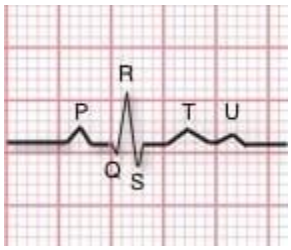
Heart Rate Monitor Project using Arduino Uno and AD8232 EKG Module

Introduction

The purpose of this project is to develop a heart rate monitor using an Arduino Uno and an AD8232 EKG module. This system captures the heart's electrical activity, processes the data, and provides real-time visualizations along with various heart rate variability (HRV) metrics.

What is ECG?

An ECG is a paper or digital recording of the electrical signals in the heart. It is also called an electrocardiogram or an EKG. The ECG is used to determine heart rate, heart rhythm, and other information regarding the heart's condition. ECGs are used to help diagnose heart arrhythmias, heart attacks, pacemaker function, and heart failure.



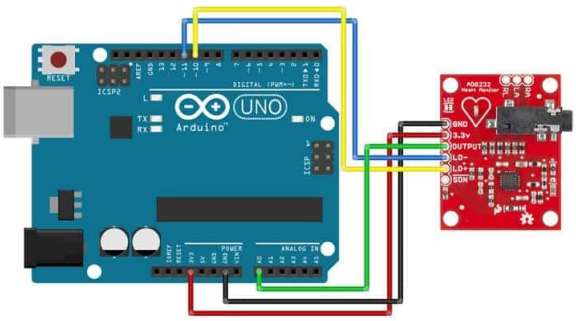
Components

The following components were used in this project:

- Arduino Uno
- AD8232 EKG Module
- Connecting wires

Circuit Diagram/Connection between Arduino and ECG Sensor AD8232

The AD8232 Heart Rate Monitor breaks out nine connections from the IC. We traditionally call these connections "pins" because they come from the pins on the IC, but they are actually holes that you can solder wires or header pins to.

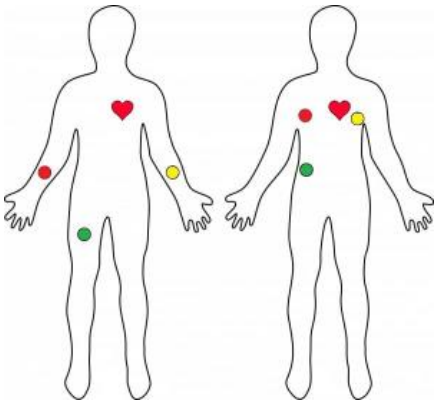


We'll connect five of the nine pins on the board to Arduino. The five pins you need are labeled GND, 3.3v, OUTPUT, LO-, and LO+.

Board Label	Pin Function	Arduino Connection
GND	Ground	**GND**
3.3v	3.3v Power Supply	**3.3v**
OUTPUT	Output Signal	**A0**
LO-	Leads-off Detect -	**11**
LO+	Leads-off Detect +	**10**
SDN	Shutdown	**Not used**

AD8232 ECG Sensor Placement on Body

It is recommended to snap the sensor pads on the leads before application to the body. The closer to the heart the pads are, the better the measurement. The cables are color-coded to help identify proper placement.



COLOR	NAME
RED	RA(Right Arm)
YELLOW	LA(Left Arm)
GREEN	RL(Right Leg)

Arduino Code

The Arduino code captures the analog signal from the AD8232 EKG module and transmits it via serial communication to the computer for further processing.

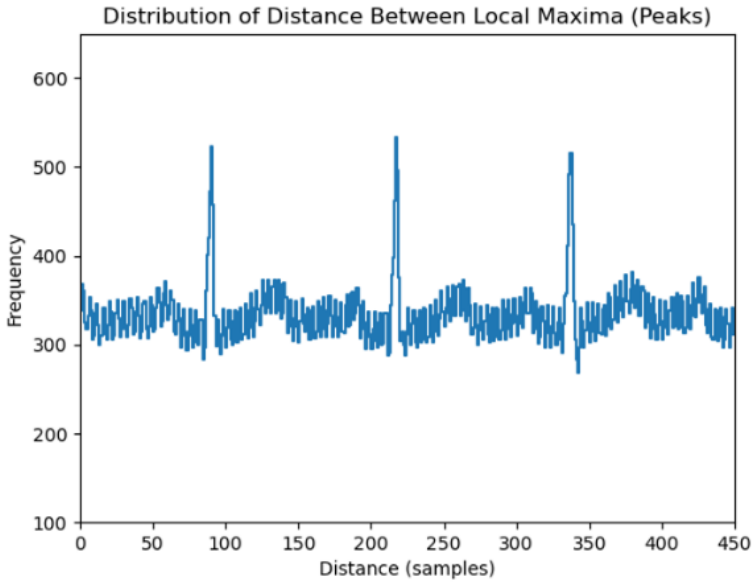
```
void setup() {
  // Initialize the serial communication
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(A0));
  delay(4);
}
```

Data Processing and Visualization in Python

The data received from the Arduino is processed and visualized in real-time using Python. The following steps outline the data processing and visualization stages.

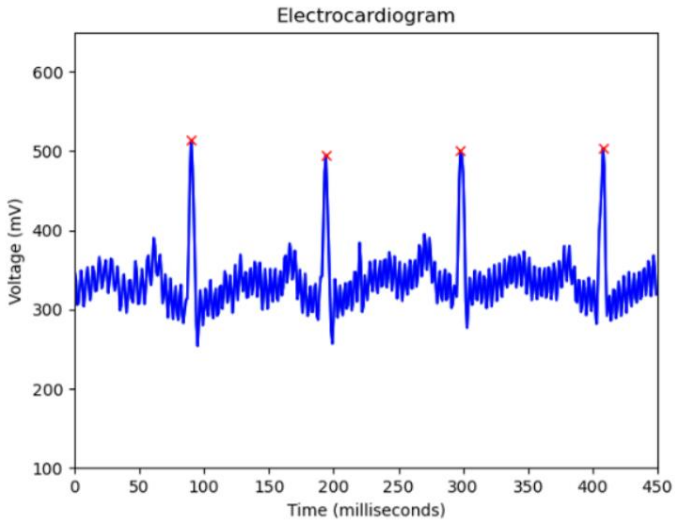
1. Real-Time EKG Signal with Noise

Initially, the captured signal is noisy. Below is a real-time representation of the noisy EKG signal:



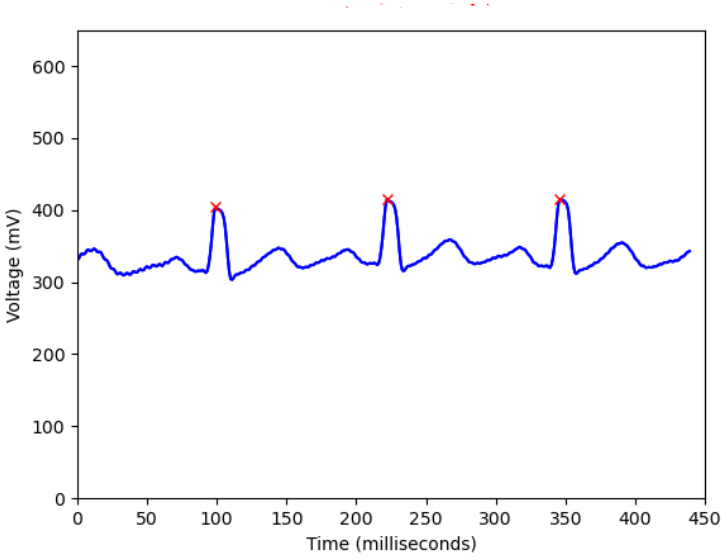
2. R-Peaks Detection

By analyzing the signal, the R-peaks, which are crucial for heart rate calculation, can be detected. The detected R-peaks are marked on the EKG signal as shown below:



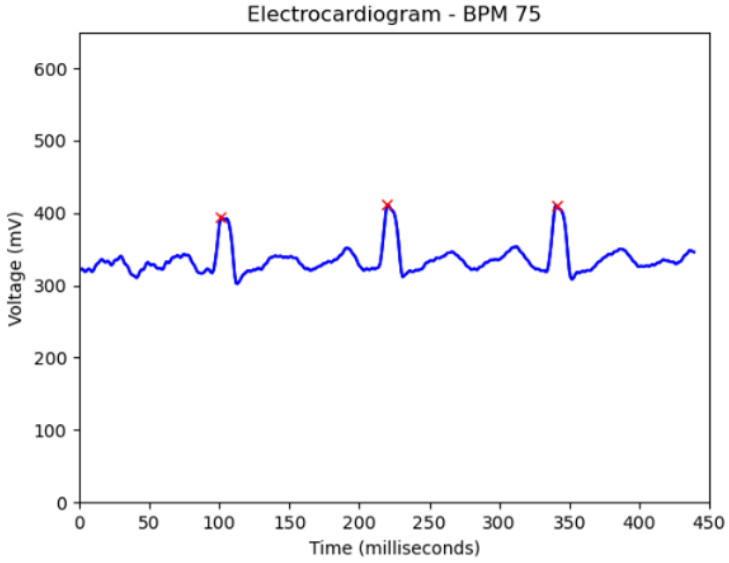
3. Signal Filtering

To reduce noise, a moving average filter and a notch filter are applied to the signal. The filtered signal is shown below:



4. Real-Time Heart Rate Calculation

The heart rate (in beats per minute, BPM) is calculated in real-time from the filtered signal. The real-time BPM is displayed on the EKG plot:



5. HRV Metrics Calculation

Further analysis involves calculating various HRV metrics such as SDNN, RMSSD, VLF Power, LF Power, and HF Power. These metrics help in assessing heart rate variability and overall cardiac health. The values are evaluated to determine if they fall within normal ranges.

SDNN (Standard Deviation of NN intervals)

SDNN is a measure of the variability of the time interval between heartbeats (NN intervals). It reflects the overall variability in the heart rate and is used as an indicator of autonomic regulation of the heart. Higher SDNN values typically indicate healthier heart function and greater variability.

Normal Range: 30-50 ms

RMSSD (Root Mean Square of Successive Differences)

RMSSD is a time-domain measure of heart rate variability. It calculates the square root of the mean of the squares of successive differences between adjacent NN intervals. RMSSD is primarily used to assess the parasympathetic activity of the heart. Higher RMSSD values indicate higher parasympathetic activity and better cardiovascular health.

Normal Range: 20-50 ms

VLF (Very Low Frequency) Power

VLF power is a frequency-domain measure of heart rate variability. It represents the power of heart rate oscillations within the very low-frequency range (0.003 to 0.04 Hz). VLF power is influenced by thermoregulation, the renin-angiotensin system, and other long-term regulatory mechanisms.

Normal Range: 0.003-0.04

LF (Low Frequency) Power

LF power measures the power of heart rate oscillations within the low-frequency range (0.04 to 0.15 Hz). It reflects a mix of sympathetic and parasympathetic nervous system activity. Higher LF power is generally associated with increased sympathetic activity.

Normal Range: 0.04-0.15

HF (High Frequency) Power

HF power represents the power of heart rate oscillations within the high-frequency range (0.15 to 0.4 Hz). It is associated with parasympathetic (vagal) activity and respiratory influences on heart rate. Higher HF power indicates higher parasympathetic activity and better autonomic function.

Normal Range: 0.15-0.4

```
Time to record (minutes) >> 0.5
Starting
Data captured
Enter the filename: ac17
Registered 72 beats per minute.
SDNN: 17.607 ms - abnormal
RMSSD: 19.176 ms - abnormal
VLF Power: 0.023 - normal
LF Power: 0.124 - normal
HF Power: 0.176 - normal
Save images? (y = yes, n = no): y
```

Conclusion

This project successfully demonstrates the use of an Arduino Uno and an AD8232 EKG module to monitor heart rate and analyze HRV metrics. The Arduino captures the EKG signal, which is then processed and visualized in real-time using Python. The system provides valuable insights into heart rate and variability, helping in the assessment of cardiac health.

Future Work

Future improvements could include enhancing the accuracy of the signal processing algorithms, integrating additional HRV metrics, and developing a user-friendly interface for easier monitoring and analysis.

References

- AD8232 Datasheet - *https://www.alldatasheet.com/datasheet-pdf/pdf/527942/AD/AD8232.html*
- *ECG Graph Monitoring with AD8232 ECG Sensor & Arduino - https://how2electronics.com/ecg-monitoring-with-ad8232-ecg-sensor-arduino/infor*