PROJECT REPORT

WEARABLE DEVICE FOR ARRHYTHMIA DETECTION

December 20, 2018

Thiago Henrique Pincinato

University of Bern

Biomedical Engineering

Contents

List of Figures

1. ABSTRACT

1.1. Summary

The aim of the project is to detect arrhythmia events by measuring the heart rate and by detecting the type of the user's activity. Moreover, the device need to be configurable for different users, which may have special heart rate values. This configuration must be done by a friendly software, in which the user can easily make his/her own settings.

1.2. Medical Background

Arrhythmia is a term used to describe irregularities of the electrical impulses responsible by the hear beat. Broadly speaking, this disorder of the electrical impulses may cause irregular heart rates, which leads to an inefficient blood distribution, causing permanent damage in organs that demand high oxygenation. [1]

Due to the serious problem causes by Arrhythmia, a wearable device able to do a prediagnostic would be extremely useful and relevant for the population health.

1.2. Functionality of the application

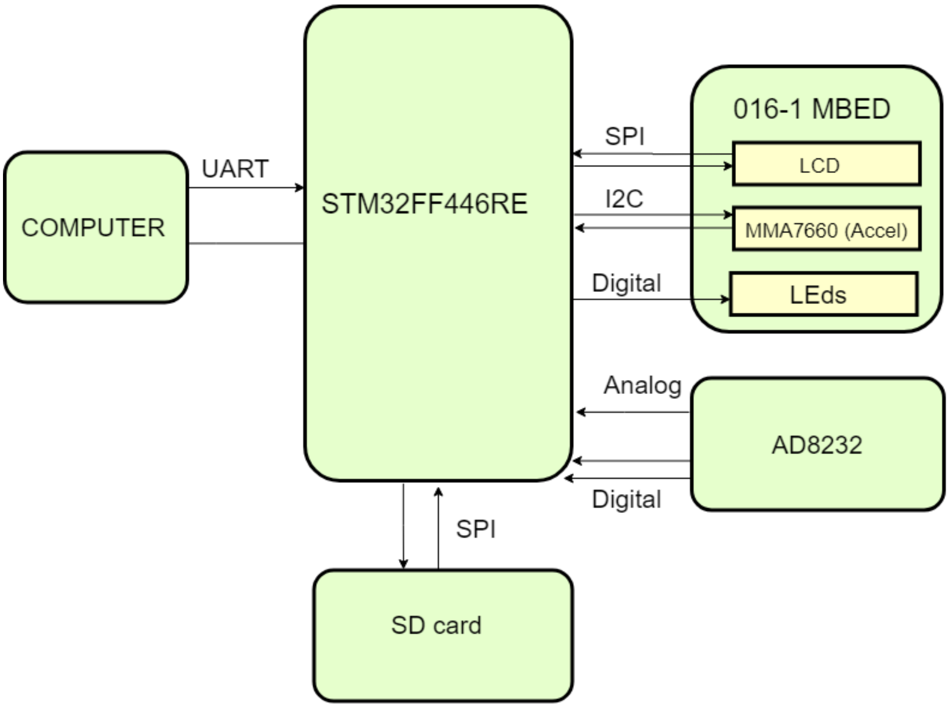
The user's heart rate is measured during daily life activities, including sport, sleep, work and so on. The values of heart rate are ranked as low, medium or high, based on a configuration that can be done by the user. Meanwhile, an another sensor is used to classified the type of the user's activity, light, normal or heavy .

Finally, a prediagnostic of a possible arrhythmia could be done by comparaing those classifications.

1.3. Hardware required

A schematic of the hardware can be seen in the figure 1.

Figure 1: Schematic of the Project



1.1.1. Heart Rate Monitor

In order to detect the Heart Rate an ECG signal is used. By analyzing the analogical ECG signal, the peaks are detected, and then, the Heart Rate is computed. The Board chosen to capture the ECG signal was the AD8232 , combined with the Cable Sensors and the biomedical sensor pads. All parts can be seen in the figure 2.

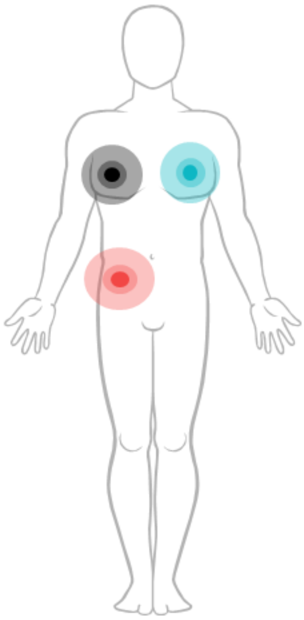
Figure 2: Heart Rate Monitor

The AD8232 need to be supplied with 3.3v and it does not requires any kind of configuration. It has three outputs, LO- and LO+ (logic outputs) that indicate if the cable sensors are connected or not; and the output (analog signal) that represents ECG signal in a range from 0v to 3.3v.

The Sensor also contains a digital input SND' that can be used to select a low power operation. In addition to that, it is possible to measure the signal in the cables by using the pin RA, LA and RL.

The placement of the cables must be done as shown in the figure 3.

Figure 3: Electrodes' placement



More information can be found in the datasheet [2].

1.1.2. Accelerometer, LCD, LEDs and joystick

In order to determine the user's activity, the accelerometer MMA7660 was used. This accelerometer is placed in the mbed 016-1 board, in which the LCD LEDs and joystick are also used.

The communication with the accelerometer is done via I2C, and the communication of the LCD is done via SPI.

All the information about the MMA7760 can be found in the datasheet [3].

1.1.3. SD card and adapter

As the device needs to be adaptable for the user, a SD card was integrated into the system. The SD card save the setting done by the user, in a non-volatile memory, meaning that the configuration are save and can be load even after power supply is removed from the device.

The integration of the SD card is done via an adapter that is illustrated in the figure 4.

Figure 4: SD card adapter

The adapter contains seven connection:

- A VCC and a GND used for power supply.
- CD (digital) output, that indicates the presence of a SD card.
- DO (digital) output, data output or MOSI for SPI communication.
- SCK (digital) input, clock used in the communication.
- DI (digital) input, data input or MISO for SPI communication
- CS (digital) input, select pin for SPI communication.

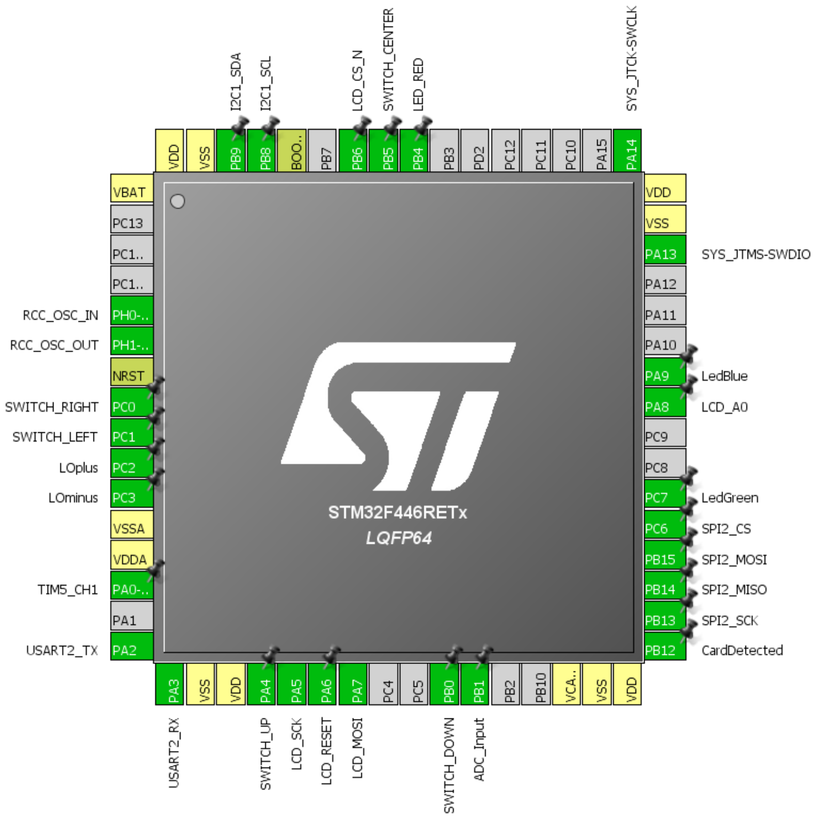
1.1.4. NUCLEO STM32F446

The main hardware of the project is the Nucleo STM32F446, which control all other board and sensors. In order to do that, several peripherals are configured as illustrated in the figure 5.

Briefly, the device utilizes 2 SPI, 1 I2C, 1 UART, 8 digital inputs, 6 digital output, 2 Timers, 1 ADC and 1 RCC Oscilator.

*One of the timer is named as LED_RED, that is why just one timer can be seen in the figure 5.

Figure 5: Pinout



1.1.5. Bill of Materials

All external hardware were bought on the sparkfun website, the table 6 shows the respective price of those boards.

Figure 6: Bill of Materials

Product	Link	Price
SparkFun Single Lead Heart Rate Monitor - AD8232	https://www.sparkfun.com/products/12650	$ 19.95
Sensor Cable - Electrode Pads (3 connector)	https://www.sparkfun.com/products/12650	$ 4.95
Biomedical Sensor Pad (10 pack)	https://www.sparkfun.com/products/12969	$7.95
SparkFun Level Shifting microSD Breakout	https://www.sparkfun.com/products/13743	$5.50

2. Functionality realized

The follow subsection describe the functionality that were realized so far.

2.1. Heart Rate Detection

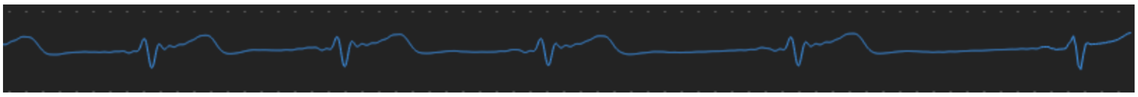
The analogical signal provided by the AD8232 sensor is sampled with a frequency of approximately 200 Hz. The frequency is controlled by a timer interrupt that activates the interrupt of the ADC.

The sample frequency was chosen based on the reference [4], which demonstrates that the most relevant information of the ECG are placed between 0.05 Hz and 100 Hz. By respecting the Nyquist criterion, the sample frequency should be two times higher than the higher frequency of the signal, and thus, the 200 Hz was used.

In reality the 200 Hz is not achieve due to the time need to the ADC to process an operation. However the results are pretty close to 200 Hz and can be approximated without issues.

Once the signal is sampled, series of signal processing take place in order to allow the computation of the Heart Rate. Those signal processing are need due to the fact that the signal sampled may contain huge amounts of noise. To better understand the behavior of the signal, we can look at the figure 7.

Figure 7: ECG signal without motion artifacts

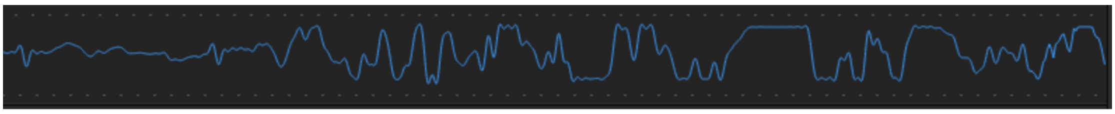


In the figure 7 we can easily recognize a ECG signal, and thus, the heart rate could be done by just computing the amount of the QRS complex in a certain unit of time.

Despite it seems simple, in a real usage of the device the user realize different activities, such as, for instance, sports, running, biking and so on. In those situations muscles present in the chest are in constant work, interfering in the ECG signal.

Such an interference is so-called Motion Artifacts, and its impact is elucidated in the figure 8.

Figure 8: ECG signal with motion artifacts

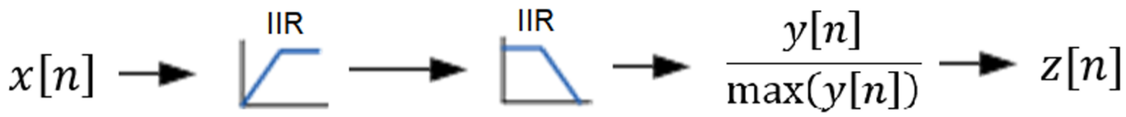


To minimize that effect, and therefore, be able to detect the heart rate the input signal is filtered. An IIR low pass filter (biquad filter) in serie with an IIR high pass filter (also biquad filter) are used. The aim of those filter are to remove the frequencies lower than 4 Hz and higher than 23 Hz,limiting the ECG signal to narrow band , in which ninety of the ECG spectral energy is concentrated.[5]

After both filter, a removing of the unstable values due to the IIR filters takes place. Only then, the signal is normalized, and an adaptive threshold is applied to detect peaks. Those peaks are then, translated to Heart Rate.

The figure 9 briefly explains the procedure done in the STM32F446RE.

Figure 9: ECG signal processing



Once the Hear Rate are detected, an average filter is applied, removing outliers in the final results.

2.2. Classification of User's Activity

The digital signal provide by the Accelerometer (via I2C) is sampled with a frequency of 100 Hz. This frequency is controlled by a timer interrupt.

By using the values from the accelerometer, the distance is computed via double integration of the acceleration in respect to time. Such integration in the time domain are transformed in a sum in the discrete domain. The reference [6] explains how this integration must be done to reduce the noise due to the integration process. Nevertheless, the integration always results in an increase of the noise effects [6].

The signal processing applied to the accelerometer values can be seen in the figure 10.

Figure 10: Accelerometer signal processing

$$x[n] \rightarrow \frac{1}{n}\sum_{i=0}^{n-1} x[i] \rightarrow -calibration \rightarrow \int dt \rightarrow \int dt \rightarrow y[n]$$

The first measure of the accelerometer is used as calibration, which is subtracted from each measure. That procedure reduce some part of the static noise.

Taking in account that the majority of the movement of a person is in the Y axis of the accelerometer, only this axis was analyzed and used in the project. As consequence, the device must be placed in a such way that it does not rotate.

Once the distance is computed, its value is classified as light, normal or heavy. In other words, the dislocation of the user determines its activity.

2.3. Communication

2.3.1 USART via DMA

In order to communicate with a personal computer, an USART was configurated via DMA. The Basic parameters of the USART are:

- Baud rate: 115200 Bit/s
- Word length: 8 Bits
- Parity: none
- Stop bits: 1

The USART_TX is configured as normal DMA mode, while the USART_RX is configured as circular DMA mode. In both cases the data width is a Byte.

The USART via DMA works correctly only when the global interruption is enabled; otherwise a FLAG is never update and the USART stays in a busy state forever.

The communication with the computer is done by following a home made protocol that will be explained in the section 2.5.

2.3.2 Protocol

Briefly, a protocol was made to establish a communication between the computer and the device. In this protocol all command start with the symbol ? and end with the symbol !. The command are defined in the figure 11.

Figure 11: Protocol of Communication

Command	meaning	device action
Act	Action Configuration	send actual configuration of the device
Hig	set high heart rate value	update high hear rate value, using new recieved value
Nor	set normal heart rate value	update normal hear rate value, using new recieved value
Low	set low heart rate value	update low hear rate value, using new recieved value

It is important to notice that the protocol is just one direction, from computer to device. In the other path, device to computer, there is no protocol to be respected, given that the software just transmit the message from the device directly to the user.

Another important point is that all message must have the size of 8 bytes. When the command plus the data are smaller than that, the message must be fulfilled with the symbol !.

An example of a message from the computer to the device is ?Hig120! . In this case the device will update the value of high heart rate to 120, and it will send an acknowledge message.

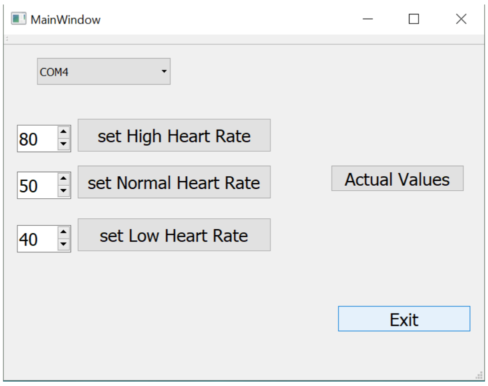
2.3.3 Software

A friendly software was developed for windows. This software allows the user to configure the device and also to check its actual configuration. In addition, the software find the COM in which the device is connected, and thus, the user does not need to worry about that.

Those features are important to encapsulate the protocol and make the experience of the user as pleasure as possible.

The figure 12 illustrated the software window.

Figure 12: Software window



As we can see in the figure 12 the user has 6 options. He/she can set the value of each heart rate classification; he/she can read the actual values of the device; the COM is automatically selected, however it can be changed; and finally, he/she can exit the application.

All options give a feedback to the user, that is, in fact, the message sent by the microcontroller or a timeout.

2.4. Storage of Information

In order to storage all setting in a non volatile memory, a SD card was used.

The SD card is accessed via SPI (mode 0), and its main parameters are:

- Prescaler: 256 (Baud rate : 175,78 KBits/s)
- Clock polarity: low

- Clock phase: 1 Edge
- CRC calculation: none
- NSS Signal type: Software

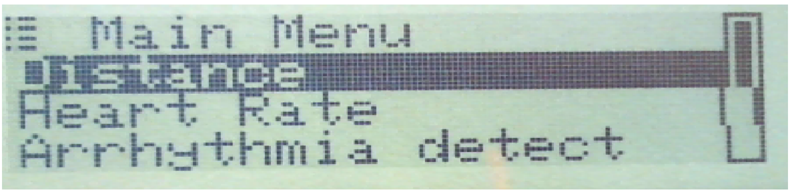
As no file system was used, the utilization of the SD card is done by directly sending commands and data. The reference [7] explicates about the protocol of communication need to communicates with the SD card.

2.5. Menu

The menu driver was used to build a menu in which the user can choose one of 4 options.

The figure 13 shows the LCD Menu. The option USB has omitted in such a figure.

Figure 13: LCD Menu



The aim of the Menu was to improve the usability of the device and increases its value by adding additional features.

3. Overview of the application's structure and control flow

3.1. Application's structure

The application starts by initializing the peripherals, my interfaces(SD, table, Analysis, ECG, Accel, ADC, timer), the LCD, the joystick, and finally the menu. During this process of initialization, the red , blue and green leds blink at the same time.

Once the application starts, the main menu shows up and one of the 4 option can be chosen by using the joysticks.

In the option Distance, the user can see the relative distance that he/she is dislocating in g units. When entering this option, the pincinato's ACCEL interface is started. The data is updated via this interface, and the new values of distance are added to the previous one, making an accumulation of the relative distances. This option could be used for measure how much a person has run or walked during a certain activity or time.

In the option Heart Rate, as the name suggest, the Heart Rate is shown. When this option is chosen, the pincinato's ECG interface is started. The date is update via this interface, and the new value is displayed in the LCD.

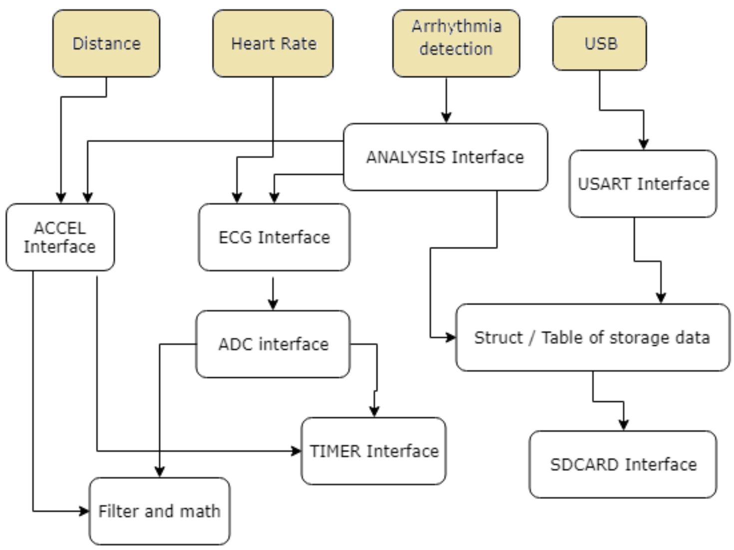
The option Arrhythmia detect shows the classification of the heart rate and the classification of the activity. When this option is chosen, the pincinato's ANALYSIS interface is started. The data is update via the interface, and the new values are displayed in the LCD. In this option, a Green Led is activated. In case a possible arrhythmia is detected the Green Led in turned off and a Blue Led is turned on.

The USB option, not illustrated in the figure 13, is responsible to enable the communication of the device with the computer. The communication works if, and only if, the user is in this option. When this option is chosen, the pincinato's USART interface is started. All the usart procedures are handled by this interface.

When leaving one of the option, the respective interface is stopped.

The schematic 14 elucidate which menu option is connected with which pincinato's interface as well as the connection between interfaces.

Figure 14: Menu and Interfaces connections

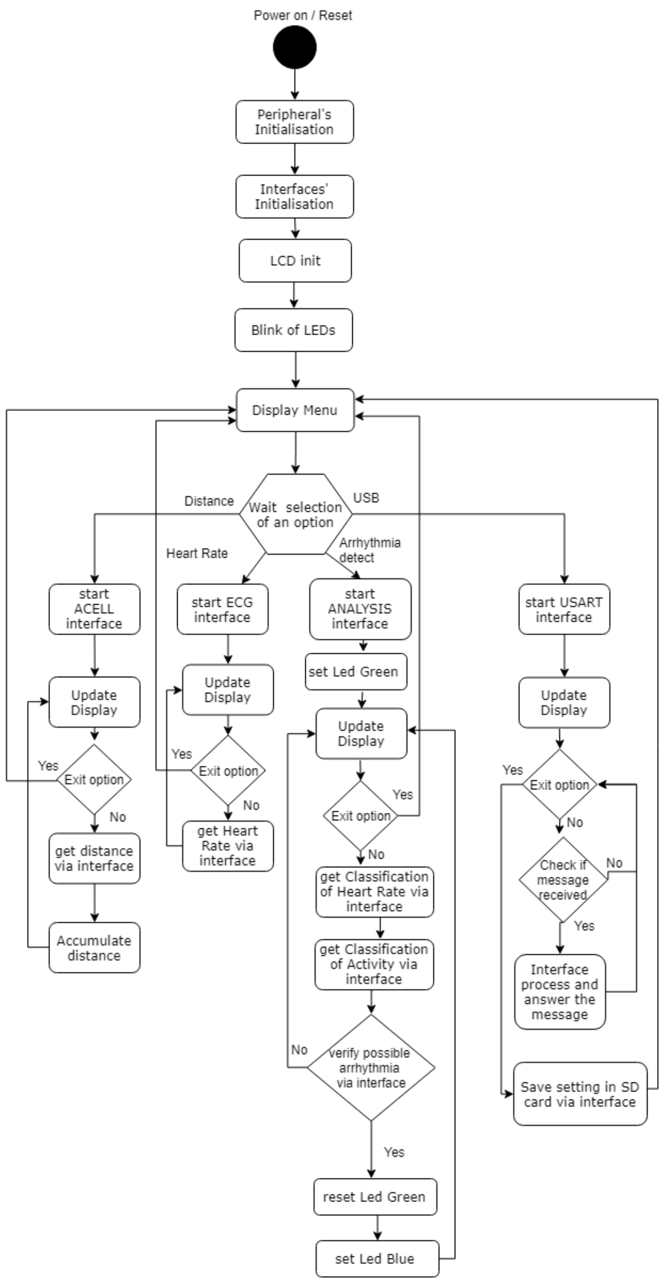


As we can seen, the program was made in such a way that the main functions, signaling processing, interruptions and communication are encapsulated in several interfaces. This approach is based on the object oriented paradigm , and thus, it has several advantages regarding the maintenance and usability of the code.

3.2. Control flow

The control flow diagram (figure 15) shows the flow of the application works, some procedure are omitted for sake of simplicity.

Figure 15: Control Flow



4. Lessons learnt

During the project some challenges have appeared, and each challenge gave me the opportunity to learn something.

First, I could realize how complex can be to process a signal in the present of noise and interference. We are so used to compute filter in MatLab or other powerful tools that we loose the idea of how complex it is to do such operation in a limited hardware. Even well known signal, as ECG signal, can become very complex when a motion artifacts are present in the signal. Due to that, it is very important to use good sensors, and not the cheapest one, for medical devices. A good sensor can provide a more reliable data. Another point is that, ADC signal are very susceptible to noise in the path of transmission. Therefore, it is cleverer to use sensor with digital signal (I2C, SPI, CAN and so on) than analog signal.

Second, Accelerometer are very cheap and practical, however, for the majority of the cases it needs to be combine with gyroscopes. Only using an accelerometer the application is very limited. In addition, due to noise, the values of accelerometer required a signal processing.

Third, I spent a lot of time to realize that the UART via DMA does not work without the general interruption enabled. In this situation I leaned that when something is not working as expected,I should take a glance at the STM documentation first and only after search on the internet.

Finally, regarding the SD card, it is very important to be able to change the SCK during application. The initialization of the SD card must be done with a SCK between 100 KHz and 400 Khz. If such frequency is maintained after initialization, the communication would be slower than we are used to.

5. Bugs and limitations

The application works with a very low number of bugs. So far, all bugs are related with the LCD display that sometimes crashes and stops to display the correct information. Besides that, no bug were found, and the application runs in a predictable way.

Concerning the limitations, the application is not able to detect the heart rate in a precise way, the value keep floating in a range of +-15 bpm or in certain situations +-30 bpm. To prove the concept and the idea of the device those values are not a limitation. However , such measurements could never be used to medical purpose. This limitation happens due to the processing of the analog signal, which contains motion artifacts.

6. Future extensions

Despite all development done , in the future it would be necessary enhance the heart rate detection, either by using a better sensor or by improving the signal processing.

It would also be necessary to change the driver of the LCD display or to improve it in order to avoid bugs in the display.

7. List of hardware and software used

The figure 16 lists all hardware and software used as well as their version.

Figure 16: List of Hardware and Software used

Hardware	Hardware/Software	Version
SparkFun Single Lead Heart Rate Monitor	Hardware	------
Sensor Cable - Electrode Pads	Hardware	------
Biomedical Sensor Pad	Hardware	------
Logic Analyser	Hardware	Saleae
STM32F446RE	Hardware	F446RE
MBED 016-1	Hardware	016-1
KEIL uVision	Software	5
STMCube	Software	MX4.26.1
QT creator	Software	4.7.0

References

[1] American Heart Association. *Definition of Arrhytmia* [online] Available at: <https//www.heart.org/en/health-topics/arrhythmia/about-arrhythmia> [Accessed 8 October 2018].

[2] Analog Services .*AD8232 datasheet* [online] Available at: <https://cdn.sparkfun.com/datasheets/Sensors/Biometric/AD8232.pdf> [Accessed 29 October 2018].

[3] NXP .*MMA7660 datasheet* [online] Available at: <https://www.nxp.com/docs/en/data-sheet/MMA7660FC.pdf> [Accessed 29 October 2018].

[4] Dr. Dipali Bansal et al, 2012. FILTERING TECHNIQUES FOR ECG SIGNAL PROCESSING. *ISSN: 2249-3905*

[5] Mrs. Sulata Bhandari and Ankit Gupta, 2015. ECG Noise Reduction By Different Filters – A Comparative Analysis. *International Journal of Research in Computer and Communication Technology, ISSN: 2278-5841*

[6] NXP, 2007. Implementing Positioning Algorithms Using Accelerometers. *Freescale Semiconductor, Application note*

[7] Chain. *How to Use MMC/SDC* [online] Available at: <http://elm-chan.org/docs/mmc/mmc_e.html> [Accessed 2 December 2018].