Levy Gabriel da Silva Galvão

Experimental data acquisition and processing system for ECG signals

Natal - RN
October, 2021

Abstract

Regarding the importance of the Electrocardiograph (ECG) instrument, this work tends to develop an experimental setup for a data acquisition system to collect ECG signals. Also all steps in the development are explicated so readers and scholars can base their prototypes with high level of details in the system specification and sizing. This platform is built with an ECG simulator in software and hardware domain; an analog front-end to adequate the acquired ECG signal from the source; and a digital interface with an ESP32 controller to post process the signal, analyse heart rate and pathologies and transmit the final ECG to other means of visualization. Also many software routines are developed so tests can also be performed in a high level of abstraction.

Keywords: ECG. DAQ. Pathology.

List of Figures

List of Tables

Contents

1 Introduction

An electrocardiograph (ECG) is an instrument responsible to record the electrical activity of the heart, considering that electrical signals are generated before the heart mechanical functions and generated by nerve impulse stimulus. In this context the ECG provides information about those signals that allows to detect a variety of cardiac disorders [1, 7].

The ECG machine has became an essential instrument throughout the years once it is a noninvasive, simple to record and minimal cost device [2]. Regarding its importance, the objective of this work is to design an experimental setup for an ECG machine, highlighting all project steps so anyone can learn experiment with a simple and accessible device for the analysis of cardiograph signals.

The first step of the ECG design is to get to know how the cardiac signal is generated, its specifications and the noises associated with the data acquisition.

Once define those guidelines, the next step is to develop a reliable source of cardiac signals, both in the software and hardware domain, also known as a simulator. A simulator from the software point of view can be used to learn how to process the signal with multiples digital signal processing algorithms in a higher level of abstraction. From the hardware point of view there is a need to generate a cardiac signal in the analog domain so it can be acquired by the proposed data acquisition system (DAQ). For the latter a consolidated ECG device could be used, but once the project relies in the simplicity of it resources, a simple circuit serves its purposes.

The third step concerns the ECG design itself. The sub steps are:

- Design the low noise preamplifier(instrumentation amplifier), since the acquired signal by the electrodes has low voltage;
- Design a low-pass analog filter serving as an antialising filter to avoid out of band noise contamination after the signal is digitized;
- Design a high-pass filter to eliminate the DC component and any baseline wander;

- Design a Notch filter (reject band filter) centered in 60 Hz to remove any interference from the power grid;
- Data acquisition by an analog-to-digital converter (ADC);
- Program the micro controller interface between the analog front end and the digital part responsible for analyzing, processing, storing and displaying the final graphical output.

Where the first five sub steps integrate the analog front end that is responsible for conditioning and digitizing the signal.

The final step is to validate the system with experiments and discuss the results.

2 Specifications

2.1 Cardiac signal

The electrical potentials of the cardiac signal are acquired by various electrodes connected in the surface of the skin of the patient. Those combined generate the cardiac signal, as can be seen in the normal wave pattern of figure 1, with voltage differences in the order of $1\,mV$ between given points in the body [9].

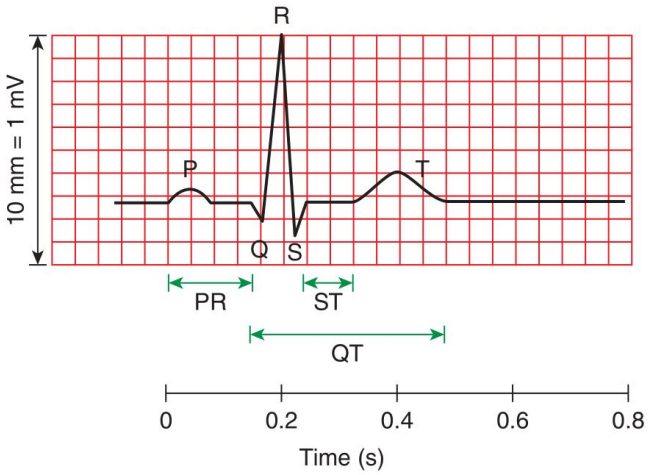

Figure 1: Normal waveform pattern of cardiac signal obtained in ECG. Source: Khandpur [9].

Regarding the figure 1, each letter has a proper meaning for each step in the cardiac cycle. Those are:

- The P wave represents the depolarization of the atrial muscles;
- The QRS section is a combination of the atria repolarization between QR and the ventricles depolarization between RS;
- The T wave represents the repolarization of both ventricles;

The interval PR represents the actrial systole, of which the diastole lasts from R until the next P wave. The ventricular systole occurs between R until the end of the T wave and its diastole lasts until Q [14].

The interval QRS representing the time taken by the heart impulse to travel from the inter ventricular system then through the walls of the ventricles is the most critical one when thinking in the design of an ECG machine since it has the higher frequency of all the cardiac signal and lasts about 0.05 to 0.1 seconds [9].

Those characteristics leads to a deeper specification of the ECG regarding the signal. Khandpur [9] defines the frequency range of the signal varying from 0.05 to 150 Hz. By the Nyquist sampling theorem, it is recommended that the sampling frequency used in digitization to be at least two times the higher frequency in the signal, i.e 300 samples/s. Despite this, Khandpur [9] exerts that a sampling rate of 200 samples/s is satisfactory, yielding 12 to 20 samples for the QRS interval. Therefore the system designed in this work will attend to the Nyquist criteria and beyond, using a sampling frequency of 500 samples/s. The disadvantage in this approach is that compared to the sampling frequency proposed by Khandpur [9], the present system will need 2.5$\times$ more space due its higher sampling frequency.

Regarding the bit resolution to store the cardiac signal, Khandpur [1] suggests two approaches: to use low-noise and high-gain amplifiers, enabling the use of low-resolution 16-bit ADC; or using a low-gain amplifier with a high-resolution 24-bit ADC. The first approach will be used in this project for the sake of a more elaborated amplifier design and signal conditioning, instead just using a better ADC.

Since the arrangement of electrodes is not the main focus of this work, the system will stick with a bipolar leads arrangement. This arrangement uses two electrodes placed in the right and left arm to capture the signal and send to the input of an instrumentation amplifier and other electrode as reference placed in the left leg [9]. This means that there is no need for a multiplexing system for multiples channels of electrodes, like in a typical 12-lead ECG [6].

2.2 Associated noises

In the chain of processing, the cardiac signal must be filtered in a way to remove certain noises and interference inherent to the data acquisition. The most common source are: interference from the power-lines (power grid) as a tone in $50/60\,Hz$ (depending on the region of the globe); noise generated mechanically due to the contact between electrodes and skin, motion artefacts firing random derived from the patient movement and muscle contraction (voluntary or involuntary); additive white Gaussian noise (AWGN) derived from thermal sources; or electromagnetic interference from other electronics devices that can extends to the RF spectrum or higher [1].

It is essential to a ECG instrument to maintain clear of those noises and interference in a level of approximately $10\,\mu V$ peak to peak to ensures ECG applications in diagnostics [1].

Each source of noise and interference can be treated in a specific manner.

Power-line interference can be solved designing a Notch filter (reject band) with cutoff frequency set to $50/60\,Hz$ (this project will validate the filter to $60\,Hz$).

Baseline wanders and muscle contraction are phenomena of low frequencies and they can be eliminated with the help of a high-pass filter. In the previous section the lower frequency defined to the cardiac signal was $0.05\,Hz$, so a high-pass filter with this value as cutoff frequency should be able to remove those interference without risking rising too much the cutoff and attenuating the P or T waves [9, 1, 7]. Sahin, Fidel, and Perez-Castillejos [11] suggests the use of a $0.5\,Hz$ cutoff frequency in a fourth order Butterworth filter, therefore allowing the use of a simpler filter with relaxed requirements and alerting that this cutoff may be tuned to meet the requirements. The $0.5Hz$ cutoff frequency will be used in this project.

AWGN and electromagnetic interference are phenomena of mainly high frequencies, therefore can be eliminated with a low-pass filter that might be designed with the anti aliasing filter. The anti aliasing filter is a analog

and low order filter, but can also be used to eliminate electromagnetic interference since it has a frequency band way higher than the signal. The AWGN is uniform thought out all frequencies, so the previous high-pass filter will contribute to eliminate some of the noise. The cutoff frequency selected for this filter will be $200\,Hz$ which comprises the signal band.

An alternative idea is to merge both low-pass and high-pass filters into a band-pass filter. The problem associated with this strategy is that it is not possible to select different filter orders to the low-pass and high-pass segment. Even though this project will stick with a band-pass filter combining the high-pass and low-pass filter cutoff frequencies, thus simplifying the circuit.

Below there is table 1 showing a summary of the analog filters to be designed in further sections.

Filter	Cutoff frequency
Notch (analog)	$60\,Hz$
Band-pass (analog)	$0.5\,Hz$ to $200\,Hz$

Table 1: Summary of filters.

3 Methodology

3.1 System overview

The idealized system follows the sketch of the figure 2

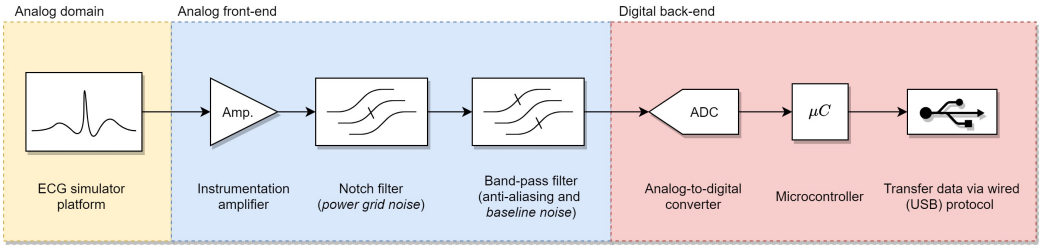

Figure 2: Idealized system for acquiring, processing and transferring ECG data.

The analog front-end contains the preamplifier and filtering. The digital part involves the ADC, signal processing and transmission.

The system analog front-end was entirely tested via the free circuit simulator software LTSpice and all plots were made with the use of the Python Plotly library, since the graphs presented in LTSpice does not have a good quality.

The digital part was simulated with the use of an ECG simulator embedded in an ESP32 outputting via a 8-bit DAC and been collected at a 12-bit ADC in another ESP32. The data was sent to a PC via USB serial connection to by displayed graphically.

3.2 Preamplifier

The amplifier architecture used is an instrumentation amplifier of which resembles the topology of a differential amplifiers but with the addition of buffers on each input. The complete circuit can be seen in the figure 3 below.

The output of the instrumentation amplifier can be written as equation 1:

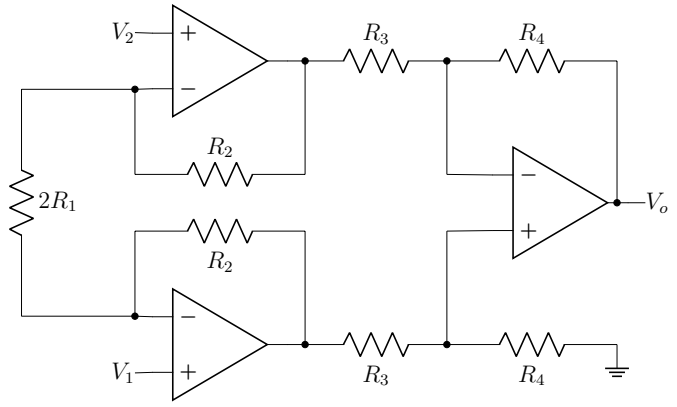

Figure 3: Instrumentation amplifier.

$$V_o = \mathrm{DG}(V_1 - V_2) + \mathrm{CMG}\left(\frac{V_1 + V_2}{2}\right) \tag{1}$$

Where DG is the differential gain and CMG is the common mode gain given by equations 2 and 3:

$$\mathrm{DG} = \frac{R_4}{R_3}\left(1 + \frac{R_2}{R_1}\right) \tag{2}$$

$$\mathrm{CMG} = \left(\frac{R_4}{R_3 + R_4}\frac{R_1 + R_2}{R_1} - \frac{R_2}{R_1}\right) \tag{3}$$

In ideal condition the common mode gain is null with the traditional choice of $R_1 = R_3$ and $R_2 = R_4$, and the final output turns to be $V_o = \mathrm{DG}(V_1 - V_2)$.

The problem is that CMG will never be null due to imprecision in the resistance value of the resistors. Despite this, there is a metric called Common Mode Rejection Ratio (CMRR) given by the equation 4 in dB that indicates how much times higher the differential gain should be relative to the common mode gain. In the case of ECG, this ratio must be higher than $+100\,dB$ [9, 3].

Also a input impedance of at least 10 $M\Omega$ in the input buffers is recommended [1].

$$\text{CMRR} = 20\log\left(\frac{\text{DG}}{\text{CMG}}\right) \tag{4}$$

Usually the differential gain for the preamplifier in a ECG is 500 [9]. A better approach would be to distribute this gain along a multistage amplifier to avoid distortion due to non linearities [12]. Despite this statement, the project will be based in a single amplification block for the sake of simplicity, .

The differential gain for the project should be close to 506 with the resistors $R_2 = R_4$ with $22\,k\Omega$ resistors and the resistors $R_1 = R_3$ with $1\,k\Omega$ resistors.

Considering the gain, the choice of resistors can be illustrated by the table 2. Also the operational amplifier LMH6629 [5] was chosen once it is SMD, has ultra-low noise and has a slew rate of $1600\,V/\mu s$ that can follow the variation of the QRS wave.

Component	Identifier	Number of components
Resistor	$22\,k\Omega$	4
Resistor	$1\,k\Omega$	4
Opamp IC	LMH6629	3

Table 2: Summary of electronic components for the preamplifier.

3.3 Analog filters

3.3.1 Notch filter

The Notch filter topology used in this project follows the circuit of figure 4. Besides implementing the low-pass and high-pass segment, this filter has a buffer so it isolates from the output and at the end, the output is fed back

fractioned to adjust the filter's quality factor (Q).

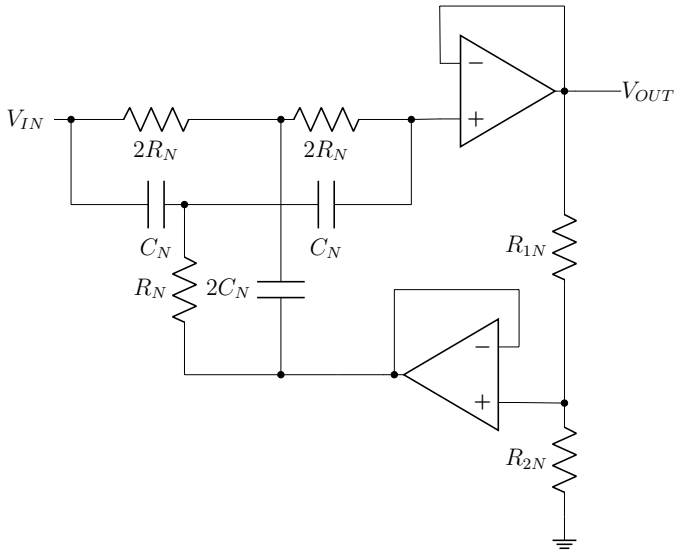

Figure 4: Notch filter.

The cutoff frequency of the previous Notch filter can be found by the equation 5. Keeping in mind that the cutoff frequency is $60\,Hz$, the capacitor and resistor could be $100\,nF$ and $13.2\,k\Omega$. Once the found resistance does not match with a commercial value, the resistor R_N can be used with a series association of a $13\,k\Omega$ and $200\,\Omega$ resistor.

$$f_c = \frac{1}{4\pi R_N C_N} \tag{5}$$

The quality factor (Q) is a function of the cutoff frequency and the bandwidth of the filter, given by the equation 6 where B_W is the bandwidth in Hertz. Setting a bandwidth of 1 Hz with reject band between 59.5 and $60.5\,Hz$, which is a band that the power grid voltage can assume in the worst case during anomalies according to ANEEL [4]. Henceforth the quality factor is found to be $Q = 60$ and the remaining resistor can be found by the equation 7 as $R_{1N} = 180\,\Omega$ and $R_{2N} = 47\,k\Omega$.

$$Q = \frac{f_c}{B_W} \tag{6}$$

$$1 - \frac{1}{4Q} = \frac{R_{2N}}{R_{2N} + R_{1N}} \tag{7}$$

The table 3 summarizes the components choices for the Notch filter.

Component	Identifier	Number of components
Resistor	$47\,k\Omega$	1
Resistor	$180\,\Omega$	1
Resistor	$13\,k\Omega$	5
Resistor	$200\,\Omega$	5
Capacitor	$100\,nF$	4
Opamp IC	LMH6629	2

Table 3: Summary of electronic components for the Notch filter.

3.3.2 Band-pass filter

Once stated before, the high-pass and low-pass filter were merged into a single band-pass filter. One possible topology for the band-pass filter would be merged a cascade association of a high and low-pass filter, but this choice wastes improvement possibilities. Instead the topology of choice is the one in the figure 5

The filter stated above is an inverting active band-pass filter. Its equations are: equation 8 to select the gain between input and output, which in this case will be set to one, resulting in $R_{B1} = R_{B2}$; equation 9 to select the capacitance of C_{B1} based in the lower cutoff frequency ($f_L = 0.5\,Hz$); and equation 10 regarding C_{B2} based in the higher cutoff frequency ($f_H = 200\,Hz$).

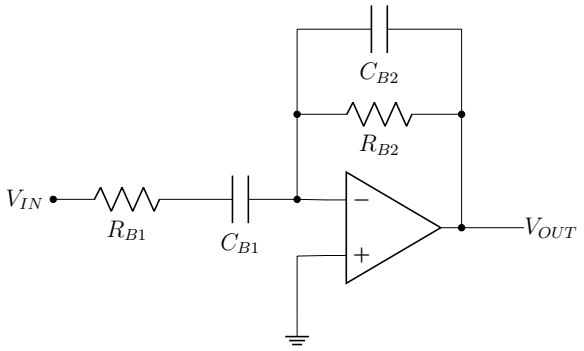

Figure 5: Band-pass filter.

$$\text{Gain} = A_v = -\frac{R_{B2}}{R_{1B}} \tag{8}$$

$$f_L = \frac{1}{2\pi R_{B1} C_{B1}} \tag{9}$$

$$f_H = \frac{1}{2\pi R_{B2} C_{B2}} \tag{10}$$

An inverting amplifier with unitary gain can be placed in the output of the band-pass filter to correct the polarity of the resulting signal, or instead just inverting when taking the output between V_{OUT} and ground node.

The values found for the components are summarized in the table 4, considering $R_{B1} = R_{B2} = 500\,k\Omega$, $C_{B1} = 636\,nF$ and $C_{B2} = 1.59\,nF$. The capacitor C_{B1} was broken into a parallel association of four capacitors $0.22\mu F$, $0.22\mu F$, $0.18\mu F$, $0.015\mu F$ and $0.001\mu F$. The same for C_{B2} but with a parallel association of $680pF$, $680pF$, $220pF$ and $10pF$.

Component	Identifier	Number of components
Resistor	$500\,k\Omega$	2
Capacitor	$0.22\,\mu F$	2

Component	Identifier	Number of components
Capacitor	$0.18\,\mu F$	1
Capacitor	$0.015\,\mu F$	1
Capacitor	$0.001\,\mu F$	1
Capacitor	$680\,pF$	2
Capacitor	$220\,pF$	1
Capacitor	$10\,pF$	1
Opamp IC	LMH6629	1

Table 4: Summary of electronic components for the band-pass filter.

3.4 ECG simulator

Prior to the tests of the digital processing algorithms, there is a need to simulate typical ECG signal with a good model in the software domain. The high level abstraction allow it to debug faster than deploying and testing direct in the hardware.

The model used was the one presented by Quiroz-Juárez et al. [10], whose consists in the use of three different nonlinear oscillators related to the cardiac natural pacemakers where three coupled oscillators represent the action potentials of the SA node, AV node and His-Purkinje complex, as illustrated in the figure 6 extracted from Quiroz-Juárez et al. [10].

Quiroz-Juárez et al. [10] also derive an ODE system of nonlinear equations to represent the given oscillators, as illustrated in the equation 11.

$$\begin{cases} \dot{x_1} = \Gamma_t \cdot (x_1 - x_2 - Cx_1x_2 - x_1x_2^2) \\ \dot{x_2} = \Gamma_t \cdot (Hx_1 - 3x_2 + Cx_1x_2 + x_1x_2^2 + 2\beta(x_4 - x_2)) \\ \dot{x_3} = \Gamma_t \cdot (x_3 - x_4 - Cx_3x_4 - x_3x_4^2) \\ \dot{x_4} = \Gamma_t \cdot (Hx_3 - 3x_4 + Cx_3x_4 + x_3x_4^2 + 2\beta(x_2 - x_4)) \\ \mathrm{ECG}(t) = \alpha_1x_1 + \alpha_2x_2 + \alpha_3x_3 + \alpha_4x_4 \end{cases} \tag{11}$$

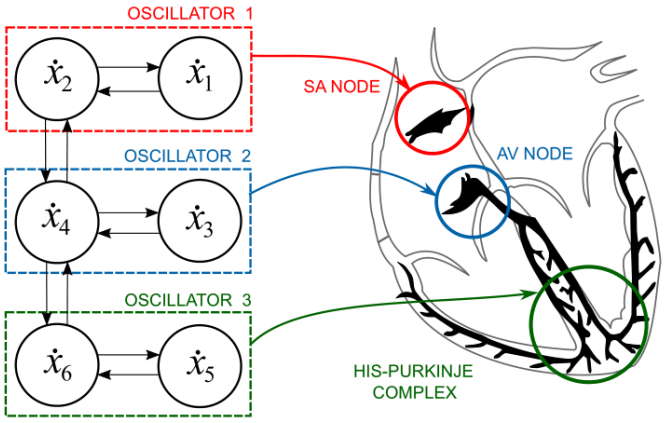

Figure 6: Diagram relating the cardiac natural pacemakers to non linear variables. Source: Quiroz-Juárez et al. [10].

The whole description is left for the original paper, but is known that a correct choice of the coefficients results in changes in the ECG waveform. Quiroz-Juárez et al. [10] comes with a set of coefficients that calibrates the ECG in different modes, varying from the normal cardiac rhythm and also represent pathologies such as: sinus tachycardia, atrial flutter, ventricular tachycardia and ventricular flutter. The table 5 illustrates the coefficients used by Quiroz-Juárez et al. [10] to generate each mode, starting from the initial excitation of $\mathbf{x} = \{0, 0, 0.1, 0\}$.

Mode	H	Γ_t	α_1	α_2	α_3	α_4
Normal rhythm	3	7	-0.024	0.0216	-0.0012	0.12
Sinus tachycardia	2.848	21	0	-0.1	0	0
Atrial flutter	1.52	13	-0.068	0.028	-0.024	0.12
Ventricular tachycardia	2.178	21	0	0	0	-0.1
Ventricular flutter	2.178	13	0.1	-0.02	-0.01	0

Table 5: Coefficients for the ECG simulator to access each mode. The coefficients $C = 1.35$ and $\beta = 4$ are constants along all the simulations. Source: adapted from Quiroz-Juárez et al. [10].

With this representation, a code in Python was created to simulate the ECG in software and further a firmware for the Espressif micro controller ESP32 was also created, allowing the internal 8-bit DAC (digital-to-analog converter) to continuously output the ECG simulation given by the model above. It was also predict a switch button in the micro controller to switch between each pathology. In both cases a fixed sampling period of 10^{-4} was used. The simulation results will be shown in further sections.

The analog front-end simulations has an ECG simulator implemented via piece-wise linear voltage source with a predefined ECG waveform.

actual hardware implementation, those routines were tested in a high level of abstraction in Python.

Before any processing in the micro controller it was implemented an moving average filter of arbitrary size of 16 samples to filter the noise in the incoming signal.

3.5.1 Heart rate detection

The algorithm design to detect the hear rate is a simply detection of the peak in the QRS complex followed by the time distance between QRS peaks. The steps in the calculation is as follows:

- Window a portion of the signal (buffer);
- Set a proper threshold in amplitude to detect the peaks in the QRS complex, it was used a threshold of 90% of the ECG maximum amplitude, but a value of 130%-140% of the RMS value of the buffer might be used;
- Proceed to peak detection where:
  - The ECG signal is evaluated if its amplitude overtakes the threshold value;
  - Considering values that meet the overrun criterion, those are tracked to monitor any amplitude rising;
  - If the amplitude is rising, it is tracked the inflection point where the signal transits from rising to decreasing;
  - Thus the inflection point is the peak of the QRS complex;
- During the peaks detection the time interval between peaks are calculated, thus its inverse giving the heart rate.

The heart rate detection algorithm was first implemented in high level of abstraction in Python using the ECG generator previously stated, so after the tests it can be safely implemented in a hardware manner. The tests results are displayed in the results section.

3.5.2 Pathology analysis

The pathology analysis is based in the results obtained for the heart rate (that involves time intervals, duration and location) out of the expected values of 60 to 100 bpm (considering that the ECG is measured with a patient at rest) and also amplitude extremes and variations [13, 8, 15].

During the test for pathologies it is important to make use of the ECG generator outputting wave forms that are not in the normal rhythm, i.e. sinus tachycardia, atrial flutter, ventricular tachycardia and ventricular flutter.

3.6 ECG embedded acquisition system

The main block of the system is the ESP32 responsible to collect data at a sampling rate of $500\,Hz$, process the signal to extract further information and transmits the data to the ECG viewer.

The data was collected with the ESP32 built-in 12-bit SAR ADC. The linear range of this ADC is adjustable according to the attenuation. Considering an attenuation of 11dB, the possible range of voltages in the ADC input varies from 150 to 2450 mV. The DAC output was configured so the voltage does not fall of this range, causing distortion due to non-linearities in the ADC.

The firmware code structure is based in the freeRTOS, using tasks and event groups to handle the tasks access. In total there are five tasks in the code, each one responsible for:

- Start the data acquisition;
- Finish the data acquistion and returning to idle mode, waiting for another session to start;
- Record data from ADC into a buffer;
- Apply the digital signal processing algorithms previous discussed in the data inside the ADC buffer and store in another buffer (DSP buffer);
- Finally, transmit via USB serial the content inside the DSP buffer;

Once the ESP32 has two cores, the ADC task was assigned to the core number 1 and all other were assigned to the core 0. This allows for the data to be acquired continuously without interference of another task.

Despite the efforts to feed the DAQ via the ESP32 ECG simulator, due to lack of resources, the signal is not well conditioned, so the solution for the simulations is to merge the both the simulator and the DAQ in a single micro controller to test the signal processing algorithms. Noise was inserted in the generation to simulate the conditions in a real acquisition.

3.7 ECG viewer

The ECG viewer platform was design as a Python application running in a PC that receives serial data from the ESP32 DAQ and plots the ECG amplitude in real-time.

Acquired data is also stored in a `.csv` file.

4 Results

4.1 Analog front-end

The results of the analog front-end (AFE) involves the outputs of the ECG simulator/generator, the preamplifier section and the analog filters, i.e Notch filter and band-pass filter, where the latter is also the output of the AFE.

Initially the figure 7 displays the outputs for each stage in the AFE. It is important to note how the amplitude and its ripples varies by each output.

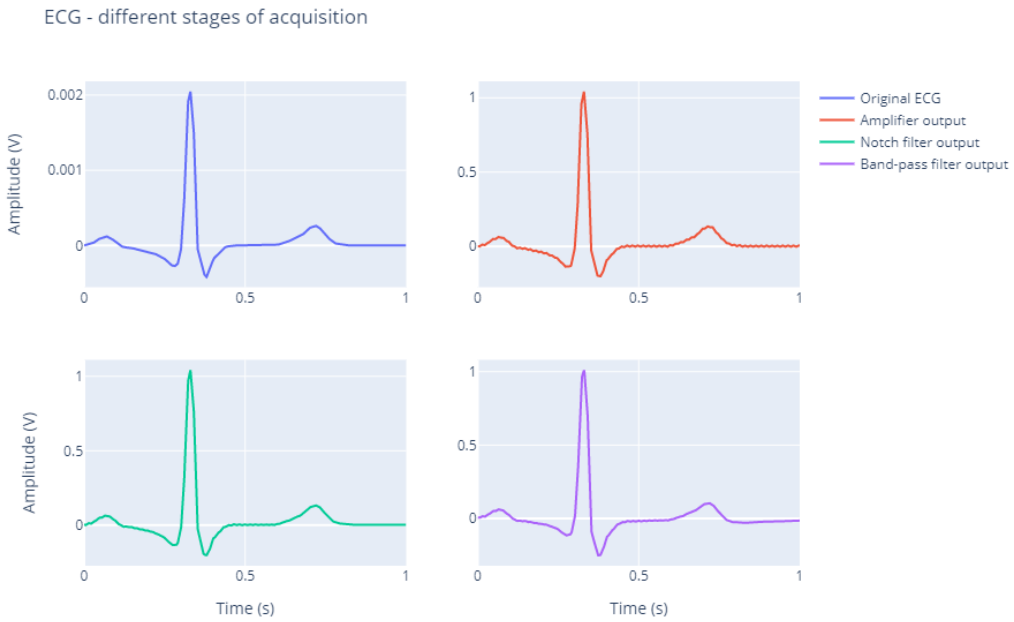

Figure 7: AFE outputs in time domain from LTSpice.

The ECG simulator generates a low voltage amplitude in mV range, as expected, since the bio potentials measured through the skin follows this behavior. Right after the preamplifier output, the ECG signal reaches an appreciable amplitude, close to a $1\,V$ peak, but the $60\,Hz$ ripples remains. Those ripples are attenuated after the Notch filter. At the end, the band-pass filter output does not show great changes in the time domain. A closer look

comparing the input and output of the AFE is presented in the figure 8.

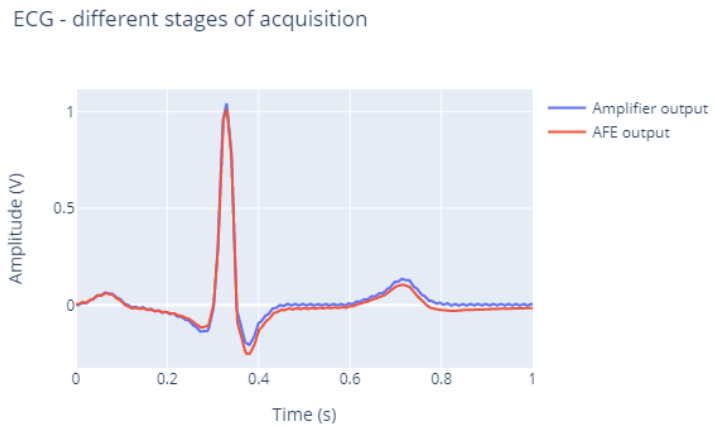

Figure 8: Closer look in the AFE outputs in time domain from LTSpice.

Applying the discrete Fourier transform in the LTSpice it can achieve the frequency response of each output, thus revealing more information, like displayed in the figure 9.

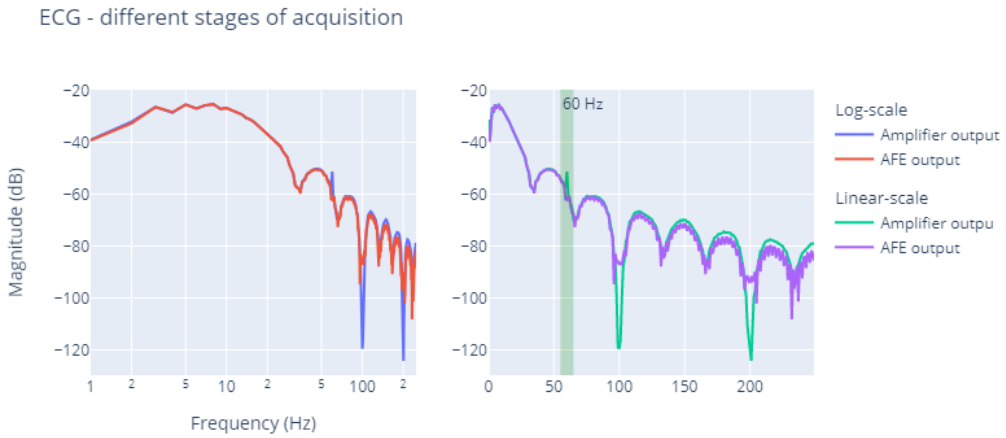

Figure 9: AFE outputs in frequency domain from LTSpice.

To avoid redundancies in the plots it will only by displayed the amplifier and AFE output (band-pass filter output). This plot shows the direct effect

of the Notch filter, reducing the magnitude of the 60 Hz component. Also is identifiable the band-pass filter result, showing a higher attenuation after the 200 Hz higher cutoff. Thus ending the AFE analysis.

4.2 Simulator

Once the ECG simulator model presented by Quiroz-Juárez et al. [10] was tested via software and hardware, thus representing two different kinds of plots.

The figure 10 shows a comparison between the ECG generated via software and hardware. The software result was generated via Python, thus allowing the represent float numbers with 32 or 64 bits. Looking for the hardware result that was achieved by the built-in ESP32 8-bit DAC, the quantize output does not show very much precision, indicating the need for a higher resolution DAC or relying in the possibility of transmitting the data via the I2C or I2S protocol with 16-bit resolution to the DAQ. But in general, the hardware output shares reassembles with the original software output.

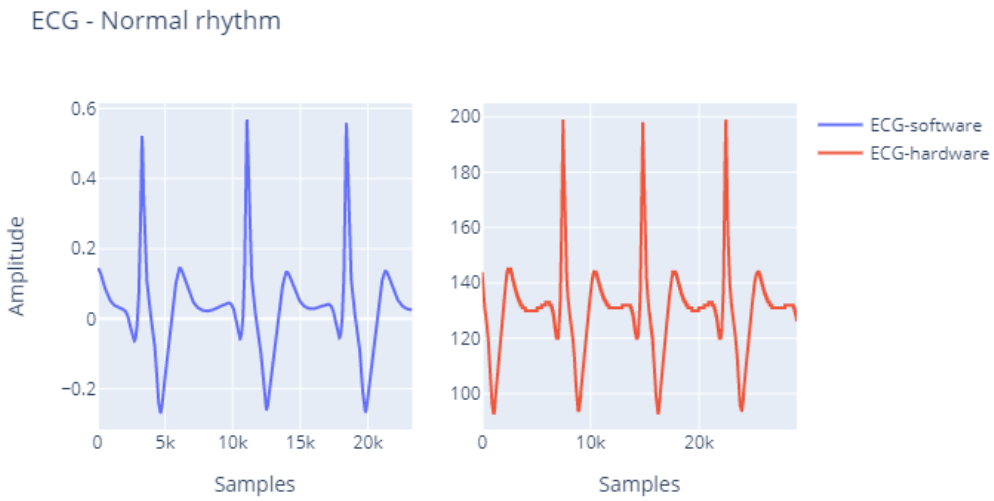

Figure 10: Comparison between ECG generated via software and hardware.

Another characteristic of the ECG simulator is the possibility to generate

multiple waveform attached to pre-defined coefficients. With those indicated by Quiroz-Juárez et al. [10], it was possibly to generate the pathologies shown in the figure 11.

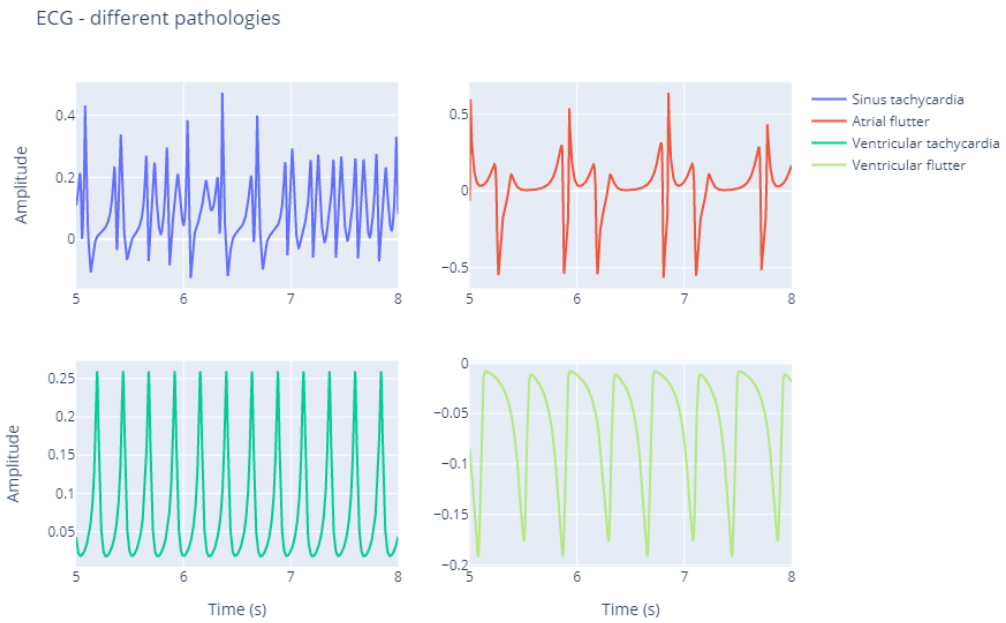

Figure 11: Pathologies simulated.

4.3 ECG signal processing

Since the hardware can only store few bytes of data, it is necessary to implement the signal processing algorithms considering the ECG signal windowed. The figure 12 show the high level graphical representation of the windowing, where the window was adjusted bigger than it will be, just for the sake of clean visualization.

4.3.1 Heart rate

The heart rate processing generated the figure 13. The QRS complex peaks that were detected are pointed clearly along the ECG signal and the heart rate evolution is between each peak is presented bellow.

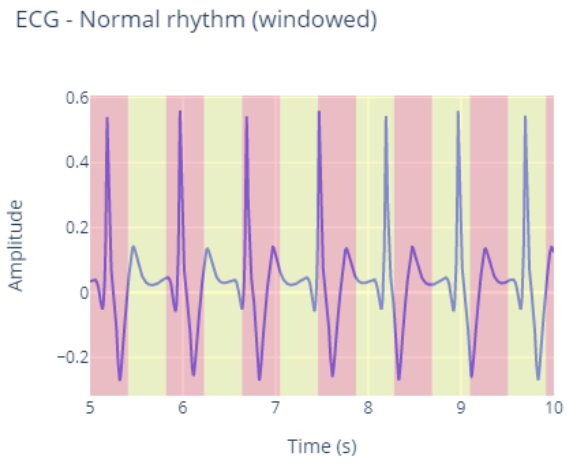

Figure 12: Windowed ECG signal.

For the signal of analysis the heart rate remains in the normal rate [15], once proving that the ECG model truly generates a good approximation for the real signal.

4.3.2 Pathology analysis

The pathology analysis is mainly based in the heart rate time-series analysis as shown in figure 14 with heart rate time series for some pathologies addressed in the simulator section. Their detection can be made by just looking to the heart rate time-series, either by tracking any outside normal rate value or checking the high variance in adjacent values of heart rate.

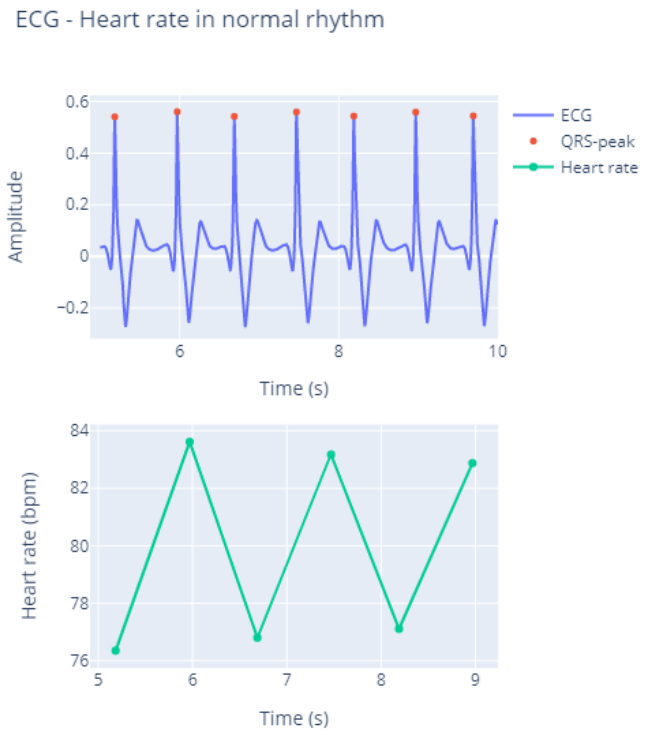

Figure 13: Heart rate extracted from ECG signal.

4.4 Data acquisition system visualization

This section address the results regarding the hardware platform of the DAQ that were obtained graphically via the ECG viewer as a Python application in a PC.

The first thing to be analyzed is the moving average filter result in the digital domain. The figure 15 shows a noise contaminated ECG signal generated by the ECG simulator and added with a true random number generator noise built in the ESP32.

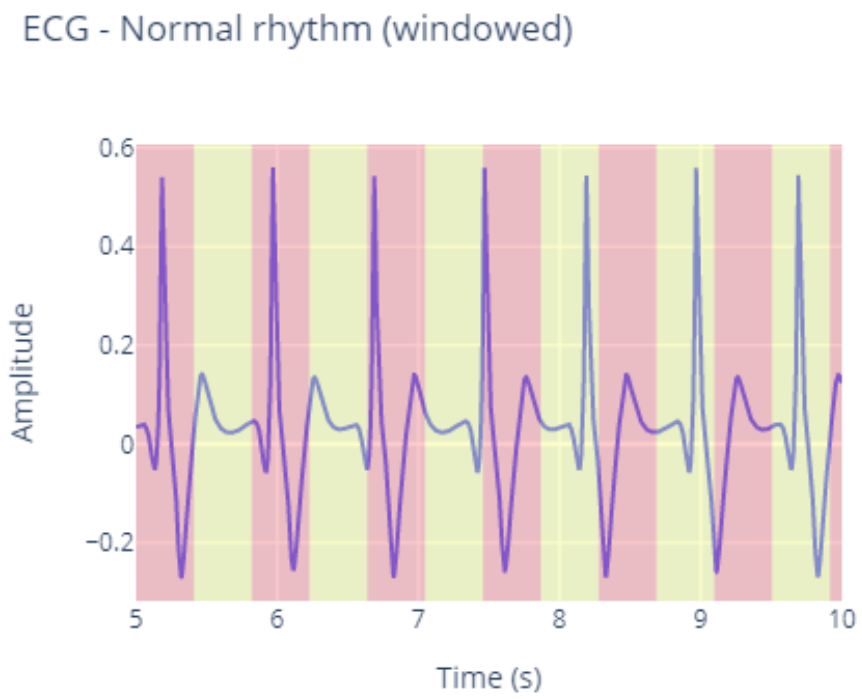

Figure 14: Heart rate extracted from ECG signals containing any pathology.

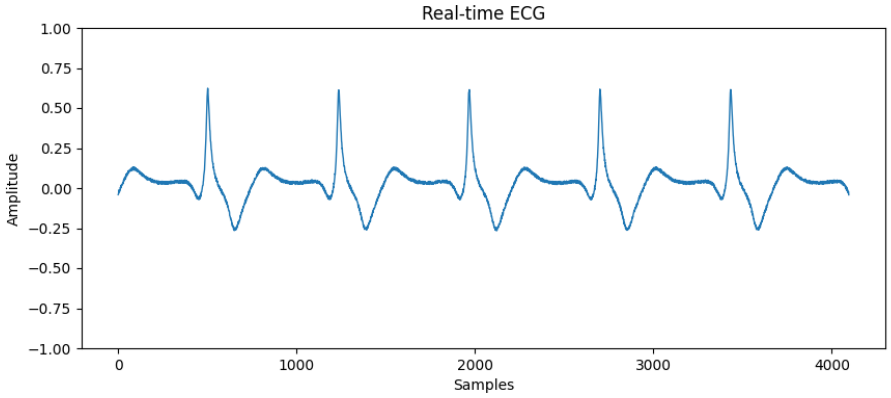

Figure 15: ECG viewer displaying a noisy ECG signal.

Right after the noisy ECG passing through the moving average filter explicated in the methodology section, the results become the one in the

figure 16. Even using a small kernel of 16 samples, this filter was sufficient to remove major noise problems.

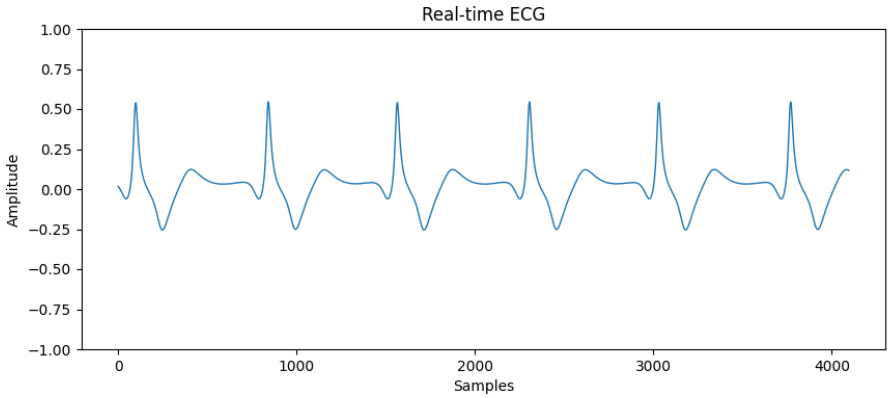

Figure 16: ECG viewer displaying a filtered ECG signal.

Finally the heart rate was also computed in hardware on top of the filtered ECG signal resulting in the figure 17 that shows a good interface to monitor the embedded DAQ results.

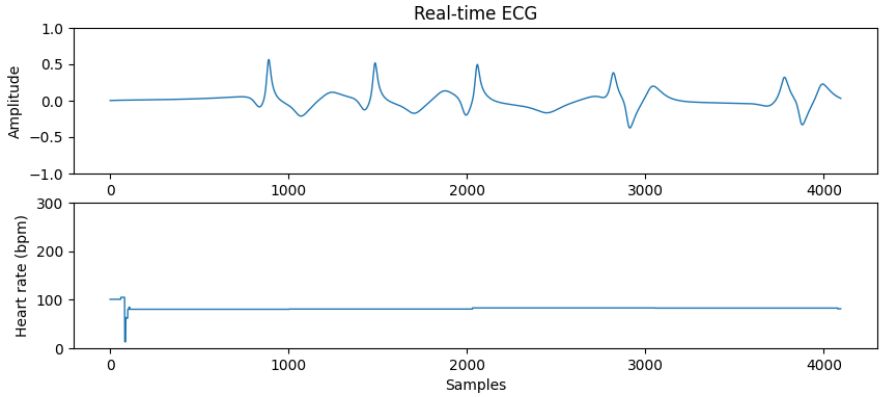

Figure 17: ECG viewer displaying embedded calculated heart rate.

5 Conclusions

The developed DAQ, besides its limitations, performed a very well job, extracting the ECG and heart rate information for pathology analysis.

The software and hardware platform allows a deeper understanding on how the system works. The high level of abstraction software has ECG simulation that helps to validate the digital signal processing algorithms before deploying in the hardware. Thus the hardware has its limitations and challenges, allowing to adapt the algorithms to a lower level of abstraction point-of-view.

In general this work indexes various detailed steps about the embedded hardware development, aiming to help researches to extract a deeper meaning of its tools.

References

[1] Raghbir Singh Khandpur. *Handbook of biomedical instrumentation*. McGraw-Hill Education, 1987.

[2] S Serge Barold. "Willem Einthoven and the birth of clinical electrocardiography a hundred years ago". In: *Cardiac electrophysiology review* 7.1 (2003), pp. 99–104.

[3] Raghbir Singh Khandpur. *Biomedical instrumentation: Technology and applications*. Vol. 1. McGraw-Hill New York, 2005.

[4] ANEEL. "Módulo 8–qualidade da energia elétrica". In: *Agência Nacional de Energia Elétrica–ANEEL* (2010).

[5] Texas Instruments. *LMH6629 Ultra-Low Noise, High-Speed Operational Amplifier with Shutdown*. Tech. rep. Texas Instruments, Tech. Rep., 2016.[Online]. Available: www. ti. com/lit/ds . . ., 2010.

[6] Yihang Zhang, Guangmin Sun, and Yuchen Yang. "12-lead ecg data acquisition system based on ads1298". In: *Procedia engineering* 29 (2012), pp. 2103–2108.

[7] M Murugappan et al. “Development of cost effective ECG data acquisition system for clinical applications using LabVIEW”. In: *2014 IEEE 10th International Colloquium on Signal Processing and its Applications*. IEEE. 2014, pp. 100–105.

[8] Vladimir Valerievich Stuchilin, Valentina Anatolievna Rumyantseva, and Iliya Sergeevich Svirin. “The Use of Digital Signal Processing Algorithms for Electrophysiological Diagnostics of Cardiovascular Diseases”. In: *Biomedical and Pharmacology Journal* 10.1 (2017), pp. 119–128.

[9] Raghbir Singh Khandpur. *Compendium of Biomedical Instrumentation*. John Wiley & Sons, 2019.

[10] MA Quiroz-Juárez et al. “Generation of ECG signals from a reaction-diffusion model spatially discretized”. In: *Scientific reports* 9.1 (2019), pp. 1–10.

[11] Mesut Sahin, Howard Fidel, and Raquel Perez-Castillejos. *Instrumentation Handbook for Biomedical Engineers*. CRC Press, 2020.

[12] A Sedra et al. “Microelectronic Circuits 8th Edition”. In: *Chapter* 14 (2020), pp. 1235–1236.

[13] Rahul Kher and Shivang Gohel. “Abnormality classification of ECG signal using DSP processor”. In: ().

[14] Lumen Learning OpenStax. *Anatomy and Physiology II*. URL: `https://courses.lumenlearning.com/ap2/chapter/cardiac-cycle/`.

[15] *Pulse and Heart Rate*. URL: `https://my.clevelandclinic.org/health/diagnostics/17402-pulse--heart-rate`.

A Schematics

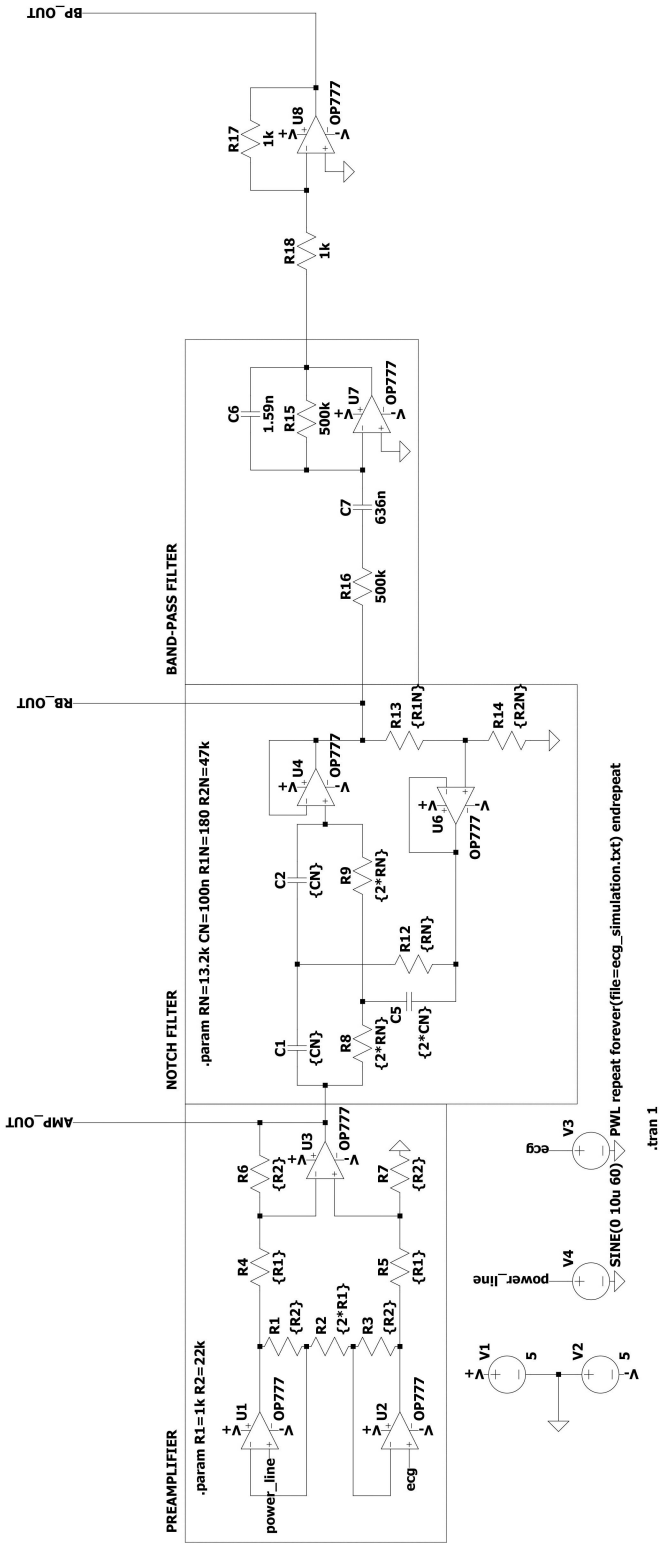

Figure 18: Analog front-end LTSpice schematic.