false;
                  secondBeat = true;
                  sei():
                  return;
               }
word  runningTotal = 0;
    for(int i=0; i<=8; i++){
      rate[i] = rate[i+1];
      runningTotal += rate[i];
    }
    rate[9] = IBI;
    runningTotal += rate[9];
    runningTotal /= 10;
    BPM = 60000/runningTotal;
    QS = true;
    }
}

if (Signal < thresh && Pulse == true){
    digitalWrite(13,LOW);
    Pulse = false;
    amp = P - T;
    thresh = amp/2 + T;
    P = thresh;
    T = thresh;
  }
 if (N > 2500){
    thresh = 512;
    P = 512;
```

```
    T = 512;
    firstBeat = true;
    secondBeat = false;
    lastBeatTime = sampleCounter;
  }
```

If there is no beat event for 2.5 seconds, variables used to find the heartbeat are reinitialized to the start up values.

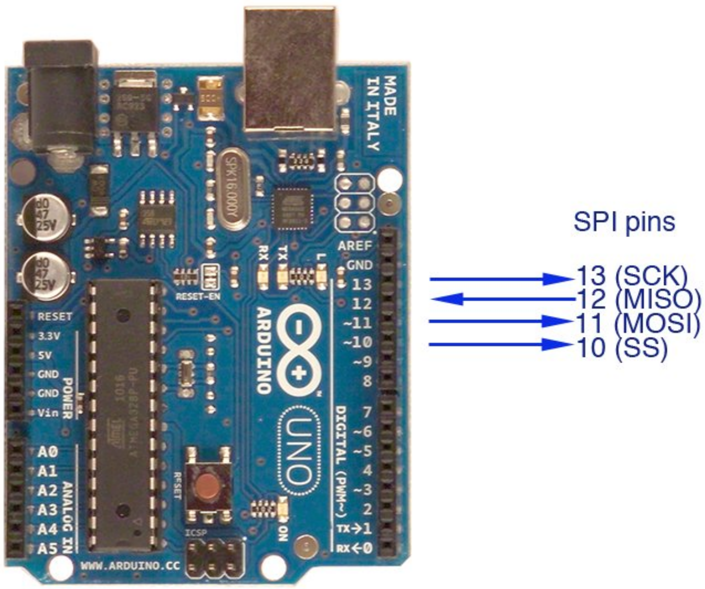
The loop function is as follows:

```
void loop(){
  sendDataToProcessing('S', Signal);
  if (QS == true){
    sendDataToProcessing('B',BPM);
    sendDataToProcessing('Q',IBI);
    fadeVal = 255;
    QS = false;
   }
   ledFadeToBeat();
   delay(20);
}
```

Code to print the data in the serial window:

```
void sendDataToProcessing(char symbol, int data ){
  Serial.print(symbol);
  Serial.println(data);
}
```

Introduction to the Serial Peripheral Interface (SPI)



Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers.

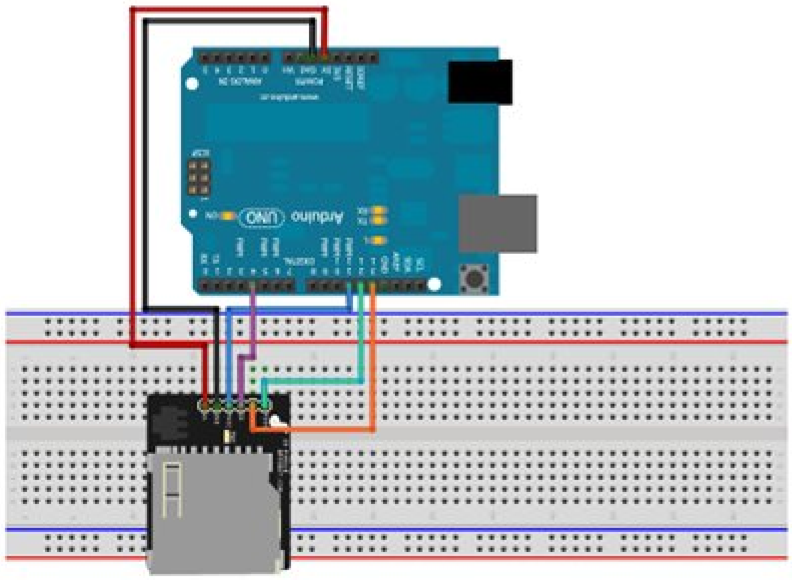
With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices. Typically there are three lines common to all the devices:

- MISO (Master In Slave Out) - The Slave line for sending data to the master,
- MOSI (Master Out Slave In) - The Master line for sending data to the peripherals,
- SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master and one line specific for every device:

- SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices.

When a device's Slave Select pin is low, it communicates with the master. When it's high, it ignores the master. This allows you to have multiple SPI devices sharing the same MISO, MOSI, and CLK lines.

Arduino code for interfacing SD card using SPI



- Power up the SD card module by connecting the 5v and GND to UNO
- Connect the MOSI on SD module to pin 11 on UNO.
- Connect the MISO on SD module to pin 12 on UNO.
- Connect the SCK on SD module to pin 13 on UNO.
- Connect the CS on SD module to pin 4 on UNO.

Code:

```
#include<SD.h>

//#include<Wire.h>

//MOSI MISO SCLK SET BY DEFAULT

//spi sd dard pins

//mosi = pin 11;

//miso =pin 12;
```

```
//sclk =pin 13;
int CS_pin=10;//chip select pin we use it when we use spi
int pow_pin=8;//power pin we have sd card attached from pin 8 to 13
              //8 is going to send 5 volt to pin 8 selecting it as digital high
int IR1_pin=2;
float refresh_rate=0.0;
long id=1;//use thhis to store the id # of our reading
void setup()
{
  Serial.begin(115200);
  Serial.println("Initializing Card");
    pinMode(CS_pin,OUTPUT); //CS pin is an output
 //card will draw power from pin 8, so set it high
  pinMode(pow_pin,OUTPUT);
  digitalWrite(pow_pin,HIGH);
  //check if card  is ready
  if(!SD.begin(CS_pin))//begin an sd card process and check if its working or connected
  {
    Serial.println("card failed!!");
    return;
  }
  Serial.println("Card ready ");
  //Read the Configuration information (COMMANDS.txt)
  File commandFile = SD.open("COMMANDS.txt");
  if (commandFile)
  {
```

```
  Serial.println("Reading Command File");

  float decade = pow(10, (commandFile.available() - 1));
  while(commandFile.available())
  {
    float temp = (commandFile.read() - '0');
    refresh_rate = temp*decade+refresh_rate;
    decade = decade/10;
  }
  Serial.print("Refresh Rate = ");
  Serial.print(refresh_rate);
  Serial.println("ms");
}
else
{
  Serial.println("Could not read command file.");
  return;
}
//write log file header
File logFile=SD.open("LOG.csv",FILE_WRITE);
if(logFile)
{
  logFile.println(", , , ,");//just a leading blank line to seperate data
  String header="ID,IR1";
  logFile.println(header);
  logFile.close();
```

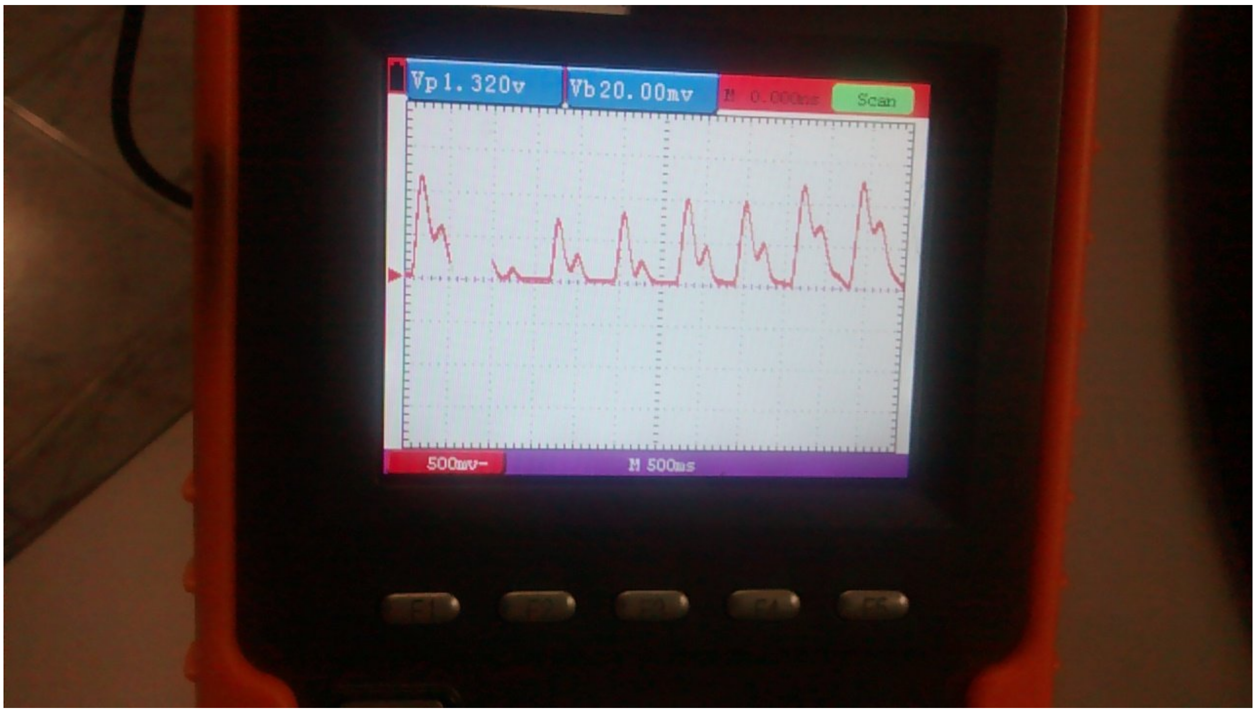
```
    Serial.println(header);
  }
  else
  {
    Serial.println("cant open the file!!");
  }
}


 void loop()
{
  int IR1_val=analogRead(IR1_pin);
   //IR1_val=(IR1_val,0,1024,0,5);
  //Serial.println(IR1_val);
  String dataString=String(id) + "," + String(IR1_val);
  //open a f`ile to write
  //only one file can be open at a time
  File logFile=SD.open("LOG.csv",FILE_WRITE);// we have data file of type file which open on sd card
log.txt and we want to write to it
  if(logFile)
  {
    logFile.println(dataString);
    logFile.close();
    Serial.println(dataString);
  }
  else
  {
```

```
    Serial.println("coulnt access file");
  }
  id++;
 delay(3);
}
```

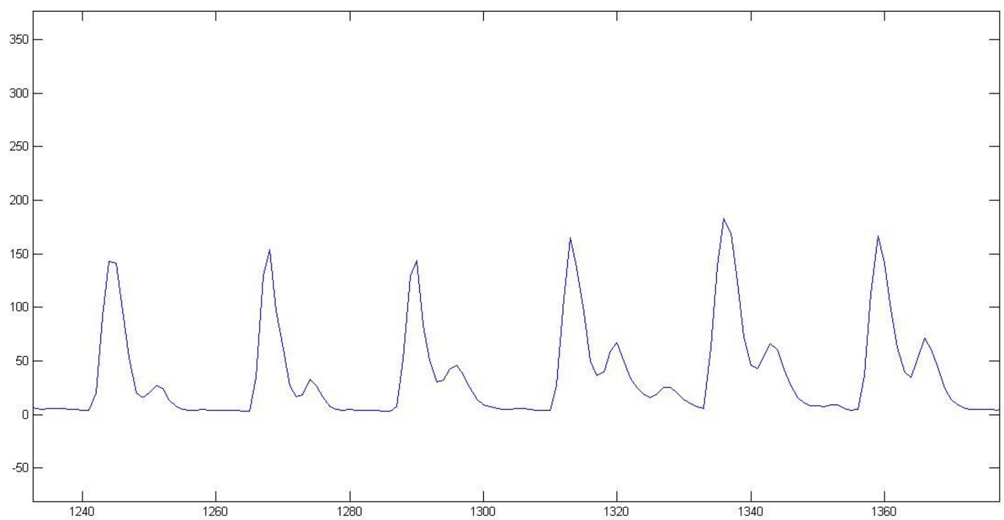
Analysis and Sample Results

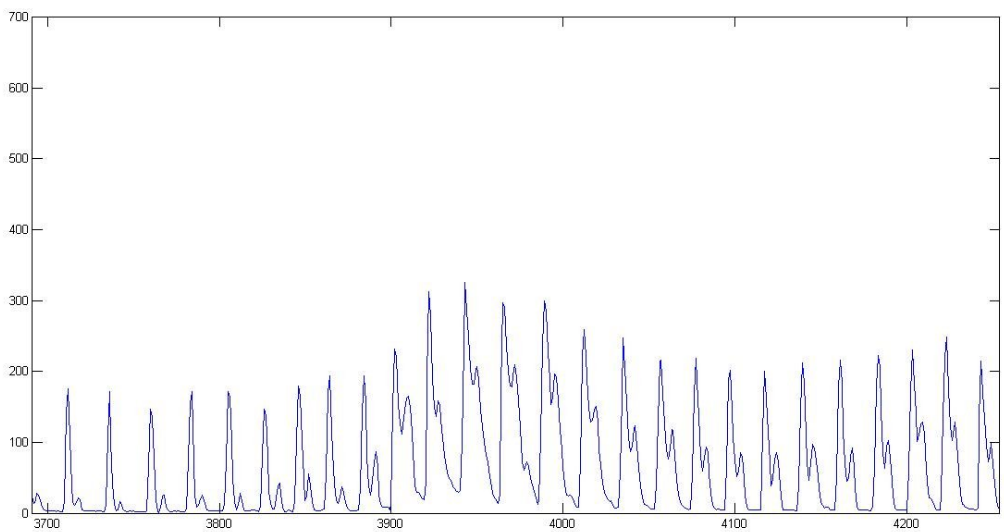
Construction of the design began with placing the components for the filtering circuit on a breadboard and getting a measurable reading from the output of the circuit. The PPG signal was filtered and amplified in the circuit using passive RC high pass circuit, active low pass circuit and operational amplifier for amplification.

The output was measured using an oscilloscope and is shown below:



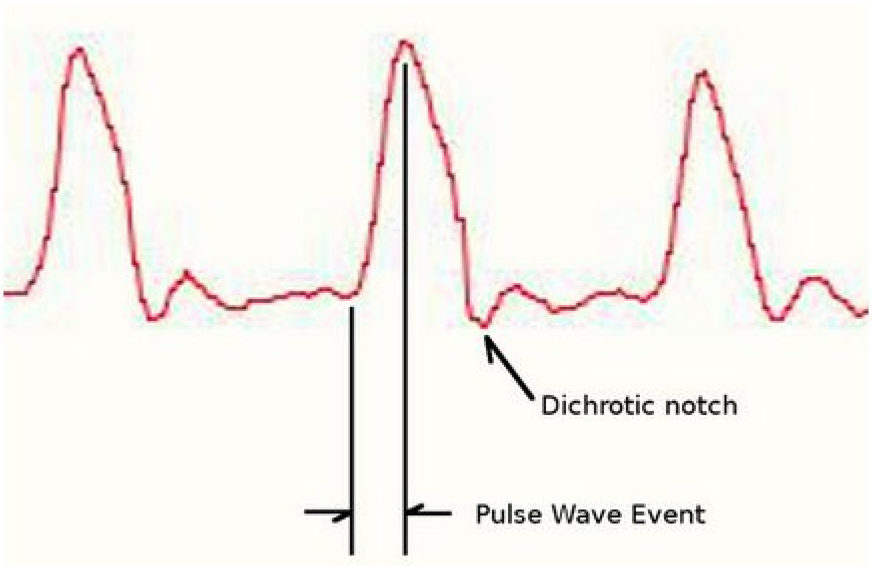
The output of the filtering circuit was then given to one of the analog inputs of the Arduino UNO borad. The data of the analog input was sent serially to the PC and was plotted in MATLAB. The MATLAB plot is shown below:





Finding the heartbeat rate

The heart pulse signal that comes out of a photoplethysmograph is an analog fluctuation in voltage, and it has a predictable wave shape as shown in the figure below. The depiction of the pulse wave is called a photoplethysmogram, or PPG.



To find the heart rate we need to find successive moments of instantaneous heart beat and measure the time between, called the Inter Beat Interval (IBI). When the heart pumps blood through the body, with every beat there is a pulse wave (kind of like a shock wave) that travels along all arteries to the very extremities of capillary tissue where the PPG sensor is attached. A rapid upward rise in signal value occurs as the pulse wave passes under the sensor, then the signal falls back down toward the normal point. Sometimes, the dicroic notch (downward spike) is more pronounced than others, but generally the signal settles down to background noise before the next pulse wave washes through. Since the wave is repeating and predictable, we could choose almost any recognizable feature as a reference point, say the peak, and measure the heart rate by doing math on the time between each peak. In this project we measure the IBI by timing between moments when the signal crosses 50% of the wave amplitude during that fast upward rise. The BPM is derived every beat from an average of the previous 10 IBI times**.**

The heartbeat rate was measured for a normal subject following the above algorithm and it was observed that it was in the range of 60-80 beats per minute which is the normal range.

Advantages and drawbacks

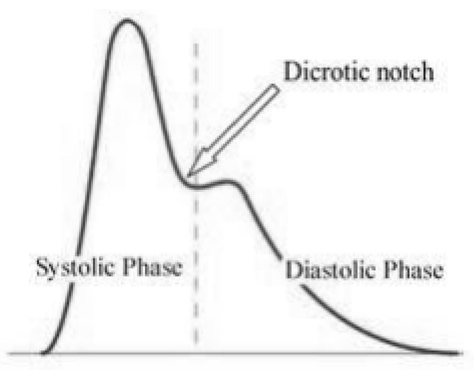
Our design has an advantage of being a low cost, simple and portable technology for heart rate measurement. It is non-invasive and the PPG signal is strong. Because of no electrical contact to the human body it is safe. However, motion artefacts is an observed drawback of the design.

Discussions

The Dicrotic Notch

Photoplethysmography (PPG) is a non-invasive optical technique that measures relative blood volume changes in the blood vessels and is widely used for research and physiological studies. Dicrotic notch represent the closure of the aortic semi-lunar valve and subsequent receding blood flow when ventricles relax. Their location is used to calculate systolic time intervals and monitor cardiac function.

PPG wave analysis helps to study diabetes & arthritis. It is unique for each individual so it would also give unique identification as biometric identification. Pulse wave analysis also helps to study large artery damage & an abnormality in the cardiovascular. PPG analysis emphasizes the importance of early evaluation of the diseases. Generally, photoplethysmography wave is composed of alternating part (AC component) and non-pulsatile component (DC component.



As shown in Figure above, the first wave in conventional PPG wave form is forward traveling called systolic peak and the second one is diastolic peak which is the reflected wave that returns from the periphery. Aortic notch or Dicrotic notch is a small downward deflection in the arterial pulse that separates systolic and diastolic phase. The dicrotic notch signal depends on the interaction of the initial pressure wave when the heart contracts, arterial stiffness that decreases the pulse transit time, and the reflected pressure wave from the peripheral arterial bed. The monitoring of of these dicrotic notches are important for clinicians for analyzing the dynamic blood volume changes that depends on the features of the heart function, size and elasticity of the blood vessels, and specific neural processes.

Detection of arrhythmia

An arrhythmia is a problem with the rate or rhythm of the heartbeat. During an arrhythmia, the heart can beat too fast, too slow, or with an irregular rhythm. The standard clinical tool for its diagnosis is electrocardiogram (ECG or EKG.). An ECG shows how fast the heart is beating and its rhythm (steady or irregular). It also records the strength and timing of electrical signals as they pass through the heart.

Although PPG gives the summary information reflecting both cardiac and blood vessel components of HRV, some research studies showed a significantly high correlation between interbeat interval data measured by both ECG and PPG in short-term steady-state recordings. Research suggests that the pulse-to-pulse interval (PPI) can be considered the same as the R-R interval (RRI) by ECG provided the subject's physiological state is static. PPG is more sensitive to body movements than ECG. Thus, arrhythmia can also be diagnosed using PPG waveform if the motion artefacts are removed and an algorithm is developed using the correlation characteristics of ECG and PPG.

Conclusion

Photoplethysmography (PPG) has been widely used in many biomedical applications such as Pulse oximetry, detection of varicose veins, muscle pump test etc. PPG is easy to set up, convenient, simple and economically efficient as compared to other biomedical instruments. This project describes the potential application of this technology for being used in heartbeat measurement systems. The results of this design show that PPG can be an alternative diagnostic tool to study the cardiovascular system and especially heart rate variability. Our design displays the near-real-time PPG waveform and heart rate and stores the data in an SD card for future reference.

Diseases like Arrhythmia can be diagnosed using this technology. Arrhythmia is any of a group of conditions in which the electrical activity of the heart is irregular or is faster or slower than normal. In conventional clinical practice arrhythmia is diagnosed using electrocardiography (ECG) but the technology of PPG can be developed further for the diagnosis of arrhythmia accurately. This technology can be used for tele-medicine e. g. for distant cardio-vascular monitoring via Internet.