RPScope - An electronics lab based on Raspberry Pi

Pedro Silva Pereira
pedro.s.pereira@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

July 2019

Abstract

With the fast growing of electronics in this recent years a new maker community has emerged that makes electronics its hobby. In these days it becomes essential these two devices: the oscilloscope and the signal generator. In this way it is quite useful if there was a single device capable of acquiring and generating analogue and digital signals and for sake of simplicity, it would had a screen that would allow the user to interact with the device. At the moment, there are already some solutions that do not completely solve this problem. In this project I will develop a device that allows observing 2 simultaneous analog and 4 digital channels, generation of sinusoidal, triangular and impulse signals. For this application is still no device on the market with the same features with a built-in screen. The coordinator of this system will be a Raspberry Pi that will control both the acquisition blocks and the generation of signals. At the end of this dissertation it will be possible to understand the whole design process that is behind this device.

Keywords: acquisition system, signal generation, Raspberry Pi

1. Introduction

1.1. Motivation and objectives

The main motivation of this work is the need to create a system of acquisition and generation of signals.

Develop a prototype in which the user can analyse and generate signals on a single device, independent of a computer and at the same time inexpensive. With a plug and play system, it is possible to have a set of tools to work in electronics in which only a connection to the electrical power is necessary.

In this way the makers who want to make their own project have thus a tool of easy use.

Considering the non-professional use, the basic functions that the system must guarantee are: power supply DC 3.3 V, ± 5 V; function generator, using a DDS for the generation of sinusoidal, triangular and square signals; 2-channel oscilloscope capable of acquiring 1 Msps with a ± 20 V range; Digital Analyzer (4 channels); and 2 outputs Pulse Wave Modulation (PWM). Another 3 requirement to take into account is the need to be cheap, robust and reliable, as this would be a great advantage in relation to competition.

1.2. State of the art

I will describe some systems that are available on the market. The nScope [5] is sold as a portable electronic laboratory, to attach in a breadboard. It allows to acquire 4 analogue channels simultaneously at 4 MSps, function generator with 2 analogical channels and 2 digital channels, also allows to supply the power lines of the breadboard with ± 5 V. A Universal Serial Bus (USB) with the aid of a program developed by the brand.

BitScope [1] is defined as a complete USB oscilloscope with oscilloscope capabilities, logical analyser, arbitrary wave generator and even a spectrum analyser.

Another type of products with the same intention already exists for sale. In kits do it yourself (DIY). Boards can be sold for soldering in several online shops. There is a system that does oscilloscope [6] and another, independent, that makes generator of functions [7]. Both systems have integrated screens, in this way there is no need for computers unlike the previously described systems.

In this project an intuitive graphical user interface (GUI) is required so that the user can see the signals and control the acquisition time. In this section, existing software components will be presented. In the case of BitScope and nScope, there are specific softwares for these equipments, however it is also important to explore open-source solutions since the realisation of this project presupposed, from the start, the adaptation of an existing interface. The characteristics necessary for a good GUI are:

- Control horizontal scale V / div;
- Control vertical scale sec / div;
- Control the trigger level;
- Start / stop button acquisition;

- Select button single shot acquisition or continuous acquisition.

The following options are required for the control of the signal generator:

- Control signal amplitude;
- Control signal frequency;
- Control the waveform type.

With the ScopeFun it is possible to adjust the amplitude, frequency of acquisition, to control the level of the trigger. In the acquisition mode it is possible to choose between continuous and one-shot. This software runs on linux but could not be run on raspberry due to the lack of libraries compatible with the raspberry processor.

OpenHantek complies with the requirements, allowing it to control the sampling rate, amplitude and signal coupling, trigger level and also contemplates the MATH mode in which it is possible to perform simple mathematical operations with both input signals. However, this software can not be used since its source code is too complex to be adapted for this work.

Xoscope is a simple software with a source code that is easy to adapt, yet has a graphic interface that is too weak and the fact that the interface has to be controlled by the keyboard which is very unintuitive.

Lastly, there is this software that meets most requirements, has a better GUI than the Xoscope and the source code it is also simple to adapt to this work. It is possible to change the signal amplitude, coupling mode, sampling rate and control the trigger.

2. System overview

In the following block diagram 1, it is possible to see how the various components of the system interconnect. The signal to be analyzed is connected to the signal conditioning block, the dsPIC ADCs will digitalise the conditioned signal. Then the dsPIC will send the data to Raspberry by SPI.

Digital channels, both input and output, connect to the GPIO of the Raspberry.

The signal generation circuit is constituted by the same dsPIC and another integrated circuit with the DDS function.

Raspberry is responsible for displaying the graphs of waveforms and allows the user to program the function generator. The screen is connected by a cable to Raspberry. 5' display is powered by the Raspberry.

Lastly there are a circuit that converts AC current to DC 3.3 V, ± 5 V to power the entire system as well as other peripherals that the user intends to power. The system only requires an external power supply of at least 5 V to power the raspberry and the existing voltage outputs.

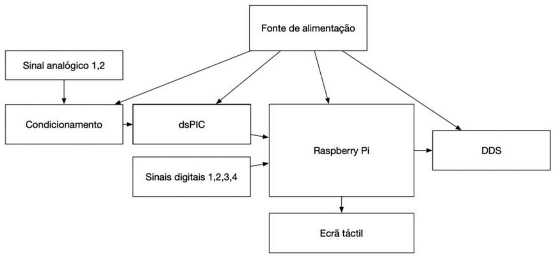


Figure 1: Blocks diagram of RPScope

3. Implementation

The final aim is to have all the systems described in the last chapter operating simultaneously on a single board which is coupled to the Raspberry, in other words, a shield. However, considering the complexity of the project, it was decided to split the system into simpler sub-systems: one board for the power supply, another for the signal conditioning circuit and another for the signal processing circuit. With the various isolated circuits, it is easier to test each one and solve possible problems that may arise. The program used for the design of all the electrical schemes and their PCBs was Altium.

3.1. Power supply

In order to be able to size the power supply circuits, it is necessary to make a preliminary study of the currents and voltages necessary to the operation of the whole circuit.

From 5V there are Raspberry included LCD - 2000 mA and instrumentation amplifier - 4.1 mA

From 3.3V there are dsPIC - 100 mA and the DDS - 8 mA.

These components makes a total of 2008 mA at 5V and 108mA at 3.3V In order to make the circuit as simple possible with a minimum of components, a Step-Down LM2592HV from Texas Instruments was selected that allows to generate simultaneously 5V and 3.3V output 2A which is sufficient to power the entire system.

To generate the -5V voltage required for instrument amplifiers, another Texas Instruments Step-Down Voltage Regulator LM1575 is also used.

The circuit was tested but the results were not consistent. The voltage values measured sometimes were those that were intended and other times were zero. Which made the board unusable. This was due to a soldering iron, especially in integrated circuits and very low clearance.

3.2. Signal conditioning

The conditioning system can be subdivided in two: input protection and signal amplification.

The first part is made of a resistive divider 1 to 5 and then a series of diodes that are connected to the supply voltages which causes the signal entering the instrumentation amplifiers to be between $\pm 5V$

The second part aims to amplify the amplitude of the signal so that the entire dynamic range of the ADC is fulfilled.

It is decided to use instrumentation amplifiers since they have digitally programmable gain, have a good common mode rejection (CMRR), which allows to amplify only the difference between the signals in their inputs, and make differential measurement.

For this, the Analog Devices AD8250 instrumentation amplifier is chosen. This IA has 4 digitally programmable gains (1, 2, 5 and 10), a maximum bandwidth of 10 MHz. The serial connection of these two AIs, that is, the output of the first IA is directly connected to the input of the second , allows several combinations of gains1 that allow the choice of different ranges.

G1	G2	Combined Gain
1	1	1
1	2	2
1	5	5
1	10	10
2	2	4
2	5	10
2	10	20
5	5	25
5	10	50
10	10	100

Table 1: Combined Gain

As it is intended to measure signals whose amplitude is negative and the dsPIC ADC is not bipolar it will be necessary to add an offset voltage.

Taking advantage of the IA reference terminal, which adds a voltage to its output, a voltage of 2.048 V is added, thus passing the signal to the ADC input between 48 mV and 4.048 mV.

The reference voltage is generated by the LM4040 component of Texas Instruments.

The programming of the gains of these amplifiers is made through three terminals in parallel. Two of the terminals are used to program the gain (A0 and A1) and the third (WREN) active on the descending slopes will change the gain for the current combination of AO and A1. For example, if the IA is programmed to gain 1 and there is a change in values at terminals A0 or A1, this change will only take effect when there is a falling edge of the signal present at the WREN terminal. The next table 2 shows how the various gains can be obtained.

Given this mode of operation it is possible to share the terminals A0 and A1 of the two IAs, however it is only possible to change the gains of one IA at a time.

A0	A1	Gain
0	0	1
0	1	2
1	0	5
1	1	10

Table 2: Ganhos depending A0 e A1

The purpose of this board was to be a prototype to later develop a final ADC-adjusted version of the dsPIC. The offset that is placed in the second instrumentation amplifier should be close to 1.65V and not 2.048V. This is the only way to properly split the scale taking into account the tensions allowed by the dsPIC ADC.

On this board it is possible that the output signal has negative voltages, which makes no sense and can be harmful to the ADC. One way to correct this problem was to place diodes that only conduct if the signal is between 0-3.3V.

Finally, the first IA was not functional, at this stage it is only possible to change the gains in one IA and not in two as supposed.

3.3. Signal Acquisition

In this project will be used a dsPIC33, more precisely DSPIC33EP64GS506. This micro-controller has all the necessary features: 2 ADC of 10 bits with 3.25 MSps, has 64 pins, 64 Kbytes flash memory, 8 Kbytes RAM, 8 channels for DMA, 9 16-bit timers, at the communications level, has 2 UART, 2 SPI^{TM} and 2 IC^{TM}.

The dsPIC oscillator [3] of the dsPIC has a frequency of 7.32MHz, which is possible to configure up to 83.25 MHz with a series of multipliers and dividers.

The ADC [2] already included in dsPIC allows a conversion rate of up to 3.25 Msps per channel with a maximum resolution of 12 bits. Given this sampling frequency it is possible to analyse signals up to 1.5MHz, respecting the sampling theorem.

It is also possible to set the desired resolution, in this case they are 8 bits because it is the vertical scale resolution of the GUI. The reference voltage of the ADC is equal to the supply voltage of the dsPIC, ie 3.3V.

Taking into account the following constraints: clock frequency of 83.25 MHz, minimum rollover of the timer2 is 64, the maximum sampling frequency is 650kS / s.

In order to establish communication between dsPIC and raspberry the interface SPI [4] will be used, and raspberry will control this communication.

- SDI: data entry
- SDO: data output
- SCK: shift clock input or output

- SS: Slave Select active at LOW.

The total memory available for data is 8Kbytes which means that it is possible to store a vector of up to 8k samples taking into account that each sample is 8 bits. Note that this prototype circuit only allows you to analyse a channel. In the final project it is intended that it is possible to analyse 2 channels simultaneously.

For the dsPIC programming, the MPLAB IDE and a pickit3 were used.

The function of dsPIC is to acquire data from the ADC and save it temporarily and then send it to Raspberry. The dsPIC also has to control how often the ADC values are read and control the IA gains.

Since it is intended that the acquisition of the signal must be made at the highest possible frequency an interrupt routine was used, this routine reads the value of the ADC and places into a buffer. This routine is executed whenever timer2 runs out. The frequency which the timer2 expires is what causes it to acquire faster or slower. This value is controlled by a command that Rasberry sends to dsPIC.

Whenever the user wants to increase or decrease the IA gain raspberry will send a command to the dsPIC so that it places the ports that bind the IA to HIGH and LOW to program the IAs with the desired gain value.

3.4. Raspberry Pi

This project assumes the use of Raspberry Pi, a computer with the size of a credit card in which the added value is to be able to be connected to a touch screen and thus the user can control the whole system from this.

It also has a series of GPIO that allow to communicate with other peripherals either by I2C, SPI or UART. The latest version, Raspberry Pi 3, has an ARM processor of 1.2 GHz, Wi-Fi, Bluetooth, 1GB RAM.

SPI communication through the *spidev* library allows frequencies up to 20 MHz

3.5. Graphical User interface

The program chosen for this project was DSoda. This software was meant to communicate with an oscilloscope that was connected by usb interface.

In order to be able to receive dsPIC data the USB interface has been replaced by SPI. All functions that sent and received data were replaced by code that allows sending and receiving data by SPI. In the program window it can be displayed 10000 samples with an 8-bit resolution. Given that the dsPIC can only store 8000 samples, it is chosen to acquire 5000 samples, which in order to fill the whole screen these will be duplicated.

In the control panel, you can change IA gains as well as sampling frequency and trigger level.

The operation of the program will then be explained as long as the user presses the *Start* button until the waveform appears on the screen.

When you press the start button, raspberry will send a command to the dsPIC, in this case it sends the character 'a' and it will begin to acquire the data. When this acquisition finishes dsPIC places a port to HIGH and starts to send the values read by the ADC to raspberry, after transferring the last value the port is placed to LOW and so raspberry knows that the transfer of the data is finished.

Then raspberry will draw the waveform of the signal in the preview window. The time that passes between the last data received by raspberry and the appearance of the signal in the window is relatively large due to the function that already existed in the original DSoda program. The next flowchart 2 explains this process.

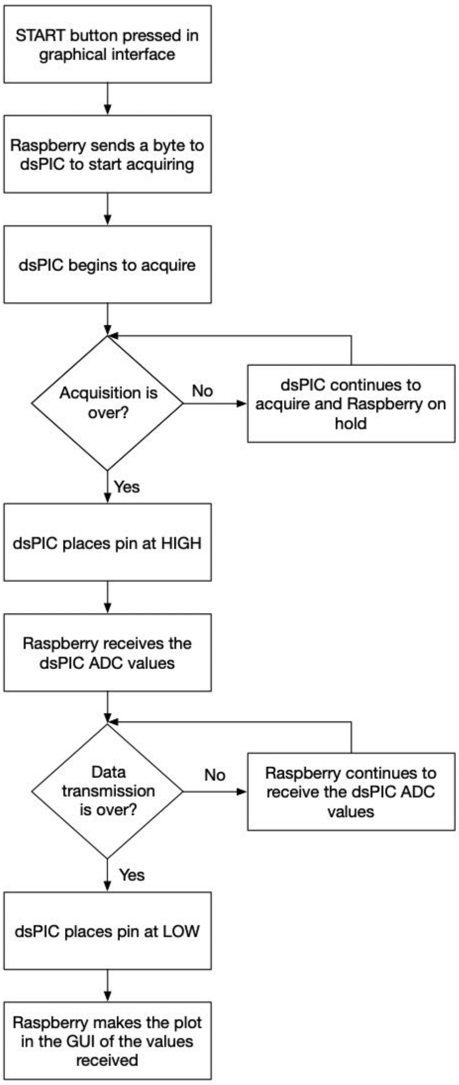

START button pressed in graphical interface
Raspberry sends a byte to dsPIC to start acquiring
dsPIC begins to acquire
Acquisition is over?
No
dsPIC continues to acquire and Raspberry on hold
Yes
dsPIC places pin at HIGH
Raspberry receives the dsPIC ADC values
Data transmission is over?
No
Raspberry continues to receive the dsPIC ADC values
Yes
dsPIC places pin at LOW
Raspberry makes the plot in the GUI of the values received

Figure 2: Flowchart signal acquisition

4. Results

4.1. Characterisation acquisition system

4.1.1 Conditioning

Programmable Gains

To verify the correct functioning of the gains, a sinusoidal signal of 1Vpp amplitude with a frequency sweep is applied, the minimum frequency is 10Hz and the maximum is 500Hz.

In the figure 3 we can see that the output signal is equal to the input signal as expected because the gain is one.

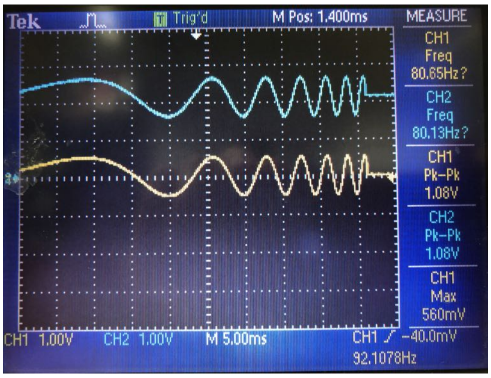


Figure 3: Frequency scanning gain 1, CH1 - original signal, CH2 - conditioned signal

In the figure 4 it is possible to verify that the gain is linear for the various working amplitudes and frequencies of the system.

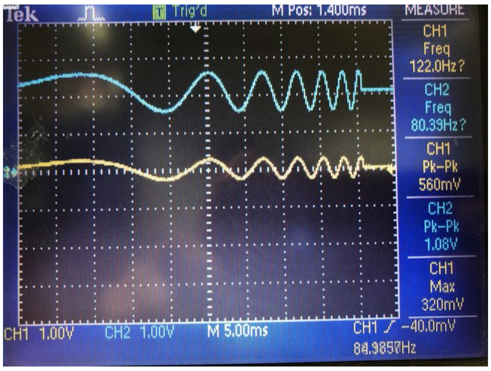


Figure 4: Frequency sweep gain 2, CH1 - original signal, CH2 - conditioned signal

In the figure 5 and 6 the noise of the higher gains of AI is quite high. However, it is possible to verify that the gain is linear in both situations

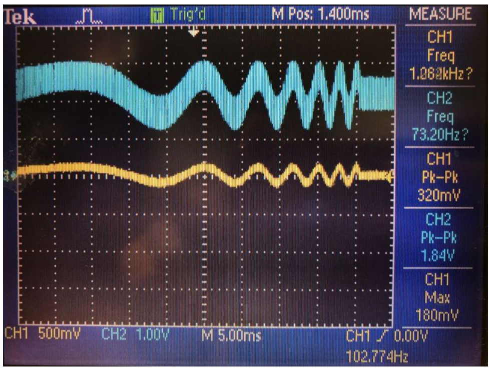


Figure 5: Frequency sweep gain 5, CH1 - original signal, CH2 - conditioned signal

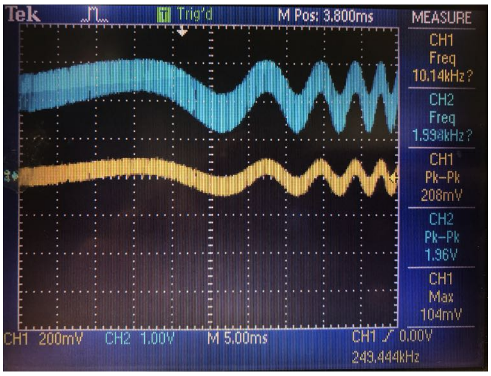


Figure 6: Gain frequency scan 10, CH1 - original signal, CH2 - conditioned signal

Bandwidth

To determine the bandwidth of the system, the frequency of a sinusoidal signal in the DDS was progressively increased until there was a decrease in the amplitude of the output signal of the conditioning board corresponding to 3dB. The measured bandwidth value was 7 MHz for a unit gain, which is much higher than the frequency at which the system will be used.

4.1.2 Acquisition signal

We started by analyzing the dsPIC ADC regardless of the conditioning circuit. For this, a sinusoidal wave with a variable amplitude was generated and the digital values obtained in the following table **??**.

Test Signal	Value read by dsPIC
3.10V	252
2V	160
1V	80
160 mV	8

Table 3: Results ADC readings from dsPIC

Voltage values greater than 3.10V saturate the

ADC, that is, the result is always 255. Values less than 160 mV correspond to a null result.

Although the ADC is 12 bits for this project only 8 bits were used, due to GUI. Thus the scale is from 0 to 255. Note that 0 indicates a reading of 0V and 255 of 3.3V.

Interpolating the values gives the following table 4 where it is possible to compare the expected values with the actual values.

Value 8 bits	Expected voltage value
255	3.3V
252	3.26V
160	2.07V
80	1.04V
8	0.103
0	0V

Table 4: Values obtained by the ADC and respective voltage value

Comparing the two tables above it is verified that the accuracy of the ADC is greater for values between 1V and 2V. For the values of both extremes the scan has errors in the order of 100 mV.

The DDS output was then connected to the input of the conditioning circuit and the conditioning circuit is connected to the ADC. In an analogous manner, the result of the ADC conversion was analyzed. These results are found in the table 5.

Signal Generated	Signal Conditioning	Value read by dsPIC
1V	3.08	247
0.5V	2.72	205
0V	2.28	162
-0.5V	1.64	120
-1	1.12	78

Table 5: Result read ADC taking into account conditioning board

In this part we intend to analyze signals with negative amplitude. For this, the scale of values is divided. Values between 0 and 128 indicate signals with negative amplitude and readings between 128 and 255 indicate positive signals, the following table 6 allows to realize the amplitude of the original signal taking into account the value read by the ADC

Value 8 bits	Expected voltage value
255	1.65V
247	1.53V
205	0.99V
162	0.44V
128	0V
120	-0.10
78	-0.644
0	-1.65V

Table 6: Values obtained by the ADC and respective voltage value

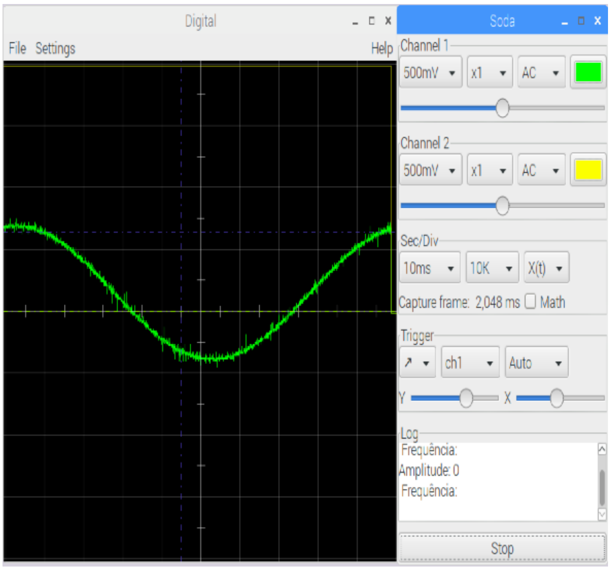
Comparing these values with the table tab:3 there is a big difference, because the offset that is placed on the second instrumentation amplifier should be a value close to 1.65V and not 2.048V. This is the only way to properly divide the scale by taking into account the voltages allowed by the dsPIC ADC. A second way of solving this problem is to change the reference voltage of the dsPIC ADC to match the supply voltage of the IA.

On this board it is possible that the output signal has negative voltages, which makes no sense and can be nocive to the ADC. One way to correct this problem was to place diodes that would only conduct if the signal was between 0-3.3V.

The requirement for the system scope has not been met, the maximum possible measuring value is approximately 1V and not 10V as predicted, this is due to the fact that the 1/5 resistive divider is not correctly implemented.

4.2. Sampling Rate

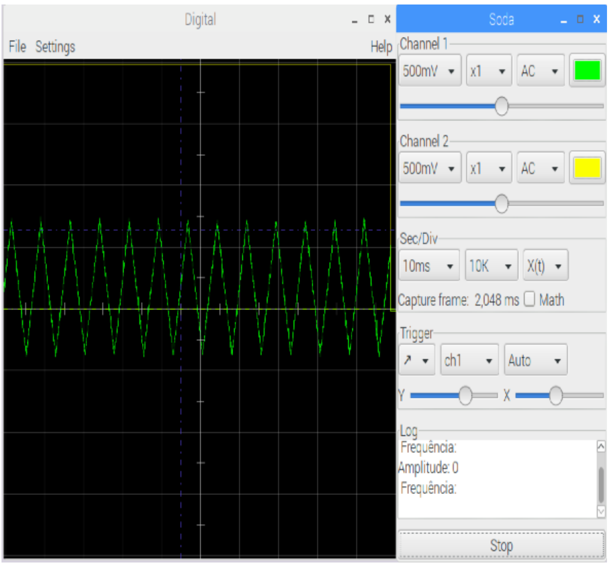
From the following figure 7 it is possible to estimate the sampling frequency. Taking into account that we are viewing a complete period of 66.66ms, and 5000 samples were acquired the sampling rate is at 75000 samples per second. This value is well below the spectable value (625kS/s). This difference is due to poor programming in the dsPIC, a wrong configuration of the main clock that will affect the interruption period.



Channel 1 500mV x1 AC
Channel 2 500mV x1 AC
Sec/Div 10ms 10K X(t)
Capture frame: 2,048 ms □ Math
Trigger ↗ ch1 Auto
Y X
Log
Frequência:
Amplitude: 0
Frequência:

Figure 7: Sine signal 1 Vpp 15Hz, T = 66.66ms

To exemplify what happens to the increase in frequency, in the following figure 8 one notices that an input signal of 200 Hz is practically imperceptible.



Channel 1 500mV x1 AC
Channel 2 500mV x1 AC
Sec/Div 10ms 10K X(t)
Capture frame: 2,048 ms □ Math
Trigger ↗ ch1 Auto
Y X
Log
Frequência:
Amplitude: 0
Frequência:

Figure 8: Triangular signal 1 Vpp 200Hz

4.3. Number of saved samples

Given that the window of the GUI allows to view 10000 samples, the dsPIC only has capacity for 8000 samples. We chose to store 5000 samples in each acquisition period because it is a multiple of 10000.

4.4. Transfer speed by SPI

The data transfer rate between the dsPIC and Raspberry is at 5 MHz because in the larger frequencies there were data losses, and the results were not consistent.

4.5. Power consumption

To measure the energy consumption of the various components an ammeter was connected in series and the following values were obtained

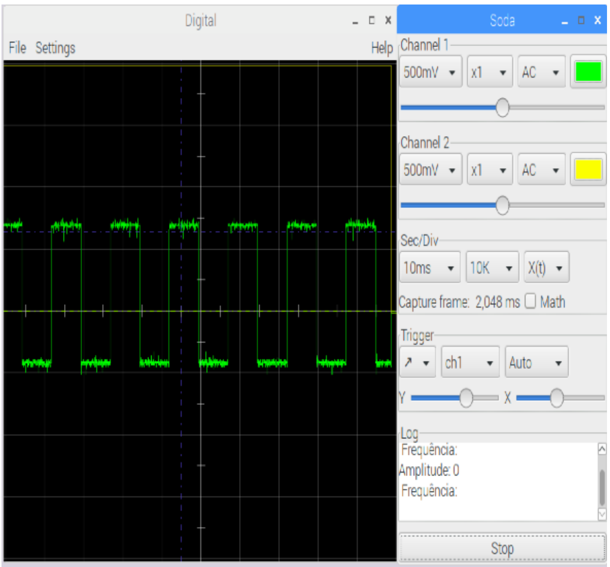
Conditioning circuit current:72 mA, this value does not depend on the gain value selected.

With respect to the consumption of the dsPIC in standby a value of 7 mA has been obtained, this value increases to 28 mA when it is being sampling. While the data is being sent to the Raspberry the measured value was 30 mA. Taking into account the dsPIC datasheet the standby values are within the parameters:8mA.

Since the feeding of Raspberry is made by USB, it was not possible to use the Agilent 34410A ammeter, in this case an ammeter was used which also allows measuring the run between two USB. The standby current on the Raspberry with the touchscreen is 600 mA and at the start it has a peak of 900 mA. With the GUI running, it consumes about 700 mA. The measured values are in accordance with similar tests [?]. The expected value is much higher (2000mA) this difference is due to a current meter error which could be quite high.

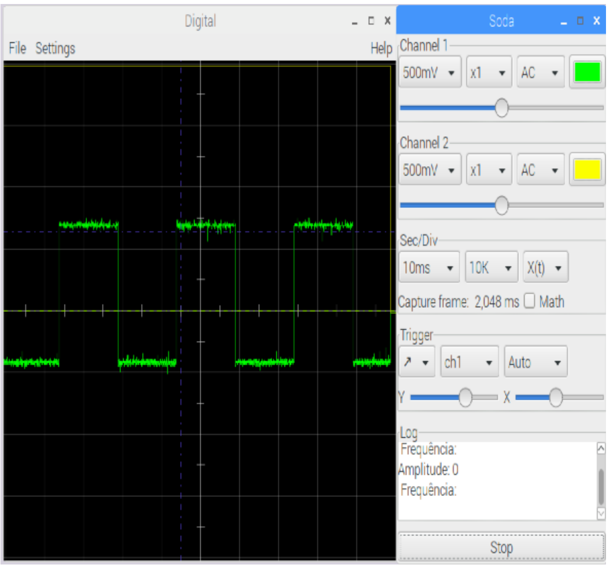
4.6. Some Signal Responses

In the following figures 9 and 10 there are 2 well-defined square waves at both 100 Hz and 50 Hz.



Channel 1 500mV x1 AC
Channel 2 500mV x1 AC
Sec/Div 10ms 10K X(t)
Capture frame: 2,048 ms □ Math
Trigger ↗ ch1 Auto
Y X
Log
Frequência:
Amplitude: 0
Frequência:

Figure 9: Square signal 1 Vpp 100Hz



Channel 1: 500mV x1 AC

Channel 2: 500mV x1 AC

Sec/Div: 10ms 10K X(t)

Capture frame: 2,048 ms ☐ Math

Trigger: ↗ ch1 Auto

Y X

Log:
Frequência:
Amplitude: 0
Frequência:

Figure 10: Square signal 1 Vpp 50Hz

5. Conclusions

The objective of this project was to create a device that allows to acquire and generate signals, both analog and digital. This system focuses on the versatility and practicality that is so much needed while working on electronics projects. For the development of this project was made a study of apparatuses with similar functionalities. The first block to be developed was a power supply that was not fully functional. Then the conditioning and acquisition blocks were built. The program became functional through a GUI adapted to this project. To date it has been possible to analyse analog signals up to approximately 200Hz.

In order to complete this work it is possible to develop the analog signal generation system ac DDS as well as the digital analyser. For the final system it is necessary to aggregate all the blocks in a single board that connects to Raspberry.

Acknowledgements

I would like to start by thanking my supervisors, Prof. Rui Rocha and Prof. Pedro Ramos for all the help, suggestions and availability given during the whole development of this thesis.

Secondly, I would like to thank Eduardo Félix, Francisco Lourenço, Miguel Silva, Pedro Feitor, Pedro Sousa, Rui Santos and Samuel Arcadinho for the friendship and support during these years at the Instituto Superior Técnico. Equally important in this last phase, Mr. Pina dos Santos, Nuno, Renato and Ruben were a great help in this project.

A special thank you to Carolina.

Last but not least, I would like to thank my family for the unconditionally support.

References

[1] BitScope. Bitscope micro oscilloscope and analyzer.

[2] Microchip. *12-Bit High-Speed, Multiple SARs A/D Converter (ADC)*, 2013.

[3] Microchip. *dsPIC33/PIC24 FRM, Oscillator Module*, 2013.

[4] Microchip. *Serial Peripheral Interface (SPI)*, 2013.

[5] nScope. nscope overview, 2016.

[6] J. Tech. Dso 138 diy kit, 2015.

[7] J. Tech. minidds diy kit, 2015.