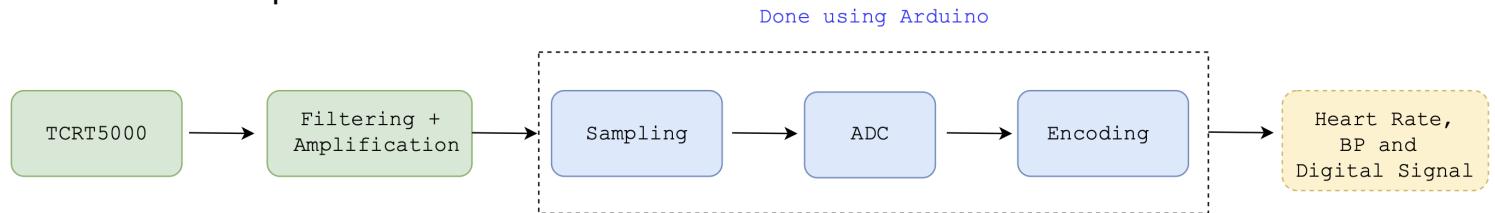


Heart Rate and BP Measurement using PPG

Abstract : Obtaining Heart rate and Blood Pressure using the Photoplethysmography(PPG) signal obtained from TCRT5000. Due to the systolic and diastolic cardiac cycle of heart results in change of blood volume in the finger which can be captured using PPG. The reflectance depends on the volume of blood flowing through the finger. So on filtering and processing the captured wave the heart rate and the blood pressure can be obtained.

Idea behind the Implementation :



Process/Steps Involved :

- 1) Biasing TCRT5000
- 2) Passive High Pass Filter (HPF)
- 3) Buffer
- 4) Active Low Pass Filter (LPF)
- 5) Inverting Amplifier and Rectifier
- 6) Processing using Arduino (This can be done using Sampler-ADC-Encoder)

Biassing TCRT5000 :

TCRT5000 is setup of IR sensor cum PhotoTransistor that transmits IR (780nm - 1mm) EMR and the PhotoTransistor gives an Analog voltage output of transmitted/reflected IR EMR.

TCRT5000 is biased according to Fig.1 (A).

For Emitter (Diode) the maximum reverse voltage (R_V) that can be applied is 5V and the maximum forward current (I_F) can be 60mA.

For the Detector (PhotoTransistor) the maximum emitter collector voltage (V_{EC}) applied can be 5V and maximum collector current (I_C) can be 100mA.

Hence the input voltage given is (4-5)V and the obtained I_F is 5mA. Where I_C depends on incident light and $5V > I_C R_1$ in order to keep the transistor in the active region.

[Link](#) to the biased TCRT5000 setup.

The output from TCRT5000 can be observed in Fig.2 (A)

(Note : Prolonged eye contact may cause damage to eye, So mobile camera can be used to check the working)

Filtering and Amplification

Passive High Pass Filter (HPF) :

Capacitor of $1\mu F$ and Resistor of $330K\Omega$ is used. Passive HPF can be circuited as in Fig.1 (B). So the Cut-Off frequency of HPF is,

$$f_{HPF} = \frac{1}{2\pi RC} = \frac{1}{2\pi(0.33)} = 0.48HZ$$

So this implies that the signal having frequency less than 0.48HZ will be filtered including the DC value arised due to the constant incident of light.

Passive HPF is used in order to maintain high input impedance in order to not alter the output from

transistor.

Buffer :

The reason behind adding the buffer after passive HPF is to maintain high input impedance for active LPF.

For Buffer 1st OPAMP of LM324N is used where the input is given to Non-Inverting terminal and Inverting/Output terminals are connected.

The output observed after Passive HPF + Buffer is Fig.2 (B).

Active Low Pass Filter :

Resistor of (10KΩ and 1KΩ), Capacitor of $4.7\mu F$ and 2nd OPAMP of LM324N is used. The active LPF is circuited as in Fig.1 (B). The cut-off frequency of LPF is,

$$f_{LPF} = \frac{1}{2\pi RC} = \frac{1}{2\pi(0.047)} = 3.39\text{HZ}$$

So this implies that the signal with frequency higher than 3.39HZ will be filtered (High frequency components).

And Active LPF amplifies the input by 10x as $A = 10\text{K}\Omega/1\text{K}\Omega$ (Inverting).

The output observed after Active LPF is Fig.2 (C).

As the Heart rate lies in the range of (60-190) bpm which is (1-3.16)HZ, Hence the above mentioned f_c is chosen.

Amplification :

Resistor of (22KΩ and 1KΩ) and 3rd OPAMP of LM324N is used. Inverting Amplifier can be circuited as in Fig.1 (D).

The Gain of the inverting Amplifier is,

$$A = \frac{R_7}{R_6} = \frac{22K}{1K} = 22$$

Hence the input is getting amplified by 22x.

The output observed after Amplification is Fig.2 (D).

[Link to the Filter and Amplification Setup.](#)

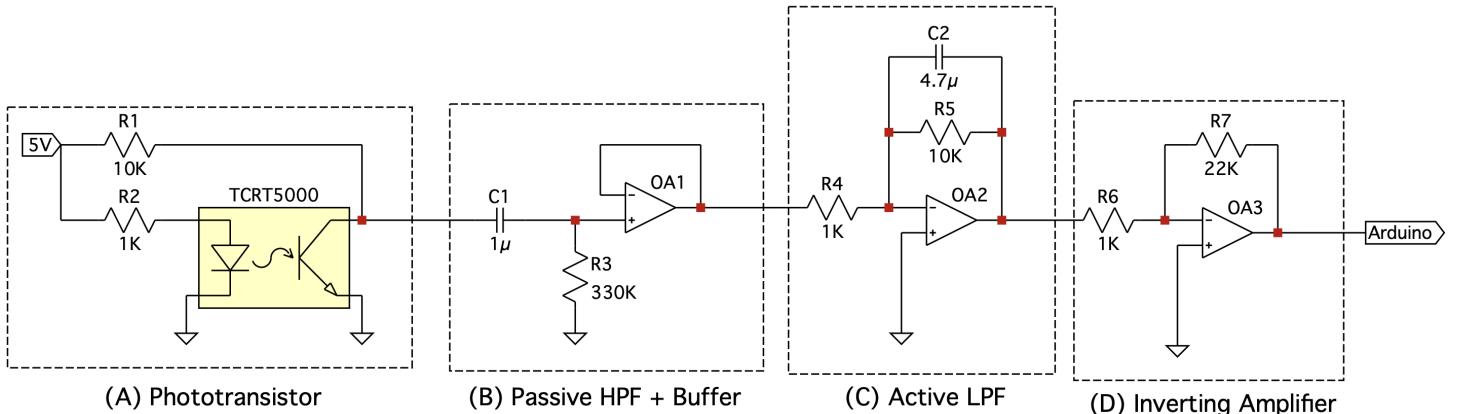


FIGURE 1

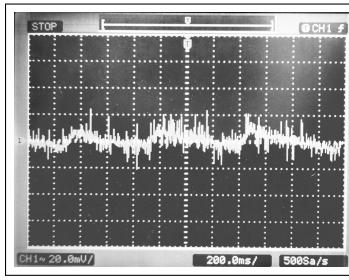


Figure.2 (A)

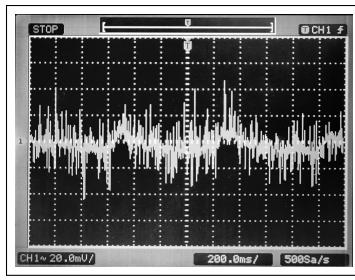


Figure.2 (B)

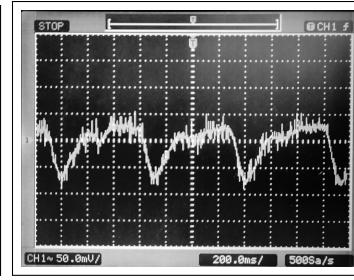


Figure.2 (C)

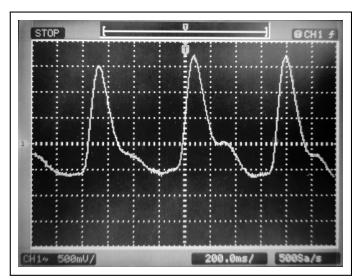


Figure.2 (D)

Processing using Arduino

The signal received from the Amplifier has to be sampled using ADC and FFT(Fast Fourier Transform) has to be performed in order to obtain the peak. The accuracy of the FFT depends on the Sampling frequency and the duration of sampling. The sampling frequency is taken to be 16HZ (Default Sampling Frequency of Arduino) which means in 1 second 16 samples (input) is taken from the signal. And coded such that sampling is done for 16s.

$$\text{Heart Rate} = \frac{60 \times \text{Sampling Frequency} \times \text{Peak}}{\text{No.of Samples}}$$

And Blood Pressure can be calculated using the same setup using the following formula,

$$\text{Systolic BP} = 184.3 - 1.32 \times \text{HR} + 0.0848 \times T_d$$

$$\text{Diastolic BP} = 55.96 - 0.02912 \times \text{HR} + 0.023 \times T_d$$

Where, $T_d = \text{HR}(\text{in ms}) - \text{Timedelay}$, Timedelay is taken as the time difference between Maximum frequency and Minimum frequency.

The output from amplifier is given to an Analog pin (A0) of Arduino and the ground of Arduino is connected to ground of the circuit.

The Arduino is programmed accordingly using certain libraries. [Here](#) is the code used in the Arduino in order to measure the Heart Rate and Blood Pressure.

Conclusion :

Here is the [Link](#) to the video of measurement of Heart Rate and Blood Pressure (Systolic and Diastolic). And here is the [Link](#) to screenshot of serial monitor of Arduino while measuring BP and Heart Rate.

Processing using Discrete Components

Sampler :

A MOSFET (N7000) and Capacitor of Capacitance 2.2pF is used to sample the signals. The sampler digitizes the given analog signal and holds the signals for ADC. The sampler can be circuited as in Fig.3 (A). The output observed after passing through sampler is Fig.4

ADC (Analog to Digital Converter) :

In order to design ADC, 2 LM324N (7 OPAMP) and 8 resistors of resistance 2KΩ is used. ADC can be circuited as in Fig.3 (B).

So it's a 3 bit ADC which has 8 outputs. It works such that for $V_{in} > V^-$ (Inverting Terminal) then the output from the respective OPAMP is V_{CC} which is 5V.

Hence Arduino is programmed with logic to encode the input from 8 digital pins (8 bit to 3 bit Encoder).

[Here](#) is the code for the logic behind the encoder.

[Link](#) to the Sampler and ADC Setup.

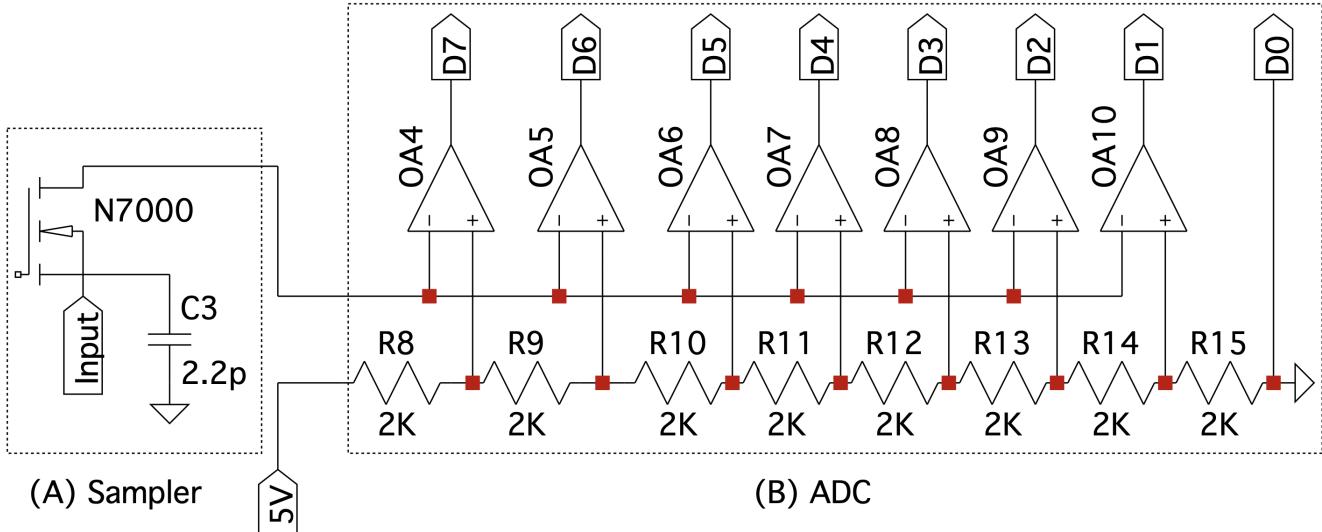


Figure 3

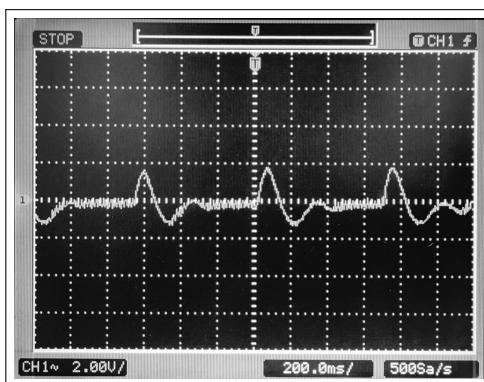


Figure 4

Conclusion :

The values obtained after encoding is plotted using serial plotter. Then the FFT of the obtained wave is calculated. The difference between the peaks are calculated and the frequency is calculated then the heart rate is obtained by multiplying the frequency by 60.

Here is the [Link](#) for the video of obtaining the digital signal from ADC and encoding it and [Link](#) for the excel sheet obtained after encoding.

Supporting Materials

- 1) [Drive Link Containing all the Videos and Pictures](#)
- 2) [Github Repository](#)
- 3) [LM324N Pin Configuration](#)

References

- 1) <http://www.ee.iitb.ac.in/stallur/wp-content/uploads/2017/02/Heart-Rate-Measurement-using-PPG1.pdf>
- 2) <https://github.com/udiboy1209/heart-rate-monitor>