ECG signal capture using Olimex ECG-EMG shield and Arduino UNO

Tomislav Mamić for MIKROTRON d.o.o.

Zagreb, January 10th 2016.

Basic idea

Olimex EKG-EMG shield (link) is a cheap solution for measuring electrical signals on skin surface. It consists of 3 electrodes - positive, negative and neutral, hooked up to a differential amplifier. One output signal is available - the difference between the positive and negative leads. Furthermore, the device contains a hardware band pass filter with cutoffs at 0.16Hz i 40Hz. This document aims to explore the posibilities and limitations of the aformentioned shield in combination with an Arduino UNO.

Initial thoughts

Since shield output is an analog signal, it will be sampled by Arduino UNO's built-in ADC. The processor used is an ATmega328. The ADC offers 1024 levels at a maximum frequency of 15kHz (this is overclockable, but not very useful here). The shield contains a variable resistor which allows for some control over the amplification coefficient. For testing purposes, I leave this resistor at factory setting.

Sampling frequency is to be chosen in such a way that the spectrum contains as much information as possible while still giving enough time to the processor to run its step code. Since the maximum theoretical frequency contained in the signal coming from the shield is 40Hz, it is, again theoretically enough to sample at 80Hz (thank you mr. Nyquist). However, it is impossible to make a sharp cutoff analog filter. For that reason, oversampling is a very good idea. Due to the fact that the amount of calculation ATmega328 has to do is rather limited, that further noise enters the system from the filter to the ADC and that the ADC itself introduces high frequency noise, I chose to oversample significantly, at 250Hz.

Characteristics of ECG signal

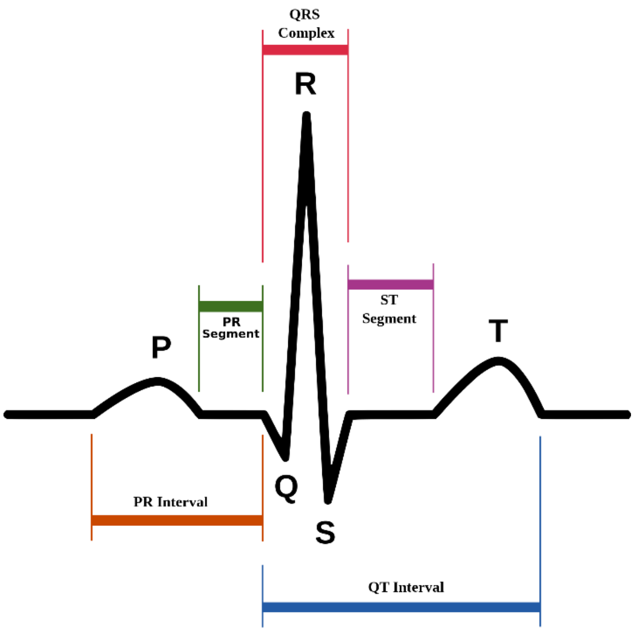

Image 1 - PQRST complex

Useful frequency content of a normal ECG signal is described in detail in [1]. Ideally, the lower cutoff limit is at around 0.05Hz with possible relaxation up to 0.67Hz for certain applications, though doing so will introduce deformation in the ST segment. The upper cutoff should not go below

100Hz for adults or below 250Hz for infants. Upper cutoff frequency as recommended by the ANSI/AAMI 2001 standard is 150Hz. Since we can't change the characteristics of the hardware filter built into the shield, it's clear that our ECG will be subpar.

Method of measurement

Potential difference between left and right forearm is measured, according to the image below. Raw signal, as well as an additionally filtered variant are taken. The raw signal is digitally filtered by Arduino through a 4th order low pass Butterworth filter with 45Hz cutoff. DC level is being calculated in parallel by an exponential smooth filter at 0.05Hz cutoff. Its output is subtracted from the filtered signal to counter baseline drift.

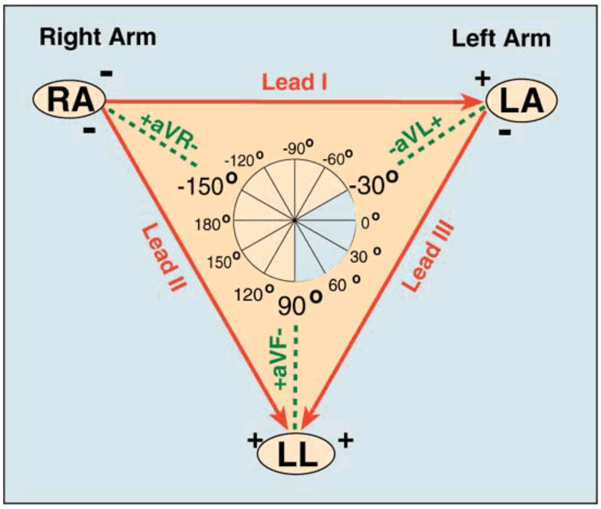

Image 2 - Einthoven's triangle

Signal processed in this way is good for pulse measurement based on QRS detection. QRS detection is based on the fact that the segment between points Q and R is the fastest rising portion of the signal together with R being typically the highest amplitude. Two conditions are implemented to recognize the QRS complex. First condition is that a signal is higher than some fixed threshold. This alone is not good enough, so the second condition states that while above the threshold, slope of the signal must be above a certain value as well. Both detection thresholds adapt to the signal - they are calculated by feeding absolute values of the signal and its 1st derivation respectively through a low pass filter, amplified by empirically determined scaling coefficients. Those coefficients were roughly determined during testing to produce the best recognition rate over around 100 test examples. Their values are set to roughly 3 for the amplitude and 2 for slope. Using this method of detection has proven quite reliable over a large data set taken from http://www.physionet.org/cgi-bin/atm/ATM. The MCU was set to send sets of four data points: time, raw ADC value, filtered signal and a logic signal whose value is 0 at all times except at the moment a QRS complex is observed.

Further processing on the PC

It is important to note right away that this functionality can be implemented on an MCU, but due to memory limitations of the Arduino UNO, it was not done here. There are, however, accessible alternatives which could run these computations.

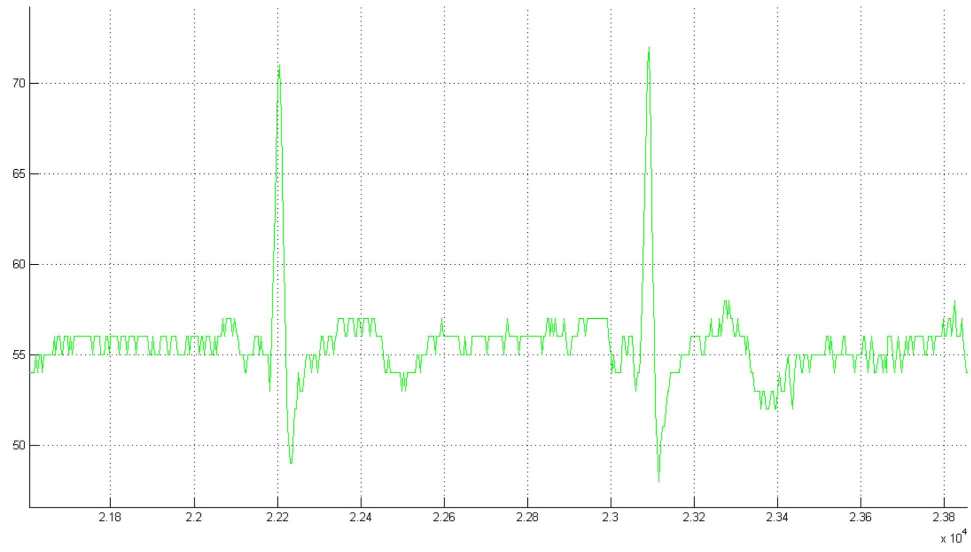

Image 3 - signal from the ECG-EMG shield

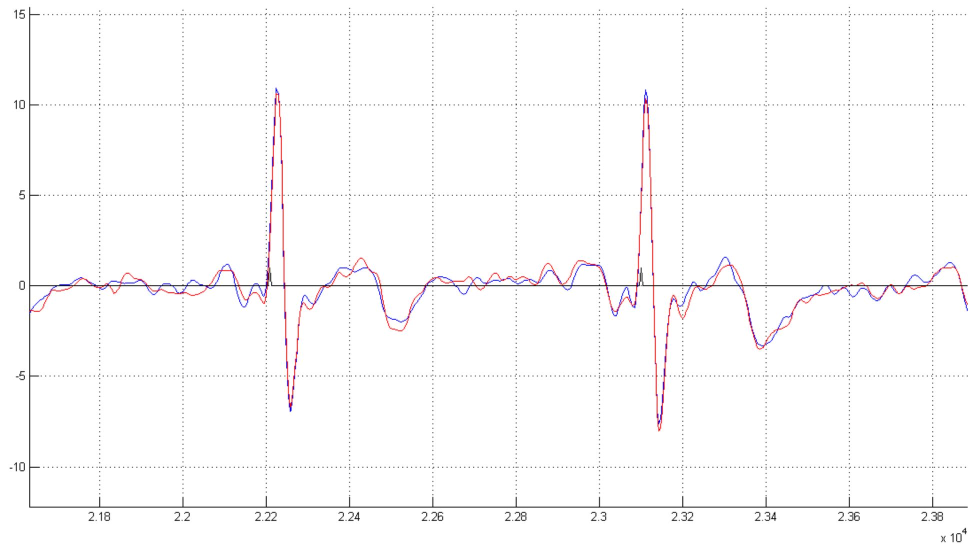

Image 4 - filtered signal

Images 3 and 4 show measurements captured by the prototype. It is worth noting that raw values span over barely 25 ADC levels, which greatly increases the effects of high frequency noise introduced by the ADC. A step in the right direction would be increasing the hardware amplification coefficient by changing the variable resistance on the shield. But for now, time parameters - durations of PQRST features are of greater interest. It turns out that too low upper cutoff frequency smears the signal in time, making these measurements a challenge. This is partially solvable for the QRS feature because its properties are easy to guess, so the error can be approximated, but the damage done to T segment is significant. These conclusions are based on measurements done on four adults - three men and one woman. None of the subjects have any history of heart problems.

Generating a typical PQRST complex

It is useful to generate a representative PQRST complex for a patient to remove all kinds of noise from the signal. This is done using QRS detection method to cut and align many PQRST complexes from a measurement and taking their average. The method of implementation is suitable for an MCU with sufficient memory storage, with delay from real time of only a couple hundred milliseconds.

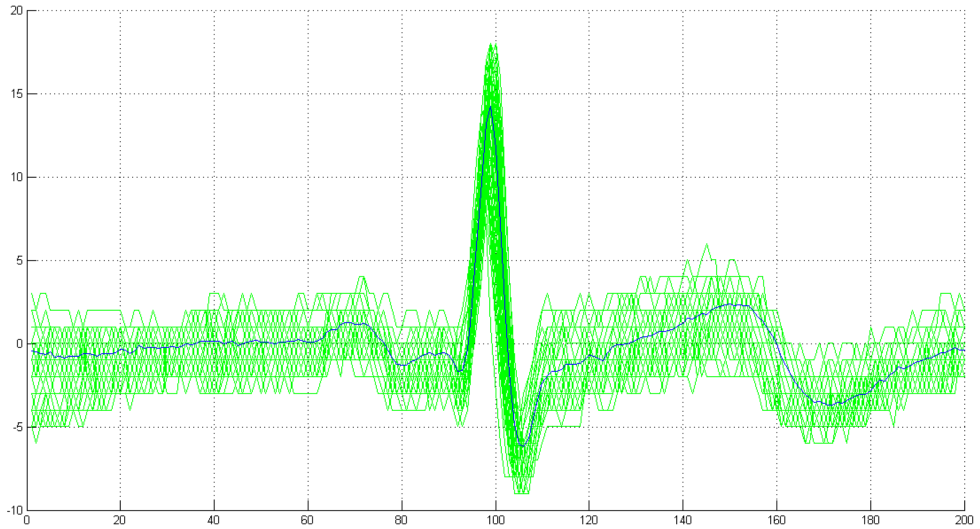

Image 5 - overlapping multiple PQRST complexes

Image 5 shows a representative PQRST complex extracted from 73 PQRST complexes caught over a period of 60 seconds. Parameters of the extracted complex are shown in the table below.

	measured	typical
P	0.06	<0.12
PR interval	0.12	0.12-0.2
QRS	0.08	0.06-0.1
ST interval	0.288	0.38-0.42*

*for 60-80 beats per minute

It is visible from the table and Image 5 that the T feature suffers a significant distortion. This distortion is observed in all measurements thus far which leads me to conclude that it is indeed a consequence of the device itself rather than someone's peculiarity.

Conclusions

An expert's opinion on the usefulness of this device is needed. Unreliability of time parameters, as well as poor amplitude range are the main problems observed. It is possible to apply very powerful classification and pattern recognition techniques for diagnostic purposes, but if the quality of measurements is poor, there is little purpose in doing so. A more thorough analysis of how exactly the signal is distorted is needed, but within one PQRST complex, time measurements are rather unreliable.

Sources

[1] Recommendations for the Standardization and Interpretation of the Electrocardiogram - Kligfield et al. - *Circulation 2007*
http://www.gammacardiosoft.it/openecg/AHA%20reccomandation%20of%20standardization%20of%20ECGCirculation-2007-Kligfield-1306-24.pdf

[2] ECG learning center, University of Utah School of Medicine - Frank G. Yanowitz, M.D
http://ecg.utah.edu/lesson/3