Technische Universität Berlin

Institut für Elektrotechnik und Informatik
Fachgebiet Regelungssysteme

Fakultät IV
Einsteinufer 17
10587 Berlin

Bachelor Thesis

Development of an Automated Testbed for Electrical Stimulation Devices with Electrode Array Support

Nolan Russ Asensio

Matriculation Number: 394810
20.08.2018

Supervised by
Dr.-Ing. Thomas Schauer

Assistant Supervisor
Dipl.-Ing. Markus Valtin

Academic Supervisor
Francisco José Pérez Cebolla

Acknowledgements

First of all, I would like to express my appreciation to Dr. -Ing Thomas Schauer for providing me with a path to find a suitable Bachelor Thesis in the department.

I would also like to express my appreciation and special thanks to Dipl- Ing Markus Valtin for providing me with a topic for my Thesis as well as for the dedication demonstrated and thorough following of the Thesis to its completion.

Furthermore I would like to thank F.J. Pérez Cebolla for showing a personal interest in supervising my Bachelor Thesis and helping me with the academic procedures at my home University.

I would like to mention as well my special thanks to Astrid Bergmann for her patience and help when I required further technical instruction while working in the laboratory.

Lastly, I would like to take the time to thank Universidad de Zaragoza and Technische Universität Berlin for providing me with the opportunity to follow through with my Thesis in an international environment.

Hereby, I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, 21.08.2018

. . . .

(Signature Nolan Russ Asensio)

Abstract

This thesis summarizes the design, build and fabrication of an automated testbed for electrical stimulation devices used, for instance, in muscle rehabilitation. The testbed will be able to provide a level of assessment of the detected and measured stimulation signals from these devices. This task can become tedious and time-consuming when done manually, given the quick and short nature of the stimulation signals typical of said devices. The testbed will provide the user, backed up by the relevant PC Software, with an intuitive tool to easily measure and assess the stimulation signal from the input channel or channels of the stimulation devices and also the detection of the active channels as well as the active electrode arrays, usually connected to the skin.

The work behind the thesis mainly focuses on the hardware design and build of the testbed. The testbed was designed as a box whose inputs are the stimulation signals and outputs are the communication data packages to the PC via USB connection. Also, the main skeleton of the microcontroller's software was developed. Lastly, a prototype was built to test the hardware design and prove the developed concept. The testbed can also be used for future developed software debugging.

EAGLE CAD Software was used for the hardware design and to create the Printed Circuit Board (PCB) which was then sent to print and was soldered with the components identified in the design stage. The testbed implements the STM32F412RE microcontroller from ST Microelectronics, which is supported by the STM32CubeMX graphical software configuration tool and was used to aid in the generation of C initialization code. The C code was built in the SW4STM32 Eclipse environment, also provided by ST Microelectronics. A basic hardware testing program was flashed to the microcontroller aided by the Nucleo-F411RE board and the STM32CubeProgrammer software, both also developed by STMicroelectronics.

Resumen

La tesis resume el diseño, construcción y fabricación de un banco de pruebas automático para aparatos de estimulación eléctrica funcional (FES) utilizados, por ejemplo, en rehabilitación muscular. El banco de pruebas dará la posibilidad de poder evaluar la detección y medición de las señales estimulantes propias de estos aparatos. Esta tarea puede llegar a ser tediosa y requerir de mucho tiempo al realizarla manualmente dada la rápida y corta naturaleza de las señales estimulantes típicas de dichos aparatos. El banco de pruebas aportará al usuario, junto al software relevante, una herramienta intuitiva para medir y evaluar de una manera simple las señales estimulantes a partir del canal o canales de entrada de los aparatos de estimulación eléctrica funcional y también para detectar los canales activos así como de la matriz de electrodos activa, normalmente conectada a la piel.

El trabajo detrás de la tesis se basa fundamentalmente en el diseño del hardware y construcción del banco de pruebas. El banco de pruebas fue diseñado como una caja cuyas entradas eran las señales de estimulación y cuyas salidas eran los paquetes de información comunicados al PC por conexión USB. También fue desarrollado el esqueleto principal del programa de software para el microcontrolador. Por último, un prototipo fue construido para probar el diseño y demostrar el concepto desarrollado. El banco de pruebas también podrá ser utilizado para depurar cualquier futuro software desarrollado.

El software CAD EAGLE se usó para diseñar el hardware y para crear la placa de circuito impreso (PCB), la cual luego fue mandada a imprimir y fue soldada con los componentes seleccionados en la etapa de diseño. El banco de pruebas implementa el microcontrolador STM32F412RE de STMicroelectronics, el cual está respaldado por el software gráfico STM32CubeMX y que fue utilizado para ayudar en la generación del programa de inicialización en lenguaje C. El código en lenguaje C fue construido con el entorno SW4STM32 para Eclipse, también suministrado por STMicroelectronics. El programa básico desarrollado para probar el hardware fue flasheado al microcontrolador apoyado por la placa Nucleo-F411RE y el programa STM32CubeProgrammer, ambos también desarrollados por STMicroelectronics.

Contents

List of Figures

List of Tables

1. Introduction

The work involved in the thesis features the design, build and evaluation of an automated Testbed for electrical stimulation devices. These devices are used to trigger muscle contractions (e.g. for the purpose of muscle/CNS rehabilitation) by generating stimulation signals made out of repetitive quick pulses of relative short width.

1.1. Motivation

The idea behind the design of an automated testbed for the evaluation of functional electrical stimulation (FES) devices comes from the necessity to quickly assess the stimulation signals typical of such devices. While this assessment can be done with an oscilloscope, the measurement process may result tedious due to having to record the data from the measurement manually, specially for long term measurements (e.g. 30 minutes).

1.1.1. What is FES

FES stands for Functional Electrical Stimulation. It is based on a stimulation signal usually comprised of periodic short electrical pulses. It is widely used in the medical field to aid in rehabilitation of paralyzed muscles but it also has a use as a therapeutic technique. This stimulation signal is usually generated by FES devices and it is connected to the patient's skin by means of electrode array elements. The stimulation signal transmitted through the skin to the muscle aims to replace the nervous stimulation signals sent from the brain through the spinal cord in order to stimulate the paralyzed or weak muscle of the patient. Figure 1.1 shows a simple FES setup to the patient's forearm with the FES device SEM43.

1.2. Objective

The problem to tackle is focused on coming up with a way to measure the stimulation signal from FES devices in a way that the measurement is recorded into a virtual space, where it can then be assessed easily by the user. The automation of the process results really helpful due to the signal's very fast and short-width pulses at a relative low frequency (not higher than 100Hz).

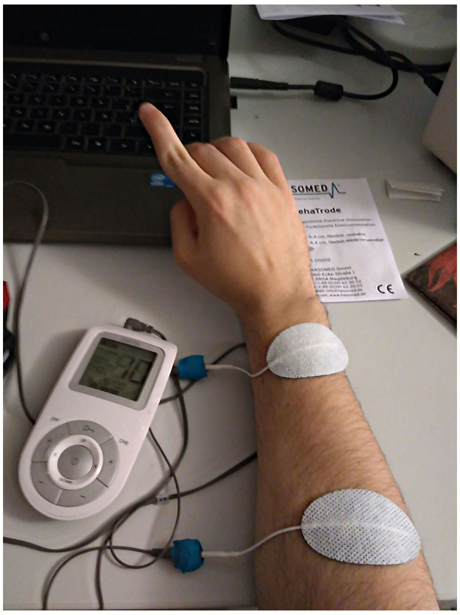


Figure 1.1.: FES example with the SEM 43 from Sanitas.

In short, FES devices feature short-width electrical pulses at low frequencies not higher than 100Hz and low current levels, between 1-150mA. They provide different modes of stimulation that give as output different waveforms for the same signal parameters like current, frequency and pulse-width. Figure 1.2 illustrates the typical waveform of a bipolar and symmetrical stimulation signal with a current peak value of 20mA and a pulse-width of 300μs.

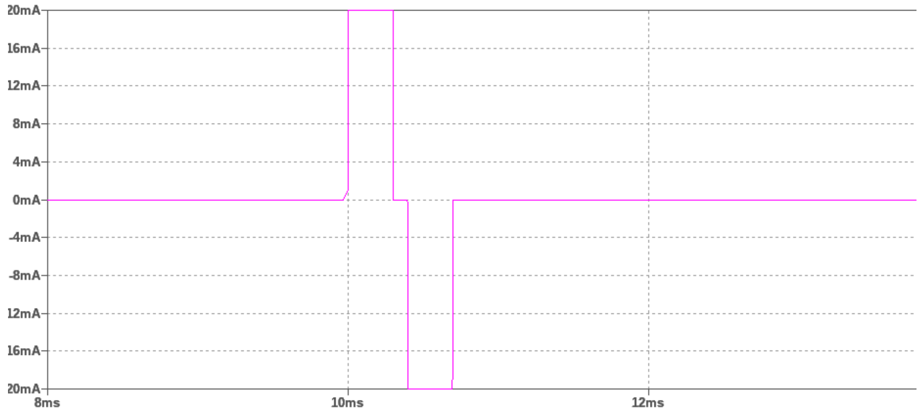


Figure 1.2.: Basic FES signal waveform.

Also, the testbed will allow to provide visual and software feedback to assess the active channels involved in the stimulation, whether if it is the stimulation channels and/or the electrode arrays which are usually coupled to the skin. This task is not currently possible by means of only measurement devices such as signal testers.

Lastly, the design of a testbed specific to these electrical stimulation devices allows for a specific design approach so that the stimulation signal is unaltered throughout the active detection and measurement.

1.2.1. Design Goals

It is important to understand the different kinds of signals these devices can generate, as the testbed must be able to be implemented in a general way and not dependent of the kind of stimulation mode it was chosen. In that sense, the objective is to be able to detect and measure an electrical signal that is current driven, whose application loads may resemble the skin, that can be unipolar or bipolar with no set current direction and very short pulse widths at relative low frequencies. So that it can be easily referenced throughout the project, the objective signal is summed up in table 1.1.

Current Pk	Pulse Width	Frequency	Shape	Direction	Polarity
1-150[mA]	15-400[μs]	typ. 20Hz	typ. PWM based	unset	any

Table 1.1.: Definition of target stimulation signal

The automated testbed should also provide the user with a visual feedback of the active stimulation channels and array channels involved in the stimulation. This can be really helpful when more than one stimulation channel is being generated at the same time and where the electrode arrays can become complex. A visual feedback provides an immediate assessment of whether the stimulation is happening as it should and if the stimulation device(s) are working properly.

Lastly, one of the most relevant objectives of the design is that the stimulation must pass through the testbed as unaltered as possible. This means the signal's waveform, frequency and current value should not be altered, since these signals are solely defined by these parameters.

It is of relevance to mention that the stimulation signal can be bidirectional, so it is important in the design stage to keep very present all the different types of signals that the stimulator devices can generate. If, in any case, limitations were to arise in the design, the testbed will have to be documented properly so that these limitations are mentioned and known by the user.

1.3. Proposal

The proposed solution to achieve this automation is based on an electrical circuit that can be structured in three main blocks: Active Channel Detection, Signal Measurement and Measurement Digitalization. These blocks would fall into the hardware's section of the testbed's design.

In order to comprehensively assess the measurements, it would be of interest to implement a PC interface. The PC interface would feature the data gathered from hardware in a comprehensive way by matter of either charts or a visual representation of the signals, just like an oscilloscope. It was proposed to implement this in a Matlab interface. Figure 1.3 shows a detailed block diagram that gives a more detailed insight of the data flow in the testbed as well as how each block contributes to the big picture.

The user would be able to control the stimulation signal to be sent by the stimulation device as well as to assess said stimulation signal and the full stimulation application in a PC interface. The signal would then pass through a demultiplexer where it would be routed towards the respective array elements. These array elements, active and passive, as well as the channels of the stimulation device, will be defined as the inputs of the testbed.

A demultiplexer is a device that allows one signal input line to be routed to an assortment of different digital output lines. In this case, the demultiplexer block allows for two signal input lines which then can be routed to their respective active or passive channels array elements.

The first stage the inputs go through in the testbed is the active detection block, where the detection of the active source channel or demultiplexer channel will also provide a visual feedback a part of sending the respective digital signal to the MCU. This block will feature a modular design so that easy inclusion of more channels and/or arrays is simple and so that the total number of inputs does not affect the overall performance of the testbed. Due to the possibility of the stimulation signals being bipolar, the detection must also be bidirectional, meaning the both positive and negative stimulation pulses must be detected, independent of which appears first.

Once the signals were successfully detected, they should converge again, unaltered, before the measurement block. At this stage, the signal should be ideally equivalent to the initial stimulation signal from the active channel of the stimulation device. The testbed will intrinsically feature three measurement models, 1kΩ Model, Skin Model and Audit Model, with the possibility to include whatever external model were necessary. The relevant measurements at this stage refer to the stimulation signal's current and voltage, which will then be sampled and digitalized by an ADC or ADCs.

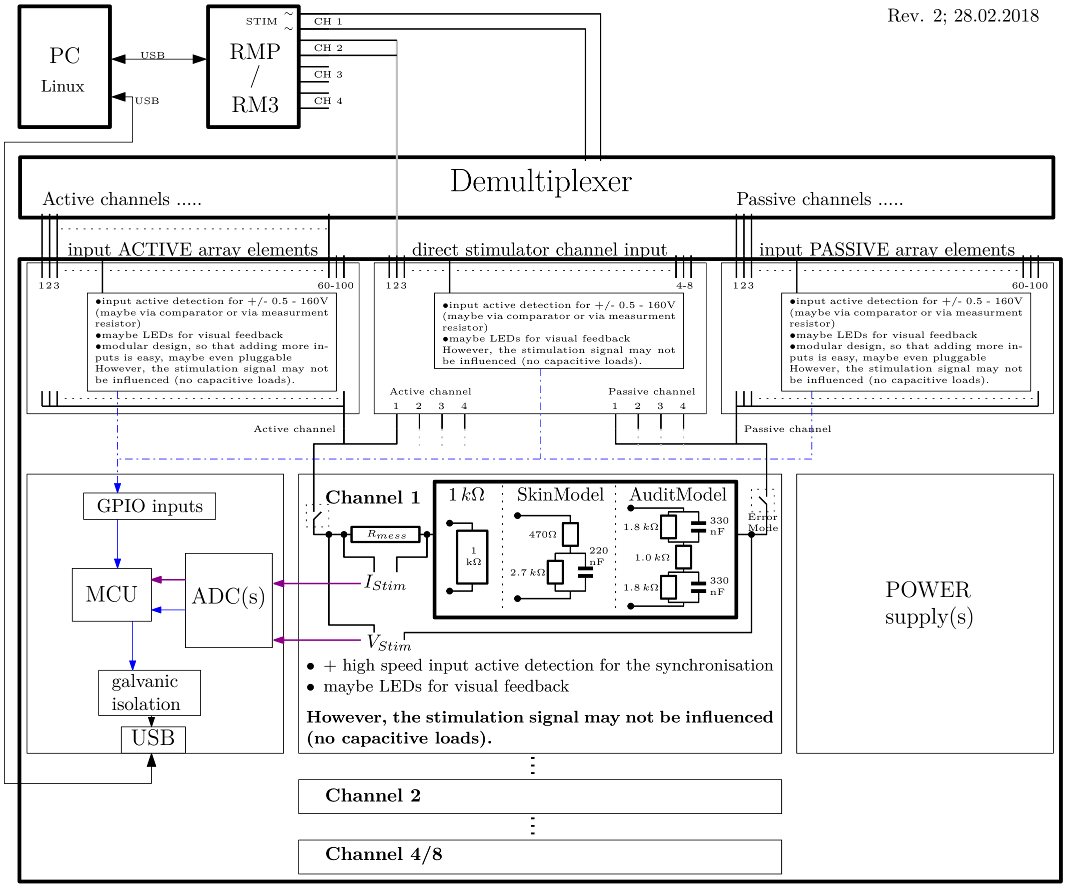

Rev. 2; 28.02.2018
PC
Linux
USB
USB
STIM
RMP
/
RM3
CH 1
CH 2
CH 3
CH 4
Demultiplexer
Active channels .....
Passive channels .....
input ACTIVE array elements
direct stimulator channel input
input PASSIVE array elements
1 2 3
60-100
1 2 3
4-8
1 2 3
60-100
•input active detection for +/- 0.5 - 160V (maybe via comparator or via measurment resistor)
•maybe LEDs for visual feedback
•modular design, so that adding more inputs is easy, maybe even pluggable
However, the stimulation signal may not be influenced (no capacitive loads).
•input active detection for +/- 0.5 - 160V (maybe via comparator or via measurment resistor)
•maybe LEDs for visual feedback
However, the stimulation signal may not be influenced (no capacitive loads).
•input active detection for +/- 0.5 - 160V (maybe via comparator or via measurment resistor)
•maybe LEDs for visual feedback
•modular design, so that adding more inputs is easy, maybe even pluggable
However, the stimulation signal may not be influenced (no capacitive loads).
Active channel
1 2 3 4
Passive channel
1 2 3 4
Active channel
Passive channel
GPIO inputs
MCU
ADC(s)
galvanic isolation
USB
Channel 1
R_{mess}
I_{Stim}
V_{Stim}
1 kΩ
1 kΩ
SkinModel
470Ω
220 nF
2.7 kΩ
AuditModel
1.8 kΩ
330 nF
1.0 kΩ
1.8 kΩ
330 nF
Error Mode
• + high speed input active detection for the synchronisation
• maybe LEDs for visual feedback
However, the stimulation signal may not be influenced (no capacitive loads).
Channel 2
Channel 4/8
POWER supply(s)

Figure 1.3.: Block Diagram of the Testbed.[1]

The digitalized measurements will be later sent to the MCU which will send them together with the GPIO Inputs from the detection block via USB connection to the PC.

1.4. Outline

The thesis follows a structure focused on the development steps that took place for the design, build and test of the testbed.

This structure is detailed as follows:

Chapter 2 is focused on explaining the theoretical background behind each circuit design block and enunciating all the technologies involved in the testbed, from a hardware perspective. It explains what is needed from a technological point of view and tries to summarize the way in which said technology is implemented.

Chapter 3 refers in detail into the proper design and development of the testbed's hardware. It focuses in thoroughly explaining the testbed's schematic and the PCB layout, detailing which devices and components are involved, in relation to the testbed's design requirements detailed in the section prior.

Chapter 4 focuses on the required developed software for the MCU so that the hardware works as it was intended to do so. It details the communication protocols involved as well as it briefly explains how the code should be implemented.

Chapter 5 describes the PCB construction steps including an evaluation stage of each block and the issues encountered during testing.

Chapter 6 provides a summary of the thesis and describes the biggest problems encountered and what other future work can be developed.

2. Testbed: Hardware I

This chapter is intended to detail and explain the Testbed's hardware requirements in relation to what devices and technologies are needed as well as the electronics theoretical basis involved in the design for each design block.

2.1. Channel Stimulation Detection

In this block, the objective is to detect which channels are active. Meaning, in which channel is the stimulation signal being generated and through which array elements is the signal being transmitted to. The purpose of this detection is to test that the demultiplexer is working correctly and to ensure that the desired signal given to the load model comes from the right source. It also helps to ensure that the array elements are properly connected. Both a Visual Feedback and a Signal Feedback are configured; visual by means of LEDs and signal because a digital signal stating whether it is active or not is transmitted to the MCU.

After testing various design approaches, it was concluded that it was needed to detect the positive and negative active signals separately. The detection is solely based on current detection which doesn't add any capacitive loads that could drain current from the main signal and alter the waveform of the stimulation signal.

2.1.1. Current Detection

The design features three diodes in parallel between positive stimulation source and load and another three diodes in parallel between negative stimulation source and load in order to have a bidirectional channel detection. This approach allows for the stimulation device and the load to always be connected. The current will always have a path from one pint to the other, no matter the polarity of the pulse.

Each three-diode branch structure, features two-diode branches pointed in the same orientation of the desired signal to detect and one other branch in parallel pointed in an opposite orientation. Figure 2.1 shows the three-branched current detection for both signal polarities.

No capacitive loads are present in the circuit except for the small parasitic capacitance components of the pn-junctions relevant to the diodes. However, due to the normal

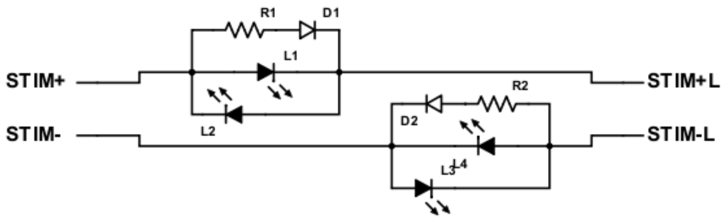


Figure 2.1.: Current detection for 1 Channel.

operating conditions in a low frequency (typ. 20Hz), this parasitic capacitance can be discriminated. In short, it can be concluded that the current passes through the detection circuit almost unaltered. Only a slight voltage drop will occur between input node and output node corresponding to the diode's forward voltage, V_f. This voltage drop solely depends on the type of LED device used in the design, so it can be chosen to be as minimum as possible. R1 and R2 limit the current through D1 and D2, respectively.

2.1.2. Signal Feedback

In order to have a logical signal to describe the current detection, an optocoupler is used. An optocoupler integrates an LED that emits infrared light and a photosensitive device which detects light from the LED. This is helpful in the sense that it allows to send an electrical signal between isolated circuits. In the testbed's case,the isolated circuits would correspond to the stimulation signal related circuits, such as signal measurement and current detection circuits, and the digital block. Figure 2.2 shows the schematic design of the detection block for detection of the positive signal of channel 1.

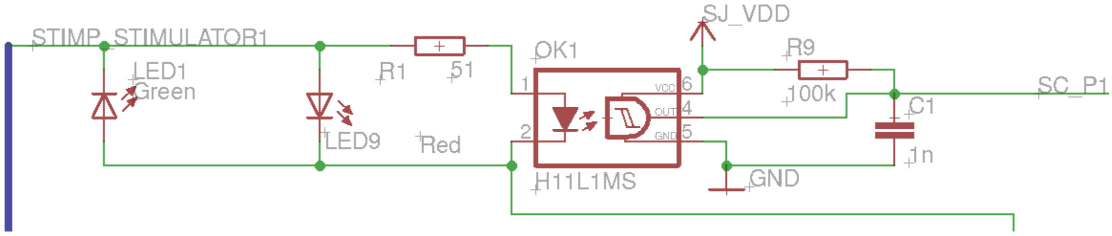


Figure 2.2.: Schematic design for the active detection of signal STIMP_STIMULATOR1.

The resistor in series with the optocoupler, whose optimal value was tested to be 51 Ohms, has the sole purpose of limiting the current through the optocoupler. This value, however, also had to be small enough so that the Voltage drop at the Current Detection stage wasn't comparable to the full Voltage drop at the load.

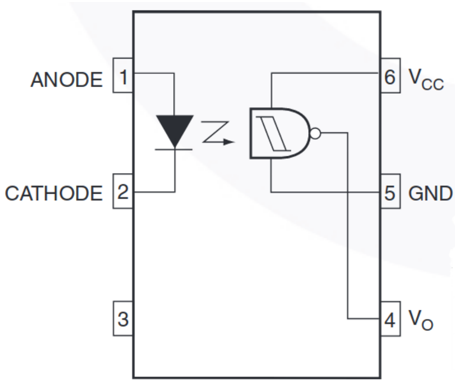


Figure 2.3.: Internal schematic for the H11L1 optocoupler.

The optocoupler chosen for this task was the H11L1M from Fairchild. The H11L1M's datasheet stated the ouptut V_o pin to be PULL-UP, so a resistor of 100k from Vcc to OUT works as PULL-UP Resistor. Then, a capacitor of 1nF has the sole purpose of enlarging the time the output is active for easier reading. Figure 2.3 shows the internal schematic layout of the H11L1M optocoupler as specified in its datasheet[2].

2.1.3. I/O Expander

It is possible for stimulation devices to be connected to a high number of array elements. Which is why it is of interest to design the testbed with the capability of easily expanding the number of channels to detect. That is why one of the biggest design constraints that was followed for the Active Channel Detection was for it to be modular, that's to say, easy to duplicate and implement for more channels if necessary. So, in short, if there were 100 array elements connected to the stimulation signal, that would mean 100 output signals from 100 optocouplers. Since these optocouplers will give one digital output signal each to be directed to the MCU for further processing, the inclusion of I/O expanders as intermediaries between optocouplers and MCU is necessary. The use of I/O Expanders keeps a certain number of individual logical values in a circuit grouped into one device which can then be rightfully controlled by the MCU with less pins than the actual task would require.

Some general Purpose I/O Expanders can rely on I2C technology in order to transmit data between all the GPIOs connected to it and the MCU. It depends on the Serial Data Line (SDA) and the Serial Clock Line (SCL) to establish the connection between Master and Slave. In this case, the MCU works as the Master and the I/O Expander as the Slave. The connection is established when the Slave receives from the master the correct address the Slave has been assigned by means of the Ax pins. The master then ends transmission with a stop bit. The address of the I/O Expander has been left to be manually configured by means of jumpers that will either connect the address pin to VCC or ground. The idea behind leaving this configuration to the user is because of the expansion capabilities of the testbed. The design features two IOE of 16 pins each. So, more IOE would be required in order to accommodate for a case of 100 channels. Manual address configuration ensures that maximum number of slaves to one master can be achieved, if needed.

The specific IOE chosen is the TCA9535 from Texas Instruments. It provides two I/O ports of 8 pins each and a PULL-UP interrupt pin, which will be very handy to control when the pin readout is activated. The SCL and SDA pins from the TCA9535 are also PULL-UP pins, so they were configured accordingly by means of a 10k resistor to VDD. Figure 2.4 shows the final schematic pin layout for the I/O Expander of the positive channel detection.

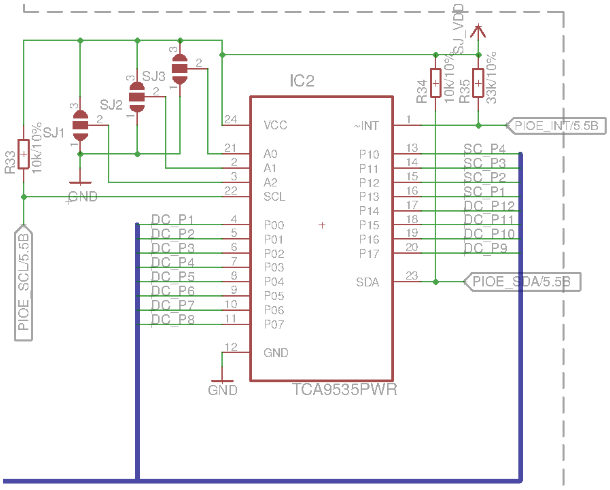


Figure 2.4.: Schematic pin layout of the TCA9535 for positive channel detection.

2.2. Stimulation Measurements

At the moment, there are three load models for the stimulation signal assessment. A 1k Ohm Resistor model allows for a proportional current to voltage conversion following Ohm's Law, a skin model allows to test the signal as if it were connected to the skin of a subject, and an audit model. The testbed also features open connections to allow extra Models to be tested. The choosing of the model to be tested can be manually configured by means of the respective jumpers. Figure 2.5 shows the abstract block design of the necessary stimulation measurement block including the different available load models.

In order to simplify the explanation and calculations of the maximum and minimum ratings of the measurements, the design was done for the 1kΩ Model. The difference

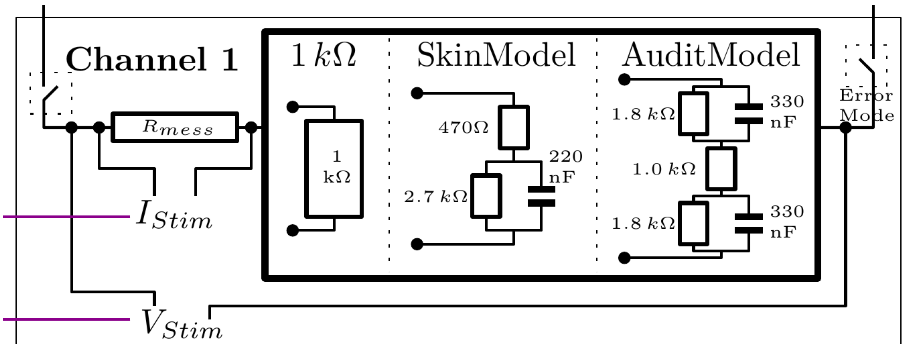


Figure 2.5.: Amplified view of the measurement block from the diagram block presented in Chapter 1.

from one model to another is the capacitive component of the equivalent load, meaning for 1k model a 0F capacitance component.

Before any measurements can occur, it is important to reference the input stimulation signal to the same voltage reference the measurement circuits will have. That's to say, the testbed's ground. For this, a 0Ω resistor must be connected at the load's positive line and ground. This resistor will reference the voltage drop of the load as well as the voltage drop at the measurement resistor.

2.2.1. Stimulation Voltage Measurement

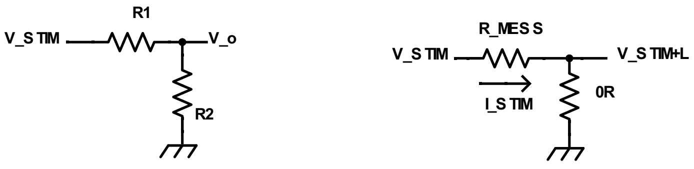
Once the reference is sorted, the voltage drop across the load model can be measured. This measurement will be from the ground reference connected and the other node of the load. Since the reference is connected to the positive stimulation input it only makes sense to measure the voltage at the negative stimulation input, as shown in Figure 2.6a.

Since the stimulation's current ratings are from ±[1mA..150mA], following Ohm's Law that translates to ±[1V..150V] for the 1k model. This voltage window must first be conditioned and lowered to a working voltage level limited by the ADCs input voltage range. In order to keep low power consumption throughout the design, this supply level for the analog part of the testbed was set to ±5 V. So, this means the voltage coming from that node needs to be much lower. This can be achieved by means of a Voltage Divider. The resulting voltage from the voltage divider is as follows:

$$V_o = \frac{R_2}{R_1 + R_2} V_{stim-} \tag{2.1}$$

For a voltage ouptut of maximum 1V, the stimulation's voltage must be divided by a factor of 1.5e6. So, for a R_2 of 10k, R1 must be 1M49 Ohm. Resistors in series can be used to adjust the factor. A very high impedance ensures very low current demand for the voltage measurement. A source follower allows to isolate the measurement from the

Signal Conditioning circuit, ensuring no more current is demanded from the stimulation signal and a proper impedance coupling.



(a) Voltage measurement (b) Current measurement

Figure 2.6.: Measurement circuits

It was discussed to keep the measurement operating range from -1V to +1V due to the fact that the ADC will most likely require a unipolar analog input. Meaning, the measurement will have to be elevated so that the whole range is over 0V. The full measurement range would be 2.5V for a ±1V with an extra safety voltage level of 0.25V at each side, which fits properly with the requirements of the ADC its analog side is supplied with a +5V voltage source.

2.2.2. Stimulation Current Measurement

The stimulation's current measurement is simply a direct implementation of Ohm's law, as shown in Figure 2.6b. By ensuring a certain impedance, the measurement of the voltage drop will be directly proportional to the current. So, for a 150 mA current stimulation, we would like to have a 1V voltage drop through $\mathrm{R_{Mess}}$ for the same reason as in the stimulation's voltage measurement. This means $\mathrm{R_{Mess}}$ should have a value of about 6.7 Ohms.

$$V_o = I_{stim} R_{mess} \tag{2.2}$$

We can adjust the factor by means of a resistor in parallel to $\mathrm{R_{Mess}}$
Using 1V as maximum rating for the current measurement ensures low voltage drop from the stimulation's signal. Analog to the voltage measurement, a source follower allows to isolate the measurement from the Signal Conditioning circuit, ensuring no more current is demanded from the stimulation signal.

2.2.3. Active Stimulation Detection

It is essential to know when the positive and negative pulses take place in the time domain. In order to time stamp these events, comparators are used to get a logical output that will describe the events. The output from the comparators will be either high (VDD) or low (0V), whether the signal from the current measurement is higher or lower in respect to V_{ref}. The signal's Rising Edge and Falling Edge can then be detected by software to construct the time stamp. The comparators will make use of a voltage divider to establish the reference voltage. This voltage reference will then have to match the signal-to-compare's reference voltage. A manual adjustment of the voltage divider will allow for proper matching with the actual signal-to-compare's offset of the theoretical value.

As later explained in section 2.3.3, there is the need to elevate the measurement signal to have a unidirectional signal in order to provide a differential input of the measurement for the ADC block. By using this converted to unidirectional signal we can actually take advantage of the unipolar quality of said signal as input for the comparators. This allows to design the comparator block in the digital block itself isolating the comparator block from the analog block by means of a source follower. The detection can only be designed at the current stimulation measurement branch for the following reason. In the voltage stimulation measurement branch, this comparator block would alter the signal conditioning for the ADC and an accurate measurement wouldn't be able to be assured. Low power rail-to-rail comparators can be chosen for this task due to the signal's squared short pulse-width and the requirement of a fast slew rate.

Two comparators can be used in order to detect the positive and negative pulses. By having as V_{ref} the same voltage as the elevated voltage in the signal conditioning for the ADC, the comparators will provide a high and low state when the elevated signal is either over or under V_{ref}. In addition to allowing an operation with these voltage values, the comparators must also be fast enough so that the transition between high and low states is practically instantaneous for the pulse widths they are supposed to detect.

The chosen comparators for this task were the TS3021 from ST Microelectronics. As stated in the device's datasheet, "The TS3021 single comparator features high-speed response time with rail-to-rail inputs. With supply voltage specified from 2 to 5 V, this comparator can operate over a wide temperature range: -40°C and 125°C"[3].

Lastly, the comparator block is suited with visual feedback with a one LED and a 1kΩ resistor branch connected to each of the two comparators' outputs. This visual feedback not only will provide for a physical way to assess whether the measurement is working properly or not but it will also help at debugging stages to ensure the MCU is programmed correctly in relation to detecting when the pulses take place.

2.3. ADC

The testbed required a digitalization of the signal outputed to the load, so that it can be later reconstructed at PC level. This means the measured signals, which are an analog bipolar voltage with a range of ±1 V, need to be converted to a digital signal readable by the MCU. This digitalization is done by means of a 16 bit ADC.

2.3.1. ADC Parameters

Before choosing the ADC to implement in the Testbed, it was important to lay out the requirements of the design to find which ADC would fit best.

Resolution

It's already been mentioned the ADC would have a 16 bit resolution. The ADC's resolution determines the accuracy between digital and analog signal, that's to say, it determines how accurate the reconstruction of the analog signal can be from the digital converted signal. In short, an ADC with a higher bit resolution will ensure greater reconstruction accuracy but heavier signals to process. Reconstructions from an ADC with a low resolution can translate into choppy reconstructions that deviate from the original signal.

In this sense, a 16 bit resolution will prove to be enough for the voltage range of the measured signal, 2.5V. This is proved with Equation 2.3.1 where the Least Significant Bit calculated relates how much voltage value one bit compares to. The lower the number, the easier will be to have a faithful reconstructed signal from the digitalization.

For a ±1V voltage range where the smallest signifcant value will be of the order of mV, the LSB should be able to provide a voltage resolution capable of said voltage order. After applying Equation 2.3.1, we get a 38.15μV/bit resolution, which shows the LSB to be three orders below the order of the minimum signifcant voltage values desired.

$$LSB = \frac{V_{max} - V_{min}}{2^n} \tag{2.3}$$

Frequency

16 bit ADCs in the market range up to 3MSps without being too costly. So, even if there are very fast and very accurate ADCs in the market at the moment, the ADC should be tailored to the design's needs to keep budget tight.

As a requirement, the time-based parameters of the measured signal come into play. The PW of the stimulation signal can range from 11 to 4000 μs in 1μs steps. So, the optimal sample time the ADC should feature would be of 1 μs. Keeping Nyquist-Shannon's sampling theorem in mind, the ADC should provide a sampling rate of 0.5μs. In ADC standards, this translates to 2MSps.

Input Signals

Two analog signals are the signals of interest to convert to digital. These signals correspond to the current measurement and voltage measurement of one channel of stimulation signal to load. This means the design requires two simultaneous ADC conversions, that's to say, two simultaneous sampling ADC cores. The best way to ensure the conversion is actually simultaneous is with an ADC that integrates this feature. This is helpful over picking two identical ADCs because, if they are separate, we cannot ensure they will behave identically for the same operating conditions and at may be hard to delimit these different behaviors. Having both ADCs encapsulated in the same IC, we limit the cases where this can be an issue and, if they still show any misbehavior from ideal conditions, the changes will be reflected and documented in the component's datasheet as error.

Reference Voltage

Given the measurement signals are bipolar, the ADC will have to allow for a balanced, differential input range. Usually, this is referenced by a reference voltage $\pm V_{\text{ref}}$. However, the voltage reference must encapsulate the full voltage spectrum of the measured signals. So, for a $\pm 1\,\text{V}$ of input signal plus an extra 0.25V for the buffer , the ADC must either include an internal V_{ref} of 2.5V and/or allow for an external 2.5V reference.

2.3.2. The MAX11198 as the Testbed's ADC

The component's datasheet defines the MAX11198 as "a dual-channel SAR ADCs with simultaneous sampling at 2Msps, 16-bit resolution, and differential inputs"[4]. It also features an internal 2.5V reference and reference buffers. It comes in a tiny 16-Pin, 3mm x 2mm, TDFN Package, that will greatly contribute to keeping the Testbed's full size limited.

The MAX11198 allows for SPI communication in order to transmit the digital outputs. These outputs can be transmitted simultaneously as they feature one pin for each digital output. It also includes an input pin that triggers the conversion. This pin is really helpful in this application since it is known when the conversion needs to happen. The conversion needs to be periodic and constant through the testbed's operation, so a clock signal could be connected to the CNVST pin as first approach. Figure 2.7 describes the IC's functionalities with a block diagram.

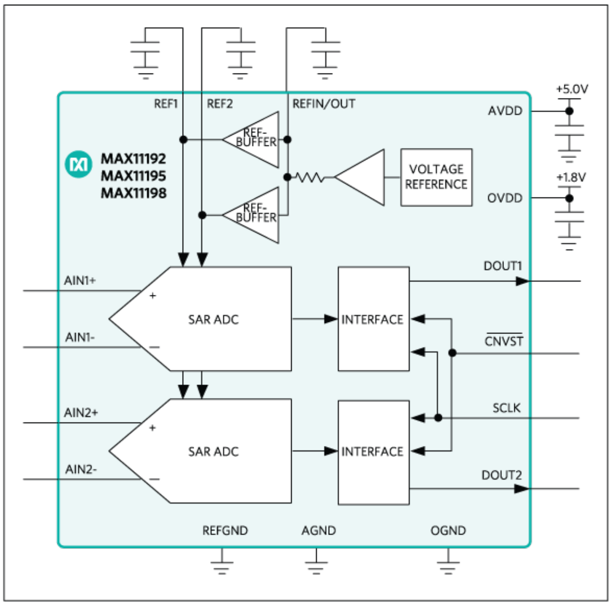


Figure 2.7.: MAX1119x: Functional Diagram

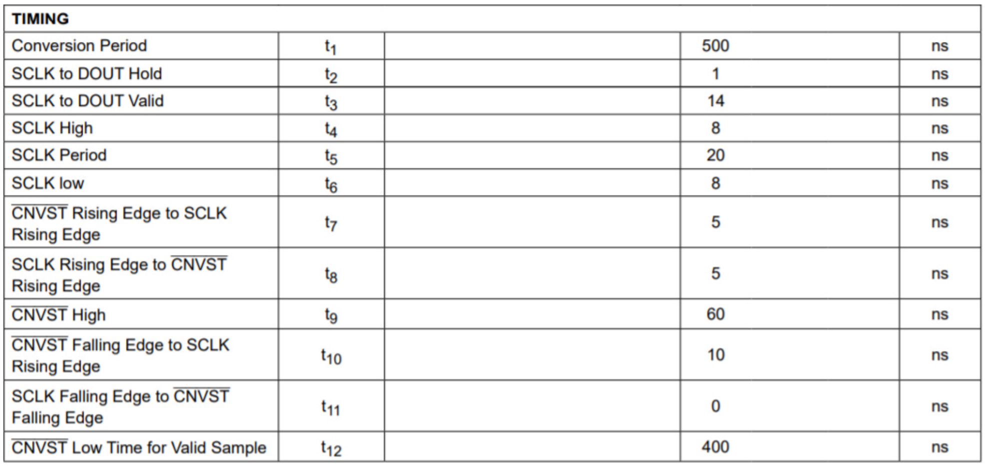
Time parameters

It is important to understand the ADC's limits concerning the timing, since they need to fit not only the testbed's demands but also the MCU's capabilities. In fact many of the features of the ADC previously mentioned highly depend on timing for the design to work properly. For instance, as already mentioned, the conversion time will play a big factor but of equal importance is the clocking time the ADC demands the MCU for the SPI communication to work properly. Figure 2.8 shows the timing values relevant to the MAX11198[4].

2.3.3. Measurement Signal Conditioning to ADC

The datasheet for the chosen ADC, the MAX11198, states that "the analog inputs of the MAX11198, AINn+ and AINn-, should be driven with balanced differential signals. The input signals can range from 0V to VREF"[4]. It has been discussed that VREF in this design had to be 2.5V. Refer to section 2.3.1.

However, the measurement signal is a Bipolar Single-Ended input, so the measurement signal has to be conditioned to fit the ADC's allowed input signal. The signal conditioning circuit designed basically elevates the whole signal by 1.25V, the maximum amplitude the signal can experience with positive and negative current. The idea is to have the whole signal in the positive spectrum, fitting the ADC's input requirements where it stated that the input signals must range from 0V to VREF, being VREF 2.5V. Since it



TIMING				
Conversion Period	t_1		500	ns
SCLK to DOUT Hold	t_2		1	ns
SCLK to DOUT Valid	t_3		14	ns
SCLK High	t_4		8	ns
SCLK Period	t_5		20	ns
SCLK low	t_6		8	ns
$\overline{\text{CNVST}}$ Rising Edge to SCLK Rising Edge	t_7		5	ns
SCLK Rising Edge to $\overline{\text{CNVST}}$ Rising Edge	t_8		5	ns
$\overline{\text{CNVST}}$ High	t_9		60	ns
$\overline{\text{CNVST}}$ Falling Edge to SCLK Rising Edge	t_{10}		10	ns
SCLK Falling Edge to $\overline{\text{CNVST}}$ Falling Edge	t_{11}		0	ns
$\overline{\text{CNVST}}$ Low Time for Valid Sample	t_{12}		400	ns

Figure 2.8.: Excerpt from Electrical Characteristics of the MAX11198 datasheet.

has to be balanced at AINn+ and AINn-, the previous measurement signals' 0V should then be half VREF after conditioning. Figure 2.9 shows the designed signal conditioning block, with the resistor values specified for the desired application generating the required differential input signal.

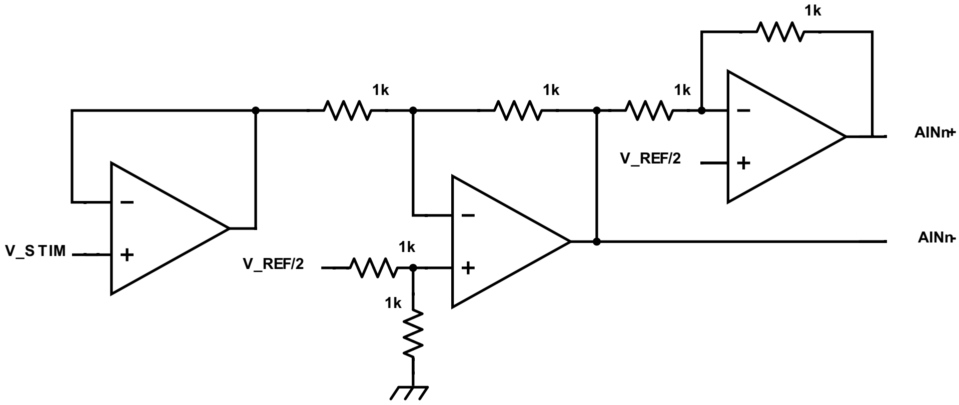


Figure 2.9.: Signal Conditioning circuit design

2.3.4. Anti-Aliasing Filter

An Anti-Alias Filter has the sole purpose of limiting the frequency bandwidth of the signal to sample in order to ideally block unwanted signals that induce error to the measurement better known as noise. In practice, however, these signals are not blocked but greatly attenuated the higher they are from what's known as the cut-off frequency. Figure 2.10 roughly illustrates the comparison between ideal and real application as seen on a short paper posted on-line by National Instruments [5].

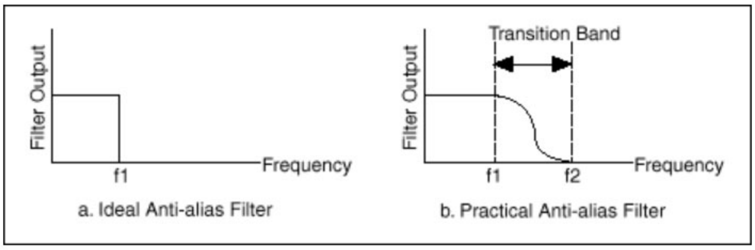


Figure 2.10.: Ideal and Practical frequency diagram of a low-pass filter.

Cut-off frequency f_c

An anti-alias filter is, in essence, a low-pass filter. A low-pass filter attenuates all frequencies above the cut-off frequency, $\mathrm{f_c}$. This frequency is recommended to be at least a decade less than the ADC modulator sampling frequency. In our case for a 2MSps, a $\mathrm{f_c}$ of 200kHz will knock down out-of-band noise at those frequencies by a factor of 10 or more (for a filter of Order 1).

Once the cut-off frequency is established, the desired low-pass filter can be designed. We must consider that the Anti-Alias filter has to be designed for a Differential input, which is actually an advantage in the sense that by including a differential filter, we ensure a high CMR.

Resistor design and Thermal Noise

The goal of the Anti-Alias filter is to limit noise from the original signal, so it is a priority to not add any more noise to the signal, which resistors connected in serial are prone to do by means of Thermal Noise. Thermal noise is defined as an error voltage by the following equation:

$$v_n = \sqrt{4kTR} \tag{2.4}$$

where k is known as the Boltzmann constant k = $1.38\mathrm{e}^{-23}[\frac{J}{K}]$, T[K] represents the component's temperature and R[Ω] the resitor's impedance. Choosing a 51Ω resistor gives a 1nV noise Voltage for a 125°C, maximum temperature operating condition for the ICs).

In short, we will have a 51Ω resistor in serial at VINn+ and another 51Ω resistor in serial at VINn-, with a maximum thermal noise of 1nV(T = 125°C) each. Then, the filters total resistance impedance R_s will be 102Ω.

Inductor and Capacitor design

Since f_c is higher than 100kHz, it is recommended the filter includes an inductor. Inductor and capacitor required values in the design are calculated as follows:

$$L_s = \frac{R_L}{2\pi f_c} \tag{2.5}$$

$$C_s = \frac{1}{2\pi f_c R_L} \tag{2.6}$$

R_L will be the same impedance as R_s. So, for $R_s = 102\Omega$ and $f_s = 200$kHz we get values of $L_s = 81.17\mu$H and $C_s = 7.8$nF.

A high enough capacitance ensures more charge available for the sampling capacitors inside the ADC. However, the higher the capacitance, the slower the discharging rate which can gravely impact the correct working of the design. 7.8nF is higher enough than the sampling capacitors, which are usually in the order of 10-12pF. Figure 2.11 shows the designed anti-alias filter.

There will be one anti-alias filter for each ADC. The anti-alias filter would be connected between the now elevated and differential measurement signals and the inputs to the ADC.

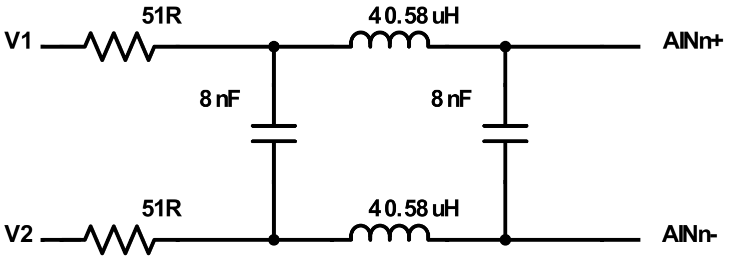


Figure 2.11.: Designed Anti-Alias Filter

2.4. MCU

In any electronics design, it is necessary a sort of brain that controls and monitors everything that goes on in the design, ensuring proper communication and timing between devices. This task solely relies on the MCU. In order to find the suitable MCU for the design, a proper understanding of what is demanded by the devices and the task at hand is required.

A thorough list of all the interfaces the MCU would have to include was conducted by naming the different technologies each device would use to communicate with the MCU, taking into account, as well, the total number of pins required in the design. In addition, it was of relevance a fair estimation of the memory requirements for the design and the related speed for the transfer of said chunks of memory. Lastly, some expansion requirements were kept in mind so that the MCU would allow for said expansion without having to require another MCU that wouldn't fit the final designed layout or would require a major redesign of the schematic and PCB layout.

2.4.1. Requirements

Putting into perspective the environment and working conditions the MCU will be submerged in, the MCU will have to allow for the fastest communications possible so that the calculations and data transmission between devices does not interfere with the signal's actual speed.

Speed is not the only requirement for the MCU, however. As stated above, the communication technologies involved as well as the memory capability actually may be more relevant and could limit more deciding the correct MCU for the design.

Interface requirements

As stated in section 2.1.3, the technology involved in the communication with the I/O Expanders was an I2C protocol.

In short, the I2C communication protocol "allows multiple slave digital integrated circuits (chips) to communicate with one or more master chips." [6] It only needs two signal wires (SCL and SDA) and is intended for short-distance communication, more than enough for the I/O expanders.

The TCA9535PWR, as stated in its datasheet[7] allows for two sets of data transfer speed: Fast Mode(400kHz) and Standard Mode(100kHz) and show a serial data setup time of 100ns and 250ns, respectively. The serial data setup time is relevant in the sense that it defines how long the data must remain stable before it is sampled. This is the data read and sent by the MCU as I2C communication uses the SDA wire for reading and transmitting the data from master to slave and vice-versa depending on the first bit in the string.

Another interesting feature of the TCA9535 is that it includes an INT pin so that an interrupt signal can be sent from the external I/O Expander through this pin to signal the microcontroller that one or some of the pins have changed state. This allows the I/O Expander to remain as a simple slave device, simplifying the I2C bus communication protocol due to knowing when it is relevant to send the read command from the MCU to the I/O Expander. The I2C protocol and how the IC behaves in said communication is very well documented in the component's datasheet and provided to be very useful to program said communication protocol with the MCU.

So, the total amount of pins required for a successful communication between I/O Expander and MCU is three (SDA, SCL and INT). Given there are two I/O Expanders in the design (one for positive detection and one for negative detection), means the need of the MCU to include two I2C and 6 pins reserved for these devices.

Referencing section 2.3.2, the MAX11198 featured SPI-compatible serial interface with a dual DOUT bus due to it's simultaneous sampling ADCs. SPI communication protocols need of 4 signal wires and is also intended for short-distance communication between devices. Its 4 independent signals allow for more complex Master-Slave structures that can all run on the same Source Clock. However, this SPI communication only needs for one way data communication for the converted signals by the ADC synchronized by the same SCK line. For this, we can setup two SPI blocks so that one of the interfaces is setup as MISO and the other one as MOSI. Both SPI blocks will share the same SCK line, so both SPI blocks will be synchronized as desired. The SCK will have to run at 50 MHz, as stated in the datasheet. The ADC will then require 2 SPI blocks for a total of 4 pins reserved to this communication.

In relation to the ADC, it is relevant to point out that the ADC enables for an external way to start the conversion by means of the CNVST pin. Given that the required time length between each conversion is known, the MCO feature from the MCU can be used to clock a signal of 2Mhz at the CONVST pin. This allows to automate the ADC's conversion-rate and, thus, automate the signal's measurements. This feature would only require one pin of the MCU that allows for an MCO configuration.

It's always nice to have different ways of accessing various debugging stages. Given USART communication can just require two signal wires, one for transmitting (TX) and another for receiving (RX), the MCU may include at least one USART interface. The speed of USART protocols is standarized to 1Mbit/s. So, two pins from the MCU should be reserved for this interface.

It's intended for the power supply to be from a USB connection. So, given the intention to implement a USB port for supply, it also makes it a great interface with communication from a PC, which is the intended in the goals of the design of the Testbed. FS_USB interfaces need at least 2 GPIOs and the speed of the interface will be 12Mbps(Full-Speed) so that a USB isolator IC can be used.

Lastly, an external clock source provided by an oscillator can come in handy later down the line due to the need of specific frequency values for the MCO. The oscillator allows for a second clock source and needs to connect to the MCU. For this design, given the MCO's desired clock speed to be 2Mhz, an oscillator that provides 8Mhz would suffice. The MCU will require 1 GPIO as input for the external oscillator.

In addition to all of the above, two more GPIOs would be needed for the pulse detection signals originated by the comparators.

Memory Buffer Requirements

In relation to the memory requirements, this will be mostly limited by ADC's sampling rate. Given that the MAX11198 has a conversion time period of 500ns, it does 2 Million samples in one second. Due to it is a 16 bit dual ADC, each conversion period needs up to 32bit for continuous streaming. This translates to 8MB of data per second.

The actual maximum amount of memory to transmit in each pulse will be stated by the estimated longest pulse length, which can be up to 1.2ms. So, given the ADC would allow for 2Mega samples in one conversion and that the pulse cannot be longer than 1.2ms, that makes for 2.4k samples for one measurement. Since it is a dual 2 Byte ADC (16 bit), that rounds up to a total of 9.6kB of memory required per measurement. This data transmission is possible with USB1.1 (12MBit/s).

In conclusion, not much memory is needed to process each measurement for the kind of stimulation signals at hand. The timing, however, can be more of a limitation.

Requirements for Expansion

As mentioned at the beginning of this section, it is of good practice to keep expansion in mind so that the new requirements of bigger or more complex applications does not necessarily mean a whole redesign of the first approach. A redesign at higher scale usually happens when the first chosen MCU is not capable enough for the new requirements that may occur from expansion, typically related to memory space, supported communication technologies and GPIO pin count. In this implementation, the most visible expansions would come from adding more channels to the design. More channels would mean either bigger I/O expanders than 16 bits and/or more I/O Expanders. Another direct expansion could be simultaneous channel signal measurement requiring a dedicated ADC for each channel to measure.

So, in the first case of including more channels to the system, how does it impact the MCU requirements? Including more I/O Expanders does mean another dedicated I2C address meaning another I2C block. However, the I2C interfaces could be multiplexed, since the stimulation pulse will never occur at the same time.In conclusion, we can just have one extra I2C available in case of further expansion.

In the case of SPI interfaces, it wouldn't be necessarily needed either for the same reason as for the I2C block. The stimulation pulse will never happen at the same time, so the SPI interface can be multiplexed. However, additional chip-select pins would be needed as well as AND gates to create the multiplexer. Also, the channel stimulation order would have to be known before hand so that the multiplexer can switch to the next channel immediately after one stimulation pulse is done. Otherwise, two additional channels would be needed.

Lastly, the same amount of GPIOs would be needed for any additional channel since the detection must be independent from any other channel.

2.4.2. STM32F412RE

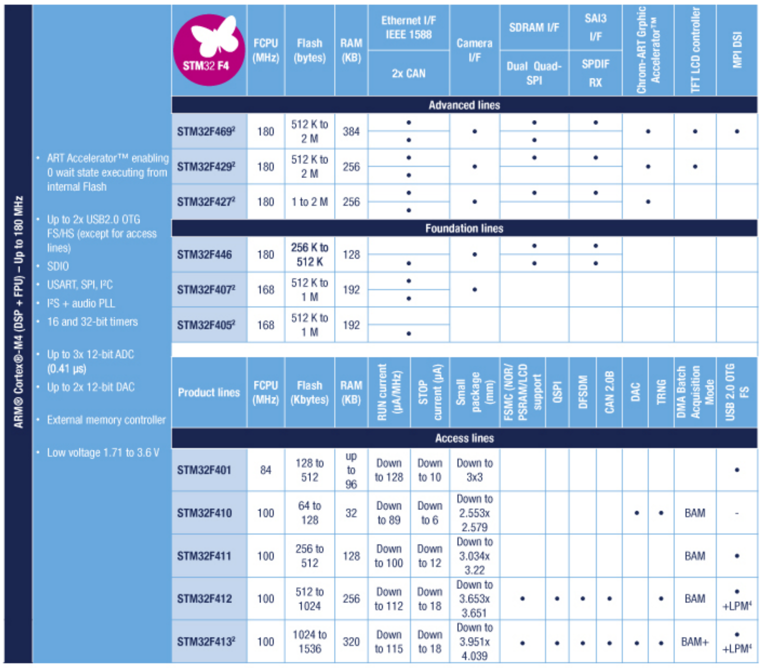
The search for a suitable MCU for the testbed was centered over the STM32F series, given that is the MCU series that the university department where this thesis was conducted that is most familiar with. Also, this familiy of Microcontrollers is provided with a lot of software support that aids with the MCU's initial configuration pin mapping and initialization, developed HAL environments, deubgging tools, etc.

As seen in figure 2.12 taken from the manufacturer's webpage[8], the manufacturer gives a simple glance of the features and capabilities of each product family, which makes the task of searching for the right MCU easier. For the Testbed, the most suitable of the STM32F4 series offered was the STM32F412, as it showcased at first glance a fast FCPU and an average size Flash and RAM. However, what was most relevant was whether if it supported the type of interfaces the design required. Table 2.1 lists the requirements by the testbed being designed in junction with the features the STM32F412RE includes.

	I/Os	I2C	SPI	USB Type	USART	Supply Voltage
Needed	19	2	2	as Device (USB1.1)	1	3.0-3.3V
STM32F412RE	50	4	5	OTG FS	4	1.7-3.6V

Table 2.1.: Features of the STM32F412RE

It can be easily seen the MCU chosen meets the requirements specified and still leaves some room for expansion, such as extra I2C and SPI interfaces and a larger number of I/O pins. A higher number of I/O pins available from the MCU allows to re purpose those not in use for debugging or as status pins.



ARM® Cortex®-M4 (DSP + FPU) – Up to 180 MHz

- ART Accelerator™ enabling 0 wait state executing from internal Flash
- Up to 2x USB2.0 OTG FS/HS (except for access lines)
- SDIO
- USART, SPI, I^2C
- I^2S + audio PLL
- 16 and 32-bit timers
- Up to 3x 12-bit ADC (0.41 µs)
- Up to 2x 12-bit DAC
- External memory controller
- Low voltage 1.71 to 3.6 V

STM32 F4	FCPU (MHz)	Flash (bytes)	RAM (KB)	Ethernet I/F IEEE 1588	2x CAN	Camera I/F	SDRAM I/F	Dual Quad-SPI	SAI3 I/F	SPDIF RX	Chrom-ART Grphic Accelerator™	TFT LCD controller	MPI DSI
Advanced lines													
STM32F469[2]	180	512 K to 2 M	384	•	•	•	•	•	•		•	•	•
STM32F429[2]	180	512 K to 2 M	256	•	•	•	•		•		•	•	
STM32F427[2]	180	1 to 2 M	256	•	•	•	•		•		•		
Foundation lines													
STM32F446	180	256 K to 512 K	128		•	•	•	•	•	•			
STM32F407[2]	168	512 K to 1 M	192	•	•	•							
STM32F405[2]	168	512 K to 1 M	192		•								

Product lines	FCPU (MHz)	Flash (Kbytes)	RAM (KB)	RUN current (µA/MHz)	STOP current (µA)	Small package (mm)	FSMC (NOR/ PSRAM/LCD support	QSPI	DFSDM	CAN 2.0B	DAC	TRNG	DMA Batch Acquisition Mode	USB 2.0 OTG FS
Access lines														
STM32F401	84	128 to 512	up to 96	Down to 128	Down to 10	Down to 3x3								•
STM32F410	100	64 to 128	32	Down to 89	Down to 6	Down to 2.553x 2.579					•	•	BAM	-
STM32F411	100	256 to 512	128	Down to 100	Down to 12	Down to 3.034x 3.22							BAM	•
STM32F412	100	512 to 1024	256	Down to 112	Down to 18	Down to 3.653x 3.651	•	•	•	•		•	BAM	• +LPM[4]
STM32F413[2]	100	1024 to 1536	320	Down to 115	Down to 18	Down to 3.951x 4.039	•	•	•	•	•	•	BAM+	• +LPM[4]

Notes:
1. 1.7 V min on specific packages
2. The same devices are also found with embedded Hardware crypto/hash
3. Serial Audio Interface
4. Link Power Management

Figure 2.12.: List of the STM32F4 series with most relevant features

2.4.3. External Oscillator

It was hinted in section 2.4.1 the need of an external clock source in order to be able to produce a 2Mhz signal for the CNVST pin of the ADC. The only requirement then for the external clock's frequency is that it has to be higher than the desired signal's frequency so that in can then be pre-scaled. So, an external oscillator of 8Mhz would suffice.

An oscillator from the G-210STF family was picked due to its small size SMD package, fair price and wide range of frequencies so that if a higher frequency were needed in the future, the oscillator can be easily changed with minimal effort.

3. Testbed: Hardware II

In this chapter we will focus in the actual design and development of the testbed's hardware given an understanding of which technologies are involved in the previous chapter. The chapter will detail the design process of the schematics and PCB layout which were designed using EAGLE CAD software, developed by Autodesk[9]. EAGLE is a powerful tool for electronics design since it allows for back-annotation between schematic and PCB layout. The full Schematic and PCB layout expanded in layers can be found in the Annex.

3.1. Analog block

Keeping the analog block as together as possible given the placement of a large ground plane will help to keep analog and digital lines from crossing each other, also aiding in reducing capacitive coupling.

The analog block is comprised of all the devices that are directly related to the STIMP and STIMN signals. The STIMP and STIMN signals represent the unaltered positive and negative input signals from the active stimulation channel after the active channel detection block. In detail, the analog block is formed by the measurement block, the stimulation load model, the measured signal conditioner circuit and the anti-alias filter, up to the analog inputs of the ADC.

3.1.1. Board layout and routing

The board layout for the analog block was arranged at the lower part of the board due to being the more extensive in component count and ensuring that the STIMP and STIMN signals do not cross the digital block. These signals correspond to the direct output from the active channel detection block, which is considered a hybrid block, meaning it is comprised of analog and digital signals and it was designed to keep separated for this reason and because it is the most likely block to get redesigned in further expansions of the testbed, since it solely depends on the number of stimulation channels and array channels to detect.

The load models were mapped next to the boundaries of the board since they allow to connect external cables to the testbed and feature physical jumpers to choose the actual model to use in the measurement. Due to an external source having an important role in this part of the testbed, it only makes sense to keep it at an accessible part of the board for easy access and more understandable, given the size of the board.

Once the position of the load models was decided, it only made sense to keep the measurement sub-block nearby to the load model, since the measurement happens at the same signals, STIMP and STIMN. Also, it features the possibility to manually choose the ground reference to be before or after R_MESS, the resistor used in the current measurement, by means of a solder bridge and the corresponding 0R resistor. This sub-block also included a pair of extra resistors routed accordingly for each measurement to adjust the equivalent resistor measurement and match the theoretical values, discussed in section 2.2.

Then, the conditioning of the measurement signal for the ADC as well as the Anti-Alias filter were designed nearby the ADC, trying to keep the bending of the analog input signals to the ADC as wide and minimal as possible. The pulse detection comparators, given they compare the analog input signal to a reference voltage but give out a digital output that routes directly to the MCU, they were included in between the two blocks, analog and digital, with the purpose of avoiding analog and digital signals crossing each other.

Lastly, the routing of the differential signal pairs of the analog signal inputs to the ADC were kept close and together and of same length. Also, the separation between differential traces was kept constant through their entire length, avoiding 90° bends and keeping the traces symmetrical. Figure 3.1 shows the designed differential signal pairs routed to the ADC in the PCB layout.

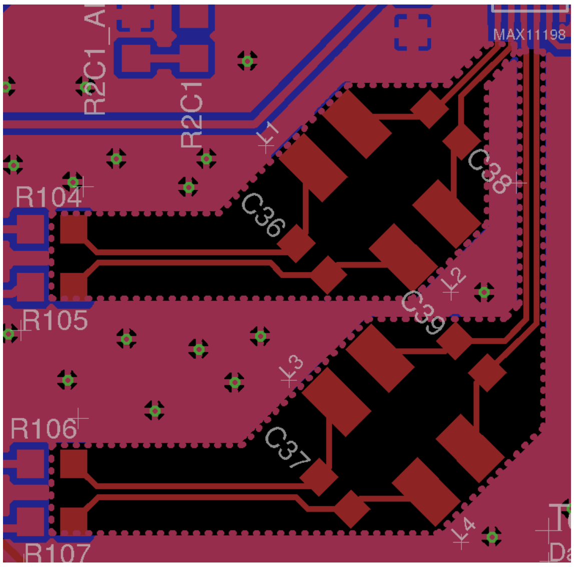


Figure 3.1.: Amplified view of the routing of the differential signal pairs to the ADC.

3.1.2. Signal conditioning for the ADC

As explained in section 2.3.3, the signal had to be conditioned in a way to fit the type of analog signal input the MAX11198 allows. For this conditioning circuit, given it had to be identical for both measurement signals, V_STIM and I_STIM, 5 OpAmps were needed for the tasks of source followers and inverter stages. The LT1819 dual OpAmp ICs were used for this purpose since their actual behavior could be simulated by means of the LTSpice SPICE simulation software by Analog Devices[10]. As described in its datasheet, the LT1819 "are dual wide bandwidth, high slew rate, low noise and distortion operational amplifiers"[11]. Since encapsulated are two OpAmp structures, in order to keep routing simple, the OpAmp used as source follower for I_MESS and the one needed as source follower for the comparators' input signal to be compared were paired together. The other two OpAmp needed for the signal conditioning stage that levels the measured signal to VREF/2 and inverts the resulting signal are also paired. The source follower for I_STIM could have been paired with the source follower at the output of the 1.25V voltage signal named VREF/2. The 2.5V VREF signal can be supplied either by the LT1790 2.5V voltage regulator or the internal available voltage reference of the MAX11198. This can be manually chosen by means of a solder jumper. All these devices mentioned need of a +5V voltage supply and the OpAmps require an equivalent -5V voltage supply as well. Figure 3.2 shows the simulated outputs for the signal conditioning circuit. In red, the voltage drop in R_MESS; in magenta, the voltage level at AIN1+; in cyan, the voltage level at AIN1-.

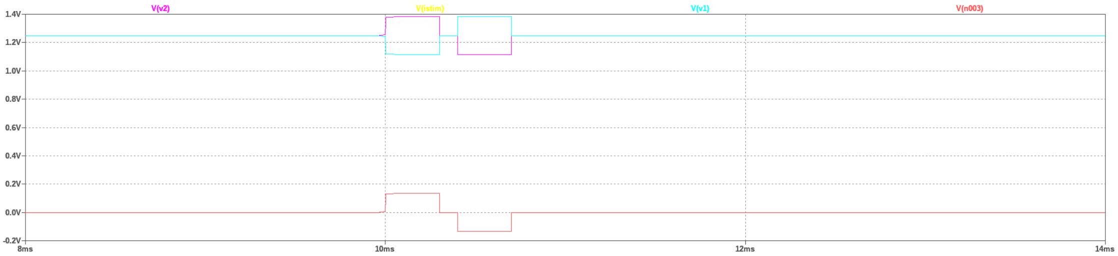


Figure 3.2.: LtSpice simulation (time-domain) for the Signal Conditioning circuit for ADC1 with a stimulation signal of I=20mA and PW=300μs.

3.2. Digital block

This block is mainly formed by the MCU and all the logical signals connected to it and the devices where these signals originate from (inputs) or have a certain function (outputs), such as the external oscillator, the two status LED branches, the output of the two comparators that are also provided with visual feedback, the ADC's digital side (DOUT, SCLK, CONVST) and the I2C buses required by the I/O Expander. The IC devices encased in this block will function properly with either a 3V or 3V3 power supply.

Two connector sets were included for debugging purposes, as well as a reset button. The first pin connector holds the required direct connection to the MCU's pins SWSCK and SWDIO required at the booting stage of the MCU to choose between normal operation or debug operation. The port follows the connector's signal order, which is as follows starting by the defined first signal: VDD, SWSCK, GND, SWDIO, GND. The second pin connector features a male pin setup (SV1) and will allow for easy connection to the debugging tools such as the logical analyzer.

Lastly, in reference to the board layout of the digital block, it is very straightforward. The MCU sits in place as the nucleus of the board, where the rest of the devices stand sort of away of the MCU so that they do no influence each other by power and heat dissipation and oriented towards the required pin they should connect to. Due to their size, the reset button and connectors sit at the border of the digital block where its easier for all the debug signals to be routed to them. Routing was kept at a certain distance of soldering pads making use of the advantage of two-sided board routing. Also, the USB signal pair trace followed the differential trace rules explained in the previous section. These signals are isolated from one plane to the other (isolated USB plane and Digital plane) by means of solder bridges.

3.3. Power Supply block

The design of the power supply was pretty straightforward and typical of an electronics project that is meant to be powered by the USB connection. It features a USB connector, an isolator IC, two DC-DC voltage converters and the necessary LDO voltage regulators to provide the respective voltage supply sources required by the devices included in the testbed's design.

3.3.1. USB power supply and signal isolation

The USB connector chosen for the design was a mini USB type-B which is capable enough to keep up with the data transfer speeds required and detailed in section 2.4.1. It was decided mini over micro because it provides a more robust physical connection with the PC's USB port given the testbed's board size.

USB connections at 12MBit/s are recommended to be isolated for security and safety reasons. This isolation allows for the USB +5V supply to be in a different reference plane as the rest of the signals of the testbed. This is done by means of a USB Digital Isolator and the DC-DC converters. As USB Digital Isolator, the ADUM4160 was chosen. It is important to understand how each pin of this IC needs to be connected so that the USB connector and Testbed stay relevant and referenced between each other without actually being directly connected to one another. As specified in the ADUM4160's datasheet, "where the isolator is powered by the USB bus voltage, 4.5 V to 5.5 V, connect VBUS1 to the USB power bus."[12]. GND1 is connected to the USB connector's ground and the node was renamed as USB_GND. This ground signal will be connected to the GND or VIN- pins of the DC-DC converters. SPU is set to VDD1 for full speed slew rate, timing, and logic conventions.

For VBUS2, however, the pin description reads: "Where the isolator is powered from a 3.3 V power supply, connect VBUS2 to VDD2 and to the external 3.3 V power supply. Bypass to GND2 is required."[12]. PIN and SPD are both connected to VDD2 as well to enable operation on power-up and full speed slew rate, timing, and logic conventions, respectively. The latter must match SPU pin setup.

USBLC6-2 provides ESD protection for the plus and minus data signals and are later renamed to USB_DP_MCU and USB_DM_MCU, respectively, passed the isolation. Lastly, SLD pin refers to the shield connection for the USB connector which is constructed of a 0R resistor connected to GND_USB.

3.3.2. The testbed's supply voltage sources

The testbed features a mix of analog and digital signal electrical circuits. Because of this reason, is typical to have two separate independent supply voltage sources. To achieve this, it is necessary two independent DC-DC voltage converters, which both will convert the +5V DC signal from the USB sub-block into a suitable DC voltage signal to later be fine-tuned by means of the LDOs regulators to match the required voltage supply sources by the testbed.

Analog block voltage supplies

For the analog signal block, it is required +5V and -5V supply sources by the Operational Amplifiers. A 5V to ±9V DC-DC converter was used. The NMH0509SC DC-DC converter was chosen for this task.

The converter's 0V pin is connected to the ground signal of the testbed and the ±9V outputs are both referenced via a capacitor to this signal as well. Two LEDs branches were included to indicate that these two supply signals are active. The next step in creating the supply sources for the analog block of the testbed was to include a voltage regulator at each output signal from the DC-DC converter to have a stable ±5V power supply. For the -5V voltage source, the MIC5270-5.0YMS ensures a stable -5V signal at its output. For the +5V voltage source, the LDK220M50R gives as output the corresponding stable +5V voltage supply. The latter features an enable pin that is connected directly to the +9V input voltage signal since it sole purpose is to enable the voltage regulation through said LDO regulator.

Digital block voltage supplies

The devices included in the digital block are such like the MCU or the ADC, which have been chosen to be very low voltage devices. And, as such, they require low voltage supplies up to 3.6V. Is typical for the digital block of electronic designs to be powered by voltage signals usually around 3V3, which is why it was chosen to implement the MEJ2S0503SC as the DC-DC converter for this signal. Analog to the ±9V supply voltage signals, an LED branch gives a visual indication whether the 3V3 voltage supply is active or not. It was discussed the necessity of including a voltage regulator for the 3V3 signal output from this DC-DC Converter given that it is already a suitable low voltage. However, it was concluded to allow the user to decide manually via a solder jumper whether to directly use this source as supply voltage for the digital voltage or to use the output signal from the voltage regulator. A dropout voltage must be ensured through the input and output terminals of said LDO voltage regulator.

Nowadays, regulators are advanced enough that can ensure very low dropout voltages like 100mV, all depending on what value is needed at the output pin of the regulator. Choosing a value of 3V as second voltage supply for the digital block ensures a reasonable amount of dropout voltage without being too restrictive or too high, and being high enough for the devices that have to be powered by said signal. Given the two different values of supply voltage for this block, the supply source signal is represented by the node SJ_VDD in the schematic and refers to the output of the solder jumper between the two supply voltage sources.

It is relevant to mention, resistors in serial with the source signals were included in junction with test-points to easily measure the actual input and output voltages of the DC-DC Converters. The same approach was implemented for the LDOs. This can prove handy at debug stages if the situation where an external source has to be used arises when the supply block fails in a certain way.

4. Software

This chapter follows the software design steps and expected function of the necessary MCU program not only for proper communication between devices in the testbed but also to allow proper data retrieval from the hardware and be able to send it to a PC interface. The software was developed in C programming language in the environment tool named System Workbench for Eclipse developed by STMicroelectronics.

4.1. Initial configuration

As first step, it was needed to create a project template for the STM32F412 with the aid of the CubeMX interface developed by STMicroelectronics. CubeMX is a useful tool to surpass the tedious repetitive microcontroller initial basic configuration such as port configuration, clock initialization and so on. Figure 4.1 shows the required pin setup for the STM32F412.

The USB interface was configured as Device Only and automatically reserved the pins related to a Device Only function for the USB_OTG_FS; PA12 and PA11, which were later renamed to USB_DP_MCU and USB_DM_MCU, respectively, to make clear which is the Plus signal and which the Minus signal. USART6 was configured in Asynchronous mode and it reserved pins PC7 and PC6 with the purposes of Receiving (RX) and Transmitting (TX), respectively.

In order to be able to load the program and debug the program in the MCU soldered to the PCB, the SYS interface was set-up as Serial Wire (SW), reserving pins PA13 as pin SWDIO and PA14 as pin SWCLK.

The SPI interfaces, SPI1 and SPI4 were set-up as Receive Only Master and Receive Only Slave. By configuring these interfaces in such a way and routing the SCK pins between each other it is possible to have a simultaneous dual-SPI between the MCU and the ADC, given that the latter had two outputs that had to be simultaneously sent to the MCU in a synchronized manner.

The I2C1 and I2C2 interfaces from the MCU were configured as I2C. Pins PB8 and PB9 would be routed to PIOE_INT and NIOE_INT, respectively, and would be configured as inputs as well as PC0 and PC1, renamed PPULSE_DETECT and NPULSE_DETECT, respectively. PC14 and PC13 were configured as outputs and would be used as status pins, renamed STATUS1 and STATUS2, respectively.

Once the pin layout is designed, it is necessary to configure pin specific parameters mostly related to time or to communication interfaces. DMA was configured for all communication protocols. The USART interface's Baud Rate was set to 115200 Bits/s with a word length of 8 bits. The USB_FS interface was set up at Full Speed (12MBit/s) with an Endpoint 0 Max Packet Size of 64 Bytes, whose only function is to provide a +5V power supply and two-serial wire communication with the MCU. The I2C were configured in Standard Mode (with a standardized 100kHz Clock Speed) and the SPI were set-up with a Data Size of 8 bit where the first bit is MSB.

Lastly, the clock configuration should be checked to ensure the necessary values for each interface is the value it is supposed to be by setting the correct prescaler values and clock sources. The full Clock configuration block diagram can be found in the Annex. For this project, the testbed required of an external oscillator of 8Mhz, which has been configured as input at PH0. This had to be configured properly as well as the correct prescaler for a proper 2MHz frequency output at pin PC9. Also, in order, to use the max frequency for the USB (48MHz), with the selected prescaler values for a max clock speed of 100MHz it was not possible for the desired USB speed with the default clock source, so the PLLI2S clock source was used instead for the USB block.

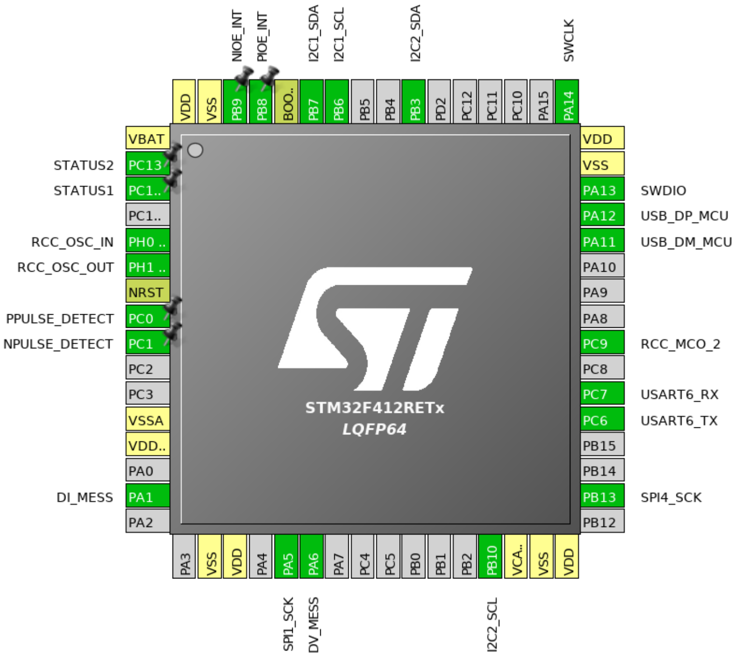


Figure 4.1.: Pin layout for the STM32F412.

The .ioc project created in CubeMX allows to then generate all the necessary file projects to then edit and to work on in the System Workbench environment. It generates a project folder that can then be imported in the System Workbench environment. In the folder Drivers, it creates the default header and source HAL related files specific to the MCU used in the project. In this case, the HAL drivers it generated correspond to the drivers required by the STM32F4 series, given that the chosen MCU for the testbed is part of this MCU family from STMicroelectronics. After, it creates the Inc and Src folders holding the respective header and source files filled already with the specified configurations needed for the project in CubeMX. In these files all the initialization functions for each interface are declared and programmed as well as the the parameters for each block as configured previously in CubeMX. It also prepares the main.c as if it were a template with the general function calls to the initialization functions that it had generated. The programmer only needs to worry about the actual protocols and functionalities specific to the task that the software should implement.

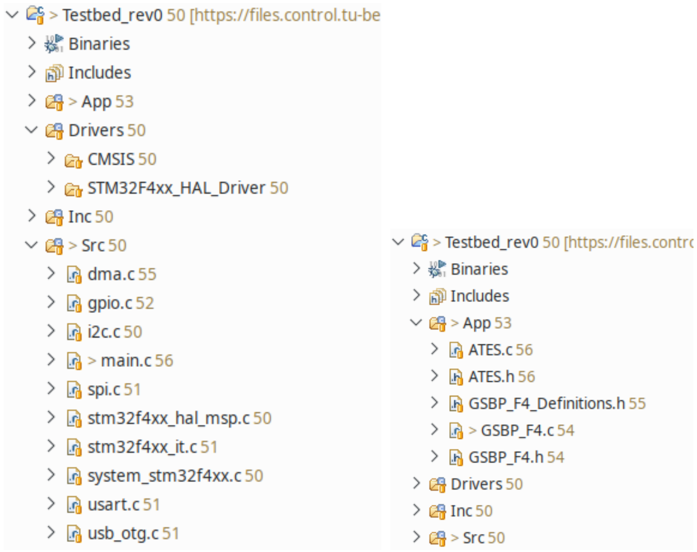
After the project was successfully generated and loaded in the System Workbench environment, a folder was created with the name App in the same workspace and was added to the project. It was intended to include any necessary external code with specific applications depending on the necessary peripheral functionalities, such as the USART and USB communication protocols with the PC and the code related to the specific tasks involved in the Testbed. By structuring the project in such a way it allows for easy implementation of new functionalities and allows for a less overloaded main program, where the latter is only formed by function calls and basic conditions, easy to understand and to work with. The header-source pair files, ATES.x and GSBP_F4.x, correspond to the testbed's specific protocols and to the communications related protocols, respectively. GSBP_F4_Definitions.h holds communication related variable definitions and function declarations that are application specific for easier localization, understanding and in-code legibility. Some examples of values compiled in this header file are such as buffer package sizes, state indexes or specific addresses involved in different communication protocols.

4.2. GSBP_F4.h and GSBP_F4.c

This section serves as an introduction to the General Serial Byte Protocol header and source files that allow for a proper communication protocol between PC and MCU, whether it were by USART or USB interface, specific to the implementation in the testbed. These files were only imported from given templates to work as an API.

4.2.1. UART communication

The UART facilitates communication through a computer's serial port using the standardized RS-232C protocol. A transmitting UART converts parallel data from a device such as a CPU into serial form so that it can then be transmitted to a receiving UART



```
v > Testbed_rev0 50 [https://files.control.tu-be
  > Binaries
  > Includes
  > > App 53
  v Drivers 50
    > CMSIS 50
    > STM32F4xx_HAL_Driver 50
  > Inc 50
  v > Src 50
    > dma.c 55
    > gpio.c 52
    > i2c.c 50
    > > main.c 56
    > spi.c 51
    > stm32f4xx_hal_msp.c 50
    > stm32f4xx_it.c 51
    > system_stm32f4xx.c 50
    > usart.c 51
    > usb_otg.c 51
```

(a) Generated files by CubeMX expanded

```
v > Testbed_rev0 50 [https://files.contr
  > Binaries
  > Includes
  v > App 53
    > ATES.c 56
    > ATES.h 56
    > GSBP_F4_Definitions.h 55
    > > GSBP_F4.c 54
    > GSBP_F4.h 54
  > Drivers 50
  > Inc 50
  > > Src 50
```

(b) User generated App folder expanded

Figure 4.2.: System Workbench C project.

by means of a physical cable. The latter then converts the serial data form into parallel data. In order to know when the data package transmitted/received starts and ends, it makes use of start and end bits to the data package. Once the package structure for GSBP is understood different functions can be created to be able to perform basic communication tasks through this connection such as read, evaluate, build and send data packages between devices. These functions were all declared in the GSBP_F4.h header file and they were defined in the GSBP_F4.c source file.

The USART interface being used in the testbed has been programmed to be used as the GSBP_Interface_2.

4.2.2. USB communication

The USB interface, also providing data transmission with a PC in a serial form, follows the UART's way of handling data packages. In that sense, the user of the program only would need to choose the kind of interface they would like to use for the serial communication with the PC, since the basic communication tasks mentioned while explaining the UART would be the same. USB and UART mostly differ in hardware form. The first always uses two cables for the transmission of data packages in a differential form, whereas the latter provides a one-way data transmission, which is why it also needs two cables, one for transmitting and another for receiving. That the data transmission through USB is differential means that when one signal (for instance P for plus) is high the other one (for instance M for minus) must be low.

The USB interface being used in the testbed has been programmed to be used as the GSBP_Interface_1.

4.3. ATES.h and ATES.c

This section serves as an introduction to the Automated Testbed for Electrical Stimulation header and source files that allow for proper operation of the testbed and a proper communication protocol between hardware devices like the I/O Expanders and the ADC with the MCU.

4.3.1. SPI

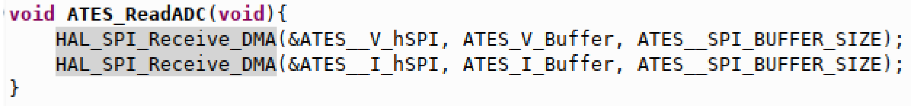
A SPI provides a serial synchronous data bus, clocked by means of the SCK signal of the interface. Any devices whose SPI is clocked to the same SCK signal will be provided with a synchronized data transfer. In the testbed, we would use two SPI interfaces from the MCU, one for each digital output from the ADC. In this sense, the SCK pins for the SPI interface were routed together as well as to the SCK pin of the ADC, ensuring proper synchronization between devices and the data transferred between each other. It was mentioned that SPI1 was configured to be used for the DV_MESS output and SPI4 would be used for DI_MESS. For easier code readability hspi1 and hspi4 were defined as ATES__V_hSPI and ATES__I_hSPI at ATES.h. The hspi symbol relates to the SPI_HandleTypeDef structure created by CubeMX for the proper use of the SPI interface in the MCU. The structure contains the configuration information for the SPI module.

The estimated buffer size of the data packages to be sent through this interface were set to 16 bits, in relation to the ADC's resolution.

Due to data transmission only being from ADC to the MCU, only a function with the purpose of reading the data and converting the received data packages was defined. Function ATES_ReadADC() makes use of the predefined function by CubeMX HAL_SPI_Receive_DMA(). This function uses as parameters which SPI to read, where to store the data read and the size of the data package in the transmission. Figure 4.3 shows the implementation of this function for the desired operation.

4.3.2. I2C

The I2C protocol is a standardized communication protocol intended for short distance communication between electronic devices based on a Master-Slave relation. Analog to



```
void ATES_ReadADC(void){
    HAL_SPI_Receive_DMA(&ATES__V_hSPI, ATES_V_Buffer, ATES__SPI_BUFFER_SIZE);
    HAL_SPI_Receive_DMA(&ATES__I_hSPI, ATES_I_Buffer, ATES__SPI_BUFFER_SIZE);
}
```

Figure 4.3.: ATES_ReadADC() as defined in ATES.c

the SPI, I2C provides for a synchronized data transfer by means of a SCK signal. The I2C interfaces were designed in such a way so that the MCU always works as the master and the I/O Expanders as slaves. This is possible due to an extra signal originating from the I/O Expander that provides an interrupt every time a change occurs in any of its ports. This signal is useful because then the MCU only needs to send a read request to the I/O Expanders whenever the INT signal is detected. By using independent I2C interfaces from the MCU for each I/O Expander, it is possible for independent read-out of the I/O Expanders in relation to the state of the INT signal.

The I2C protocol requires a device address which is standardized to be 7 bits. These were defined for both I/O Expanders in ATES.h. ATES__PIOE_ADDRESS was set to 0b0100100 and ATES__NIOE_ADDRESS was set to 0b0100110 matching the soldered jumpers in the PCB. The three jumpers at the left correspond to NIOE and the right ones correspond to PIOE, both in descending order (lowest jumper corresponds to LSB from the address bit). The Address byte is standardized in TCA9535 to have the following bit structure: 0 1 0 0 A2 A1 A0, where 0100 is fixed and Ax correspond to the user configurable address bits by hardware.

The buffer for the I2C protocol is set as an 8bit array structure, so the buffer size is set to the number of ports for one I/O Expander. In the testbed's case the TCA9535 feature two ports.

Due to the data packages from the I2C communication including the independent high or low level of the pin connected to the logical output of the channel detection block, it is relevant to store them once read as the same structure and be later on sent to the PC keeping the same structure and differentiated in ports for easier access to each individual value.

The defined functions with the task of reading from the I2C interfaces are declared at ATES.h and defined at ATES.c by the names of ATES_ReadPIOE() and ATES_ReadNIOE(), for the PIOE and NIOE, respectively. The functions are symmetrical and both make use of the predefined CubeMX functions HAL_I2C_Master_Transmit_DMA() and HAL_I2C_Master_Receive_DMA(). Even if the task is just of a read nature, as the protocol is defined, the master must first send a certain data sequence to establish the I2C Read Protocol. The protocol is as follows: The master (MCU) must send a start condition (in this case, when INT activates, which is when the signal is Low, and no other routine is active in this interface) through the SDA signal of the interface with the

R/W bit set to 0 and the register address of which port needs to be read (in this case, both ports can be sequentially read in each interrupt iteration). The master must then wait for the ACK from the slave. Once the ACK is received, the master should then send the start condition again but with bit R/W set to 1 and no command byte. The transmission is terminated when the LSB from the data byte is received. No ACK is sent rather the stop condition, P, by the MCU. Figure 4.4 shows the data package structure that must be sent by the SDA signal.

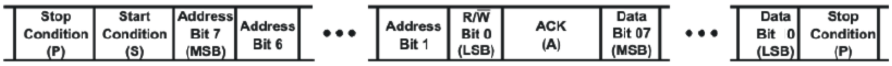
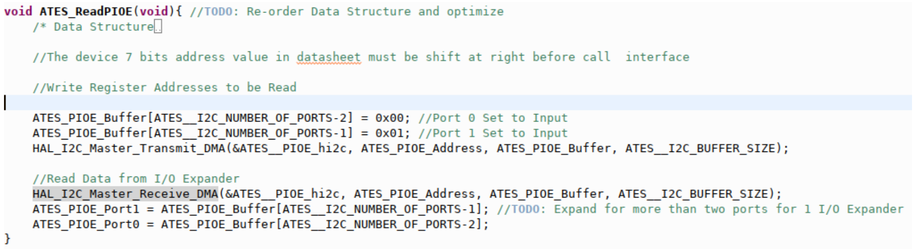


Figure 4.4.: SDA structure for the TCA9535 example

The hi2c symbol relates to the I2C_HandleTypeDef structure created by CubeMX for the proper use of the I2C interface in the MCU. The structure contains the configuration information for the I2C module. For readability, hi2c1 was defined as ATES__PIOE_hi2c and hi2c2 was defined as ATES__NIOE_hi2c. These are then used as pointers for the required parameters of the functions HAL_I2C_Master_Receive_DMA() and HAL_I2C_Master_Transmit_DM In addition to the pointers, these functions need as parameters the target device address, a data pointer to data buffer and the size amount of the data to be transferred. Figure 4.5 shows the I2C read function following the required I2C protocol required by the TCA9535. ATES_ReadNIOE() is identical to Read_PIOE() but for the values related to NIOE. Due to the fact of the Register Address being 7 bit, it must be shifted to the right prior to calling the read function.



```
void ATES_ReadPIOE(void){ //TODO: Re-order Data Structure and optimize
    /* Data Structure

    //The device 7 bits address value in datasheet must be shift at right before call  interface

    //Write Register Addresses to be Read

    ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-2] = 0x00; //Port 0 Set to Input
    ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-1] = 0x01; //Port 1 Set to Input
    HAL_I2C_Master_Transmit_DMA(&ATES__PIOE_hi2c, ATES_PIOE_Address, ATES_PIOE_Buffer, ATES__I2C_BUFFER_SIZE);

    //Read Data from I/O Expander
    HAL_I2C_Master_Receive_DMA(&ATES__PIOE_hi2c, ATES_PIOE_Address, ATES_PIOE_Buffer, ATES__I2C_BUFFER_SIZE);
    ATES_PIOE_Port1 = ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-1]; //TODO: Expand for more than two ports for 1 I/O Expander
    ATES_PIOE_Port0 = ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-2];
}
```

Figure 4.5.: ATES_ReadPIOE() as defined in ATES.c

4.4. main.c

Any MCU program is required to include a core source file and a core main() function. They don't necessarily need to be named as such but it has become a standard name so that anyone can understand where the core of the program is located.

4.4.1. Peripherals initialization

The main() function located at main.c doesn't explicitly need to do anything but it usually includes all the function calls to the initialization functions of all the blocks used in the project in a sequential manner. The CubeMX software tool generates a main.c source file following said structure and including all of the initialization function calls of all the technologies configured for the MCU where the program will be loaded. It then provides some gaps in the code so that the user can include their specific code related to the task at hand. In this gap, the initialization function call for the GSBP, GSBP_Init() was called. Then, the necessary communication interfaces to communicate with the PC needed to be enabled, GSBP_Interface_2 for an UART communication or GSBP_Interface_1 for a USB communication. Lastly, the proper code sequence allows to start the communication with the PC.

The following while loop is known as an infinite loop. The condition "1" is always met so whatever code is inside the while loop it is repeated indefinitely. The normal operations and tasks of the testbed are called here, to be repeated until power supply ceases in the testbed and, thus, in the MCU.

4.4.2. PC Communication

The first task programmed in the infinite loop is checking for new packages every certain amount of time. This certain amount of time can be easily changed by modifying the defined GSBP_CONFIG__CALLBACK_PERIOD_IN_MS which is now set to be 20ms. This task is done by calling the function GSBP_EvaluatePackage() defined in GSBP_F4.c which uses as parameter a pointer to the interface of interest used in the communication with the PC, which can be either UART or USB. The callback period is ensured by comparing a value where in each iteration it is added the respective callback period with the actual timing of the microcontroller, which said value can be known by calling the HAL_GetTick(), intrinsic to the MCU's HAL configured by CubeMX when generating the code project.

The declared variable DoMeasurements is used as condition when it is necessary to send data packages from the MCU to the PC. The task of sending the data package of interest is done by calling the function GSBP_SendPackage() defined in GSBP_F4.c. The function takes as parameters a pointer to the communication interface to be used, and a pointer to the actual package to be sent, GSBP_Handles. The pointer is declared as a package_t structure, declared in GSBP_F4_Definitions.h and it is set as GSBP_Handles.ACK when the communication is a transmission (from MCU to PC) and is set as GSBP_Handles.CMD when the communication is a received package.

4.4.3. Channel Detection

The channel detection is based on the I2C communication protocol to send the data packages from the I/O Expanders to the MCU. It has been mentioned the existence of the INT signal so that the TCA9535 always work as slaves throughout the I2C communication. The INT output from the I/O Expanders activate (active at low state) when any input state differs from the corresponding Input Port Register state. In other words, when the channel detection signals change from high to low or vice-versa. It should be intended to call the ReadI2C() function every time the INT output is activated, so long the previous read iteration at said interface had already finished. The SDA structure for the I2C Read protocol has been detailed in section 4.3.2 and it is taken care by the called functions ATES_Read_PIOE() and ATES_Read_NIOE(). The INT signals that should be used as conditions to trigger each read protocol have been renamed as PIOE_INT and NIOE_INT, respectively. This condition is done in the while loop by means of an if loop, due to the fact that the INT signals are simple input signals to the MCU.

4.4.4. Measurements

The measurement itself does not directly depend on the MCU. The digitalized measurements, DV_MESS and DI_MESS, however, are of interest for the MCU to read, store and send to the PC via UART or USB. The communication from the ADC and the MCU was designed to be SPI, where the read-out of both measurement channels would be synchronized and simultaneous. In order to take advantage of the maximum conversion speed allowed by the ADC and to be able to fit the most samples read-out in the time frames available, timing is essential. The ADC needs of an external signal to trigger the conversion of the differential analog signals routed to it. An external conversion clock signal was designed as output from the MCU for the sole purpose to automate this conversion rate, since it is known when the conversion needs to happen. The frequency of this signal was calculated to be 2MHz, in relation to the sample rate (2MSPS, f = 1/SR = 2MHz). The 2Mhz signal is generated at pin PC9 and its source is the 8MHz external oscillator.

The datasheet of the ADC thoroughly explains its conversion stage and its timing requirements. It hints that the read-out from the ADCs must start at the rising edge of CNVST. That's to say every 500ns, and should be completed before the following SAR conversion phase. Given that the SAR conversion phase is as long as the sample time, t_{12}, 400ns, and that the read-out happens during the tracking and sampling of the next data string, the read-out sequence should be completed before the 500ns.

5. Build and Testing

Before sending the designed PCB to print, it was checked with the manufacturer's DRC (Design Rule Check) file they provide in order to ensure the design is even printable with their machinery. They provide different Design Rules depending on the kind of print the costumer wishes to produce. In the testbed's case, the Multi-CB EAGLE Design Rules for 2 layers standard (w/o surcharge) was used. It features 10μm tracks, 0.2mm drills, ca. 35μm final copper (18μm start copper). A picture of the final build of the testbed for the evaluation topics discussed in this chapter can be found in the Annex.

5.1. Power Supply

The first block to be built and soldered to the PCB was the power supply. The reason behind starting with this block is because it is important to ensure a stable power source(s). This way, when debugging the other blocks which feature a certain functionality, a bad power supply can be ruled out as a cause of any issues that may occur. Also, using the actual designed power supply of the testbed at the testing stage is an advantage as it ensures the power source will work in normal operating conditions for that block as well as the optimized values of voltage source dependent components.

5.1.1. Evaluation

In order to evaluate the working operation of the DC-DC converters, their outputs were measured to ensure they approximately matched the theoretical values. To facilitate this task, testpoints were added in the PCB layout in order to allow an easy way of measuring said values.

First of all, the USB isolator IC and mini-USB connector were soldered to the board and, by means of a mini-USB cable, the board was connected to a PC to measure the actual +5V_USB signal that works as input for the DC-DC converters. Right after ensuring a proper 5V signal to use as input for the DC-DC converters, the latter were soldered to the board. A load must be connected to each output of these DC-DC converters respectively so that they properly output their ideal voltage values, which are to be 3V3 by the MEJ2S0503SC and ±9V by the NMH0509SC. Ceramic loads of 1k8 were used for this purpose and were temporarily soldered to the outputs of the DC-DC converters once they were soldered to the board. The measurement through these loads are shown and compared in table 5.1.

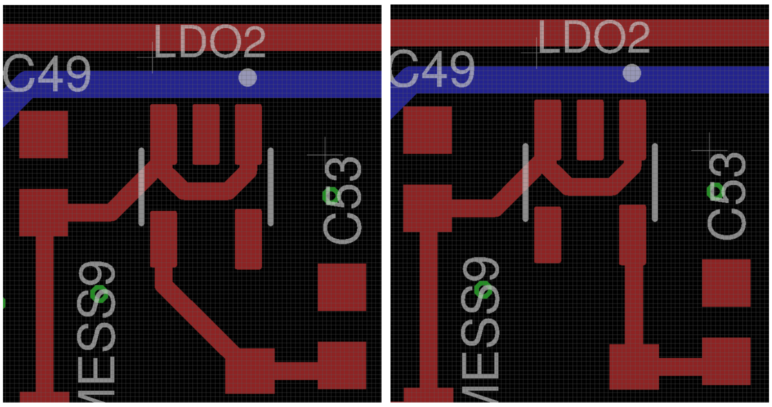
	MEJ2S0503SC	NMH0509SC	NMH0509SC	LDO2	LDO1	LDO3
Theoretical	+3V3	+9V	-9V	+3V0	+5V	-5V
Measured	+3.50V	+9.38V	-9.27V	+3.04V	+5.07V	-4.97V

Table 5.1.: Comparison between theoretical and measured output voltages from the DC-DC converters and LDOs

Once the necessary voltage conversions from the 5V voltage supply from the USB connection were tested, the LDO voltage regulators were soldered to the board. The measurement of the output voltage levels from these ICs is also available through test-points. The measurements that were conducted are shown also in table 5.1.

5.1.2. Issues

While soldering the LDO2, it was noticed the device layout was wrongly designed. Pins 3 and 5 were interchanged where pin 5 corresponds to the actual output of the LDO voltage regulator and pin 3 is a NC pin. The wrong design of the device layout does not have a major impact in the overall operation of the power supply block or the testbed meaning by major impact that it makes the board not usable or any issue of that sort. This issue means a 3.0V as VDD for the actual PCB board that was printed will not be available. However, the 3V3 voltage level supplied directly by the DC-DC converter can be used instead. This issue has been corrected in the actual EAGLE board layout by fixing the device's pin layout and updating said device in the virtual board layout. Any further PCB printings will not include this issue. Figure 5.1 shows the PCB routing before and after the the board design was updated with the fixed SMD layout.



(a) Before fix. (b) With correct SMD design.

Figure 5.1.: LDO2 in board design, before and after fix.

The package's size of the MEJ2S0503SC and the NMH0509SC were not correctly taken into account when designing their respective board layouts. Luckily, enough space was left when placing them in the PCB layout that this minor consideration did not have any effect over the build. R_MESS6 however, had to be soldered before including the MEJ2S0503SC due to the converter's size. It would be of interest to tweak the layout to leave more space between converters and so that R_MESS6 is not buried under the converter.

5.2. Active Channel Detection

Once the power supply was put into place and a stable supply voltage for the rest of the devices part of the testbed was ensured, the next block to be constructed was the active channel detection. Being the first block through which the input signal passes through it only made sense to start from there on. By following the input signal's route and trajectory throughout the testbed, we can easily debug each stage of said route so that issues found in each block can be cataloged and studied accordingly and to understand whether an issue in one block will affect or have an effect in the subsequent designed blocks.

As stated in the introduction to this chapter, not all 32 channels of the testbed need to be built. In this block, this extrapolates to not needing to build the 16 independent active channel detection diode branches. By not building all channels, time and resources are saved and more time can be focused on debugging and adding to the software. Only the four main channels which are the ones that feature the actual measurement were fully built. Testing, however was only conducted with one stimulator device at the same time, as in a normal application of said testbed, only one stimulation input channel could be active simultaneously. The other 12 channels correspond to the detection of any active electrode arrays connected to the stimulation. These channels share the same detection design as the 4 available main stimulation channels, which is why further testing and the actual build of these detection circuits for these 12 channels is not necessary for a proof of concept, as they work entirely the same as for the tested stimulation channel.

The active channel detection block features a green LED (negative stimulation signal) and a red LED (positive stimulation singal) in parallel and reverse direction to each other to provide visual feedback of the active channel during the stimulation, a 51Ω resistor in series with the optocoupler's anode pin as designed in section 2.1 and the corresponding 100kΩ PULL-UP resistor at the optocoupler's output which is routed to the TCA9535 I/O Expander accordingly. The capacitors whose sole purpose is to hold a little longer the output signal from the optocoupler were added afterwards when the block was ready to be tested, as the value of these capacitors solely depends on the implementation. The color palette mentioned for signalizing the positive and negative stimulation channel of the stimulation device is followed throughout the testbed's design where applicable.

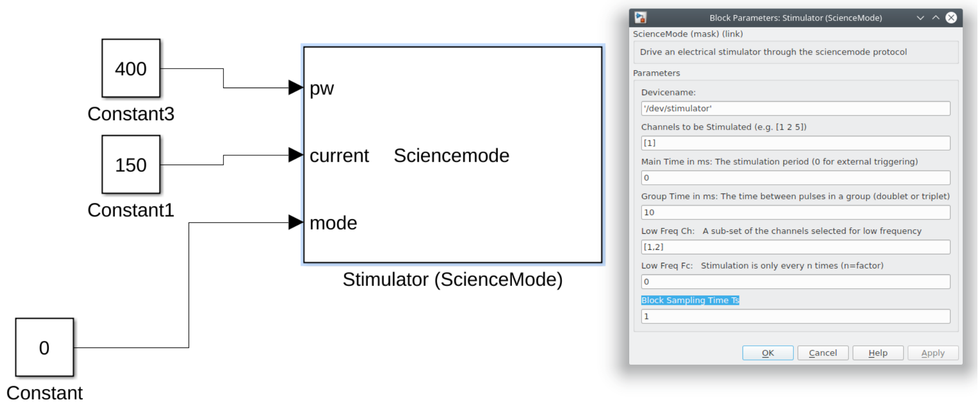
5.2.1. Evaluation

In order to evaluate and test this block, the stimulator device needs of a load for the stimulation signal to even be transmitted. In this sense, the evaluation and testing of this block was only conducted after the measurement resistors were soldered as well as, at least, the 1k load model.

In order to test the active channel detection, an actual stimulator device that would provide the necessary type of signal to detect was needed. The device used belongs to the RehaStim device family from HASOMED GmbH. Figure 5.2 shows the product image of the RehaStim1, which corresponds to the device used to generate the necessary stimulation signal. The device's generated signal is then inputed by the the channel input pins designed for this purpose by means of a connceter adapter. Then, the jumpers had to configured properly in order to allow the 1k load from the load model to be connected to the STIMP and STIMN signals accordingly. At least R_MESS from the current measurement circuit design had to be properly soldered so that the signal could travel to the load.

Figure 5.2.: Rehastim1 from HASOMED GmbH.

The RehaStim1 was used in Science Mode during the testing stage. By means of Matlab and Simulink, a block diagram could be designed in order to control the type of stimulation signal to generate. An extensivee predefined library was available for the device, so a modified project from the Stimulator_Demo.slx in the library was created that implemented just the values needed to control the desired stimulation signal, such as a pulse width value constant or the current value constant. The stimulation block allowed to configure features from the stimulation as the stimulation mode and the stimulation frequency. During testing, the mode was never changed, as mode 0 featured a bidirectional short pulse width with enough range and low frequency for easier debugging and measurement reading. Figure 5.3 shows the Simulink block diagram used to control the input stimulation signal. The constant corresponding to "pw" is in μs and the constant corresponding to the current is in mA. The mode must chosen by an positive integer number.



Block Parameters: Stimulator (ScienceMode)

ScienceMode (mask) (link)

Drive an electrical stimulator through the sciencemode protocol

Parameters

Devicename:
'/dev/stimulator'

Channels to be Stimulated (e.g. [1 2 5])
[1]

Main Time in ms: The stimulation period (0 for external triggering)
0

Group Time in ms: The time between pulses in a group (doublet or triplet)
10

Low Freq Ch: A sub-set of the channels selected for low frequency
[1,2]

Low Freq Fc: Stimulation is only every n times (n=factor)
0

Block Sampling Time Ts
1

Figure 5.3.: Diagram Block and parameter configuration from Science Mode block.

5.2.2. Issues

If a proper path was possible to the connected load from the stimulation input, the stimulation signal would be generated. If not, the stimulator device would blink red indicating an error has occurred. The most common errors encountered while testing this block came from either an unconnected load or the the stimulation program not being uploaded from the PC to the stimulator.

5.3. Measurement block and ADC

The first component to be soldered to the board was the ADC. Given its relative very small size, it was the only component soldered by means of a heat gun. Once the ADC was left into place, the rest of the blocks followed as explained.

Given the necessity of at least R_MESS to be soldered to the board since it lays right through the STIMP signal and without it the signal would not be able to travel to the load as it would provide for an open circuit, this resistor, the 0R resistor for referencing to the board's ground the stimulation signal and the resistors related to the voltage divider for measuring the stimulation voltage were firstly soldered to the board. A proper soldering connection was ensured during testing of the active channel detection block.

Before building the signal conditioning circuit for the ADC, the LT1790 was soldered to the board and tested to ensure it provided a reasonable voltage reference of 2.5V essential for the operation of this block. OpAmps were then carefully soldered due to their small size as well as the comparators. Only when the block was ready to be tested is when the jumper was solder-bridged accordingly.

5.3.1. Evaluation

In order to evaluate the proper working condition of the block, the testing and debugging stage was done following the signal's path, that's to say, ensuring the signal goes through the corresponding transitions and conversions accordingly. These measurements were mainly done with an oscilloscope and for a 150mA, 400μs signal stimulation at 1Hz for easier visual feedback and readable measurements, given that the stimulation voltage and current measurements have been designed to be up to 1.25V, matching the maximum stimulation ratings.

The measured waveforms with the oscilloscope at the measurement nodes (I_MESS, V_STIM) matched the original signal although both showed a small offset voltage. This V_{os} can be easily adjusted either further down the line or with the available adjustment resistors in the board layout. The final optimized measurement values wouldn't be chosen until the full board was proven to work as it should. The measured voltage at these nodes approximately matched their theoretical values given the offset. Both voltages showed peaks of around 1V, given the offset and not adjusted resistor values. The measurements have been compiled in table 5.2.

In reference to the comparators, the resistors need to be adjusted depending on the voltage supply level used as VDD. For the proof of concept, once measuring the real voltage level given by the MEJ2S0503SC, the best resistor values have been proven to be 8k2 and 5k1 for R1_COMP and R2_COMP, respectively. However, further optimization may be required so that the voltage level generated as comparison reference matches the elevated reference level of the actual signal to compare, which ideally is set to be 1.25V. The output of the comparator could only then be thoroughly evaluated once this value had been correctly set up.

	V_STIM(|Pk|)	I_MESS(|Pk|)	LDO4	VREF/2	V-$_{\text{COMP1}}$	V+$_{\text{COMP2}}$
Theoretical	1V	1V	2V5	1V25	1V25	1V25
Measured	880mV (V_{os}=20mV)	920mV (V_{os}=12mV)	2.46V	1.26V	1.24V	1.24V

Table 5.2.: Comparison between ideal and measured voltages from nodes of interest.

Lastly, the signal conditioned for the ADC was also evaluated following the measured signals' transformations. First, it was checked the source follower worked. Then, the V_REF/2 voltage levels as well as the following voltage divider of said signal and, lastly, the final conditioned signal at the first anti-alias filter capacitor were measured. The waveform checked out for the conditioned signal of V_STIM, as shown in Figure 5.4. The waveform corresponding to the conditioned signal of I_MESS, however, did not correspond to what it was supposed to look like at the entrance to the anti-alias filter. Only when this issue was resolved, the comparators could be properly tested and a proper pulse detection as input for the MCU could be ensured.

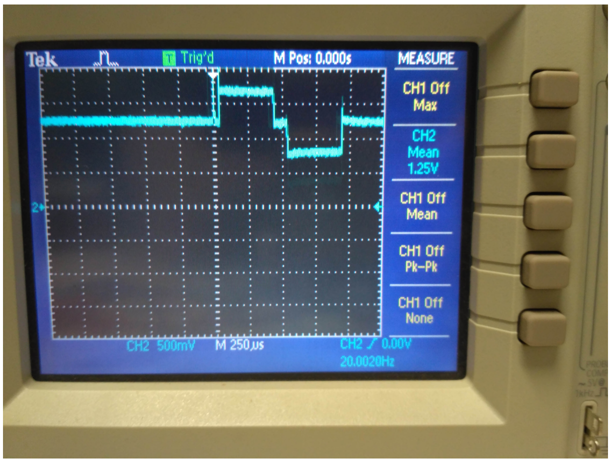


Figure 5.4.: AIN2+ node measurement with the oscilloscope for f=20Hz, I=75mA and PW=300μs.

Due to an issue with the power supply block while evaluating the signal conditioning block, further evaluation was done by applying an external voltage source to supply the testbed with the required ±5V source levels. Pins at the testpoints were added for easier coupling of the external signals to the board. The signals at the input of the anti-alias filter were measured once again without stimulation signal. Only when the supposed values at these nodes were correct as shown in the LTSpice simulation a stimulation signal was generated. The resistors to ensure a proper peak to peak symmetry between pulses for the ADC were not optimized due to the fact of this approach being a proof of concept. The resistors for the voltage dividers at the comparator stages, however, did need to be optimized to prove a proper concept application.

5.3.2. Issues

This block proved to encounter the most of the hardware related issues during the build. Given that a lot of different transformations of the same signal happen sequentially, some of the issues shown to be interconnected and once the main issue was resolved, other minor issues dependent to the prior also disappeared.

At first glance after the first trial, it could be seen an issue had appeared due to the visual feedback provided at the output of the pulse detection. The green LED, corresponding to the negative pulse detection stayed on independent of whether there was a stimulation signal active or not. This issue could be easily be attributed to the fact of a bad reference voltage at the comparison level. This issue, would have to be resolved after measuring and testing all the transformations and signal conditioning prior to the comparator stage of the block. Following the visual feedback, once the stimulation signal was generated, it could be seen the red LED to properly blink at pace of the positive stimulation pulse. The light it emitted, however, seemed very dim. The issue related to the green LED could be easily resolved by lowering the voltage level at V_+ of COMP2 by adjusting the voltage divider. The dim light emitting from the LEDs could be fixed by lowering the resistor associated to that branch to demand more current through said electrical branch.

When measuring the signal throughout the signal conditioning stages, it was found that one of the resistors related to voltage divider of VREF/2 for the V_STIM measurement signal was not correct. The voltage level at this OpAmp's pin was measured to be 1.24V when it was supposed to be half of what was measured, around 620mV. When measuring the Ohm resistance of the resistors involved in this voltage divider, it was found that one of them was not the required 1kΩ value. The resistor was changed by an actual 1kΩ resistor and this issue was resolved. It is thought the cause of the issue to be a bad soldering connection, since the issue did not reappear after replacing the resistor and soldering again.

While testing the measurement block, a power surge took place. The stimulation ended abruptly by an unknown reason and the LEDs providing feedback that the voltage supply levels are active were off. An error message from the computer's GUI used to power the board by means of the USB connection appeared stating that an error occurred where the connected device demanded more power than the USB port could handle and shut down appropriately. In order to debug the issue, the DC-DC converters were isolated from the rest of the board by desoldering the measurement resistors at the input and outputs of said DC-DC converters. First, the USB was connected and it was shown that the port provided the respective +5V and the pin VDD_1 from the ADuM4160 provided 3.3V, meaning that the isolator ICs were not harmed during the power surge and that they are not the cause of it either. Then, making sure the solder bridge for SJ_VDD was not connected, it was proceeded to reconnect the MEJ2S0503SC in order to test whether the cause of the issue could have come from the digital block or the DC-DC converter itself. When tested, the MEJ2S0503SC provided 3.6V at its output and the LED connected to its output turned red, meaning current was being administered by the converter. A 10Ω resistor was soldered at the input of the MEJ2S0503SC to measure the power consumption with and without the solder bridge connected for SJ_VDD. The measurements are compiled in table 5.3.

	V_{USB}	V_R	$I_R = V/R$	V_i	V_o
No load	4.96V	750mV	75mA	4.21V	3V
Load	4.96V	845mV	84.5mA	4.11V	2.82V

Table 5.3.: MEJ2S0503SC debugging stage after power surge.

When the same procedure was going to be performed for the NMH0509SC with a 1Ω resistor soldered at the input, after activating the power supply a column of light smoke started to appear from the soldered resistor and the power was immediately cut off. By measuring the impedance between output and ground it was proven that the device had been destroyed during the power surge. So, in order to test the rest of the components of the analog block and what might have caused the issue, an external voltage source to emulate the ±9V from the NMH0509SC was set up, and the converter was taken out of the board. First test with an external voltage source showed an excessive current draw. The first hypothesis for this current draw was thought to be that it meant there must be a malfunctioning device or bad connection that is demanding more current than what is really needed. An isolation of the LDOs from the devices that use the output from said LDOs followed in order to make sure the issue was not in the LDOs. The isolation took place by removing the measurement resistors at the output of LDO1 and LDO3. Current consumption by the LDOs showed normal values of less than 12mA each from the external voltage source. It could be concluded the LDOs were not at fault.

The outputs of the LDOs were then connected once again to the analog block for further debugging. This test showed voltage values at the output of the LDOs to be the correct ones, -4.96V for LDO3 and +5.05V for LDO1. However, the current draw was still very high, way over 500mA each. Since LDO4, the LDO related to the VREF voltage level, seemed to dissipate too much heat, the solder bridge used to connect VREF/2 to the signal conditioning block was disconnected. This test showed still a high current draw of 0.86A and LDO3 dissipating too much heat and shutting down itself, as it would output -1.36V instead of the needed -5V.

After testing each OpAmp making sure none were defect and that all voltage levels at each pin were correct, it was concluded that no devices were malfunctioning, except the dc-dc converter. Focus was shifted to the Datasheets of the components to check their current consumption and understand why such a high current was demanded from the voltage sources and why the MIC5270 (LDO3) would shut down for said current demand. The datasheet showed that each LT1819 consumes 10mA for each OpAmp. The LT1819 encapsulates to OpAmp structures, so each LT1819 theoretically demanded 20mA. This made more sense and the math would add up to the demanded 130mA by the board. After understanding that nothing was wrong with the build of the board itself and that the devices were working as they should, the datasheet for the LDOs was checked and, lastly, the datasheet of the DC-DC converter to understand why the NMH0509SC was destroyed and not the rest of the devices under such current demands.

The datasheet for the MIC5270 states that its normal operation should not exceed a current output of 100mA. If this current was exceeded, the IC is fitted with a current limiting function. When testing, the device would function over the normal operation current specification for a while until it would reach a certain temperature. This temperature was measured to be around 80°C and it was usually reached after 1 minute and 30 seconds of operation. At this point, the IC shuts off and limits the current through it by giving a lower Voltage output to prevent damage. This current was measured to be around 78mA and for the testbed and the equivalent load of the analog block this current meant a -1.74V instead of the required -5V voltage source level. The datasheet for the LDK220 stated that it could support a current demand of 200mA, explaining why it was able to withstand the current consumption of the design.

Lastly, the datasheet for the NMH0509SC mentioned the max output current values to be ±111mA. An overloaded DC-DC converter as normal operating condition was not the cause of destroying the NMH0509SC, however, given that the LDO3 already limits the current when as explained above and, even so, it was measured to only be 20mA over, making it unlikely it could have been the cause as the total power consumption was measured to be under 2W. The cause of destroying the DC-DC converter is thought to be related to a short-circuit from the oscilloscope's ground cable not being fixed to a certain pin due to a major flaw in the design stage due to the lack of a testpoint for the GND signal.

5.4. Digital Block

The build of the Digital Block was mostly centered in ensuring a proper soldering of the MCU. This component, due to its 4-sided pin layout, requires special attention to the proper placement of the pins and that they match practically perfectly with the SMD printed layout. Given its added difficulty, the MCU was the first device to solder in this block so that no other neighboring components that do not require such attention to detail like capacitors would not disturb the process. Once the MCU was properly soldered into place, the oscillator followed. Then, the switch and lastly the rest of the passive components were soldered to the board.

The developed software needs to be flashed to the testbed's microcontroller. Being the STM32F412RE part of the STMicroelectronics F4 Series, a nucleo board from a similar microcontroller (NUCLEO-F411RE) can be used for this purpose through the available serial wire connections that were included to the design. Ideally, one should get hold of the proper development kit for the microcontroller their project uses, when said project has new features to be tested and one is not familiar with the workings and behavior of said MCU.

STMicroelectronics also provides their product users with a dedicated software to connect the MCU to the PC and be able to access its memory directly. This means a program can be flashed to the MCU controlling the start Address as well as the possibility of performing a full chip erase. In this case, the software, called STM32CubeProgrammer, was used to connect the nucleo board to the PC and, with its jumpers connecting this part to the rest of the development kit, to provide a connection to the testbed's soldered MCU through the ST Link interface and the serial wire connections.

5.4.1. Evaluation

Hardware wise, the evaluation of this block was pretty straightforward. Proper working of the switch was evaluated with the multimeter to test when pressed, a connection between both pins would happen (Normally Open operation). In reference to the oscillator, the proper 8Mhz output signal could be easily measured by switching on the testbed and measuring by means of an oscilloscope the output signal of the IC. Lastly, the proper direction of the soldered LEDs can be easily tested with the multimeter by connecting the positive cable to the anode and the negative to the cathode of the LED an observe whether it emits light or not.

In order to evaluate whether the MCU actually worked once soldered to the testbed, the typical "Hello World!!" equivalent for microcontrollers was tested by toggling one of the status pins on and off. This was done by including to the developed software a function call to HAL_GPIO_TogglePin() in the infinite loop of the main.c. This function allows for an easy implementation of the required task just by setting as parameters the port where the pin to toggle is located and the exact pin to toggle, which in this case was configured as STATUS2_Pin (Green LED). A delay of half a second by the function call HAL_Delay() was implemented as well for easier visual feedback of the LED. The test was successful, meaning the MCU was properly connected and that an actual program can be flashed.

At this point, the reset button was also tested simply by keeping it pushed and looking whether the STATUS2 green LED would keep off and that it would start blinking again once the reset button was not pushed anymore. The test showed the MCU was able to reset itself and function normally independently, when only powered by the USB connection.

6. Conclusion

This final chapter provides a summary of the 5 months worth of work dedicated to the design, build and evaluation of a testbed for FES devices. It also details any unsolved issues as well as future optimization from related resolved issues and flaws. The chapter then concludes by summarizing the testbed's working conditions and limitations and by enunciating expandable work that can be done in the future.

6.1. Summary

The workload began with a short introduction and get-to-know to the technologies and working environments that would be necessary throughout the development of the thesis. These included an introduction to what is FES as well as a familiarization of the STMicroelectronics software tools and programming mentality.

Once the working environment was setup with the necessary working tools, the objective of thesis was tackled at a design level. The design thought-process and design steps follow the order as they have been explained in the previous chapters. Given the broad spectrum of the type of signal to be detected and measured, the design thought-process was focused on the detection of a short squared-pulse bipolar signal of low frequency. The different ideas were simulated with LTSpice and tested in a breadboard layout, in order to prove the final concept of active channel detection to be implemented in the testbed. The final concept was then designed in EAGLE in a modular design. The rest of the electrical design was only designed in paper and included to the overall EAGLE design block by block. Only the necessary blocks that proven to be a bit more complicated to understand the actual signal levels and behavior were simulated in order to grasp easier the concept needed to implement and to ensure the solution was achievable, such as the signal conditioning for the ADC block. During the schematic design, the layout for some devices had to be also designed when they were not provided by the manufacturer. Special care had to be taken into account by following the recommended footprint designs included in the devices' datasheets.

Only when most of the required components that would become the testbed were designed and connected in the schematic, work was shifted to the PCB layout. Easy integration between schematic and PCB layout thanks to back-annotation available as an EAGLE project allowed for the routing of the PCB layout to be an easier task. The design of the PCB layout was focused in keeping the overall size of the testbed small.

So, more time than expected took place in ensuring optimal board space and layout, also for further expansion. The designs were reviewed and checked with the program's DRC (Design Rule Check) and ERC (Electrical Rule Check) tools to ensure that the designed layout was printable. Once it checked out and all issues resolved, the board was sent to print by an external manufacturer. At this point, a BOM (Bill Of Materials) was made in order to document all the required components for the construction of the testbed and a budget, so that they could be properly ordered and ready for the time the PCB would be printed.

Due to the long delivery time for the printing of the PCB (10 days), the workload shifted to the software part at hardware level, at least ensuring a starting point and a solid structure covering the basic functionalities and tasks the MCU should perform. These were limited to communication functionalities internal to the testbed as well as for an external communication with the PC. These were not able to be properly tested, however, due to having only one month of available time left at this point in the development of the thesis and the actual build and hardware evaluation was still not completed.

After all the components and the board were delivered, the actual build of the prototype could begin. The construction of the board has been thoroughly described in chapter 3 and the evaluation of each block during construction is documented in chapter 5.

The thesis was considered terminated when the prototype had been evaluated enough to the point to assure a valid proof of concept of the hardware design that was proposed as solution for the aforementioned problem in chapter 1.

The work done can be summarized into the following work steps

- Study and familiarization with FES and the working environment.
- Problem identification and solution proposal.
- Design and basic simulation of specific blocks.
- Project documentation for development.
- Construction and build of prototype for concept testing.
- Evaluation and documentation of proof of concept.

6.2. Dissemination

This work was intended as a first approach for the automatic detection and measurement of electrical signals related to the field of FES. The built board had only the intention to prove that the designed solution was plausible and its only intent was for testing. It can still be intended as a debugging tool for related applications focused on FES and to debug any related developed software supported by the MCU used in the project, following the generated MCU configuration program as starting point.

6.3. Problems Encountered

The main issues encountered during the development of the thesis rely on the necessity of learning new working environments, tools and technical skills. However, every tool and software used was properly well documented and all issues that would be encountered related to the unfamiliarity of the working environment could be easily checked online or proper manuals. Also, very useful implementation examples were given and made software related tasks easier to develop, as they gave actual insight of how the programs behave and what can and cannot be done.

As for an electrical standpoint, the thesis does not feature any complicated theoretical technologies or concepts. However, given the technical approach, the challenge appeared when trying to visualize the end-goal of the build and the lack of experience in this sense proved to be a set-back in relation to time and also to some major design flaws that would have helped to speed-up and make easier the evaluation process. As an example, the lack of understanding of the actual kinds of connectors that could be used for certain input purposes made it hard to visualize how the channels were supposed to be tested or connected, as well as the external loads that can be connected to the testbed. A major issue related to this lack of experience in the technical side would be the lack of a testpoint for the GND signal of the testbed. This flaw contributed to the use of an extra floating cable used for the reference of the oscilloscope and that possibly caused the short-circuit given that all SMD connections in the testbed are to the open-air and it could have easily allowed for some voltage source level to be connected to ground, even for a small period of time.

In regards to design flaws, the major issue would be that of not taking into account the current values demanded and supported by some devices. This makes allusion to the issue where the LT1819 devices demanded 20mA of current each and, given there are 6 of them, the total current demand of the analog block surpassed the current the MIC5270 could support (no more than 100mA). The LDO, however is suited with an internal current limiter so normal operation does not harm the component. In regards to this flaw, the NMH0509SC also showed to only be able to provide ±111mA at its outputs, not enough from what the analog block actually demands.

6.4. Outlook

Major future work can be done to the thesis. First of all, easy fixes could be easily implemented to surpass the issues that were found during the evaluation part of the thesis. These issues are all documented and a solution has been found for them. An implementation did not follow, however, due to lack of time and because fixing these issues would not contribute to the proof of concept, such as replacing the defective NMH0509SC DC-DC converter for a new one or either replacing the MIC5270 for an LDO that can support more than 130mA of current or replacing the LT1819 ICs for other OpAmp encapsulated IC with less power consumption.

The most future work that can be developed to this thesis is focused in the software. The software needs to be properly developed and tested to ensure it works as needed for a further PC environment to be able to take the data gathered and provide the end-user with an understandable interface to easily monitor and control the FES device used in a certain FES application. A part from testing the proper configuration of the MCU program already developed and the proper function of the communication interfaces, a PC program in C++ needs to be developed so that the testbed could be controlled by means of the PC and, lastly, a Matlab based interface should be developed in order to represent the measured stimulation signal and the detected active channels so that the task of signal-monitoring is simple and efficient.

A timing diagram should be properly defined in order to better understand how the program needs to behave and when. This timing diagram should include events such as when the ADC needs to be read by the MCU or when and how often do the I/O Expanders need to be read. A timing diagram will allow to grasp how each part fits in the overall program flow.

Other future work to be implemented should be focused in optimizing certain features. A re-design of the PCB layout would be in order to ensure optimal size and space use given the total number of components needed for the task at hand. Also, some other values should be optimized depending on the final end-application. The board layout provided allows for easy adjustment of important stages susceptible to change such as the voltage divider for the voltage measurement or the measurement resistor for the stimulation signal's current. However, the values of other components such as the hold capacitance at the output of the optocoupler should be optimized accordingly to the application of the signal as well as whatever suits best the I/O Expander, given the latter to be a device that would need to be replaced when considering to expand the channels to detect. Also, the values of the anti-alias filter depend on the ADC used, and the maximum values of the conditioned signal from the measurements, if the V_{ref} were to change.

6.4.1. Limitations

The testbed was designed for a general use in FES related applications. The stimulation signals that the testbed was designed for were summarized in chapter 1 in table 1.1. The testbed was not tested for any other sort of signal (like sinusoidal signals) or over the specified operation values. In this sense, any application outside of the mentioned operation results in uncertainty in regards to the testbed's behavior as it was not documented and instances like incorrect measurements or even the destruction not only of the testbed itself but also of any external devices connected to the testbed may or may not occur. No protection circuit design was conducted as the main focus was set in proving the concept to detect and measure was possible and manageable.

The printed testbed allows for up to 4 independent stimulation channels and 12 other channels for electrode array support. Any application with the need of more than the mentioned number of channels will require a new board to be designed although the modular design allows for this task to be easily performed.

The testbed was designed to be powered just by a micro-USB cable connection to a PC. However, by use of the testpoints and manually disconnecting the relevant power components like the DC-DC converters and the LDOs by means of not soldering a resistor at R_MESSi SMD footprints, external voltage sources can be used instead. The lack of a testpoint for the GND reference of the testbed makes this task more tedious.

6.4.2. Expandability

The purpose of the testbed featuring a modular design was so that the evaluation of the designed blocks could be easily re-purposed in other FES related applications or so that the number of channels that can be detected wouldn't be a factor. An example of expandability follows.

The first approach design for the testbed featured in this thesis features 16 independent input channels, that they all then converge once more into the signal STIMP and STIMN for further measuring and load stimulation. This layout only allows for one stimulation channel to be active at once that can be measured, meaning just one stimulation device is being measured. If there were the case of a more complex FES application where more than one device is used and active at the same time, another measurement block would be needed. However, this case is not usually the case where two stimulation channels are active at the same time. The array elements, however, can be active at the same time, which is why the detection phase was designed independently for each channel. Given the design is modular, the same design can be applied and repeated to any number of channels. In this expansion example, either more I/O Expanders or bigger I/O Expanders of more pins would be needed. If more I/O Expanders are used for the expansion, given these devices will communicate with the MCU via I2C protocol, there is a limited number of address combinations possible for the I/O Expanders to share the same I2C interface from the MCU. This means that depending in the extension of the expansion, another MCU as the one featured in this thesis may be required as well that features a higher amount of interfaces.

List of Acronyms

MCU	Micro-Controller Unit
ADC	Analog to Digital Converter
API	Application Programming Interface
SPI	Serial Peripheral Interface
I2C	Inter-Intergrated Circuit
RCC	Reset and Clock Control
HSE	High-Speed External (oscillator clock)
SYSCLK	System Clock
LSB	Least-Significant Bit
MSB	Most-Significant Bit
RAM	Random Access Memory
DRAM	Dynamic-RAM
DMA	Direct Memory Access
HAL	Hardware Access Layer
GUI	Graphical User Interface
TDFN	Thin Dual Flat No-lead Plastic Package
MSps	Mega-Samples per second [u]
CMR	Common-Mode Rejection
R	Resistor
C	Capacitor
L	Inductor
IC	Integrated-Circuit
CAD	Computer-Assisted Design
PCB	Printed Circuit-Board
LDO	Low Drop-Out (voltage regulator)
ESD	Electro-Static Discharge
USB	Universal Serial Bus
SPICE	Simulation Program with Integrated Circuit Emphasis
OpAmp	Operational Amplifier
NC	Not-Connected
SMD	Surface-Mount Device
SMT	Surface-Mount Technology
pk	Peak (Voltage)
FES	Functional Electrical Stimulation
SAR	Successive Approximation Register (conversion)
MCO	Master Clock Output

Bibliography

[1] Markus Valtin. Aufbau testsystem, 2018.

[2] ON Semiconductor. *H11L1M, H11L2M, H11L3M - 6-Pin DIP Schmitt Trigger Output Optocoupler*, 2005. http://www.mouser.com/ds/2/149/H11L1M-185291.pdf.

[3] ST Microelectronics. *TS3021, TS3021A - Rail-to-Rail 1.8V high speed comparator*, 2016. https://www.st.com/en/amplifiers-and-comparators/ts3021.html.

[4] Maxim Integrated Products, Inc. *MAX11192/MAX11195/MAX11198*, 2017. https://datasheets.maximintegrated.com/en/ds/MAX11192.pdf.

[5] National Instruments. *Anti-Aliasing Filters and their usage Explained*, 2018. http://www.ni.com/white-paper/54448/en/.

[6] Sparkfun. I2c, 2018. https://learn.sparkfun.com/tutorials/i2c.

[7] Texas Instruments. *TCA9535 Low-Voltage 16-Bit I2C and SMBus Low-Power I/O Expander with Interrupt Output and Configuration Registers*, 2018. http://www.ti.com/lit/ds/symlink/tca9535.pdf.

[8] STMicroelectronics, 2018. https://www.st.com/en/microcontrollers/stm32f4-series.html?querycriteria=productId=SS1577.

[9] Autodesk Inc. *EAGLE Overview*, 2018. https://www.autodesk.com/products/eagle/overview.

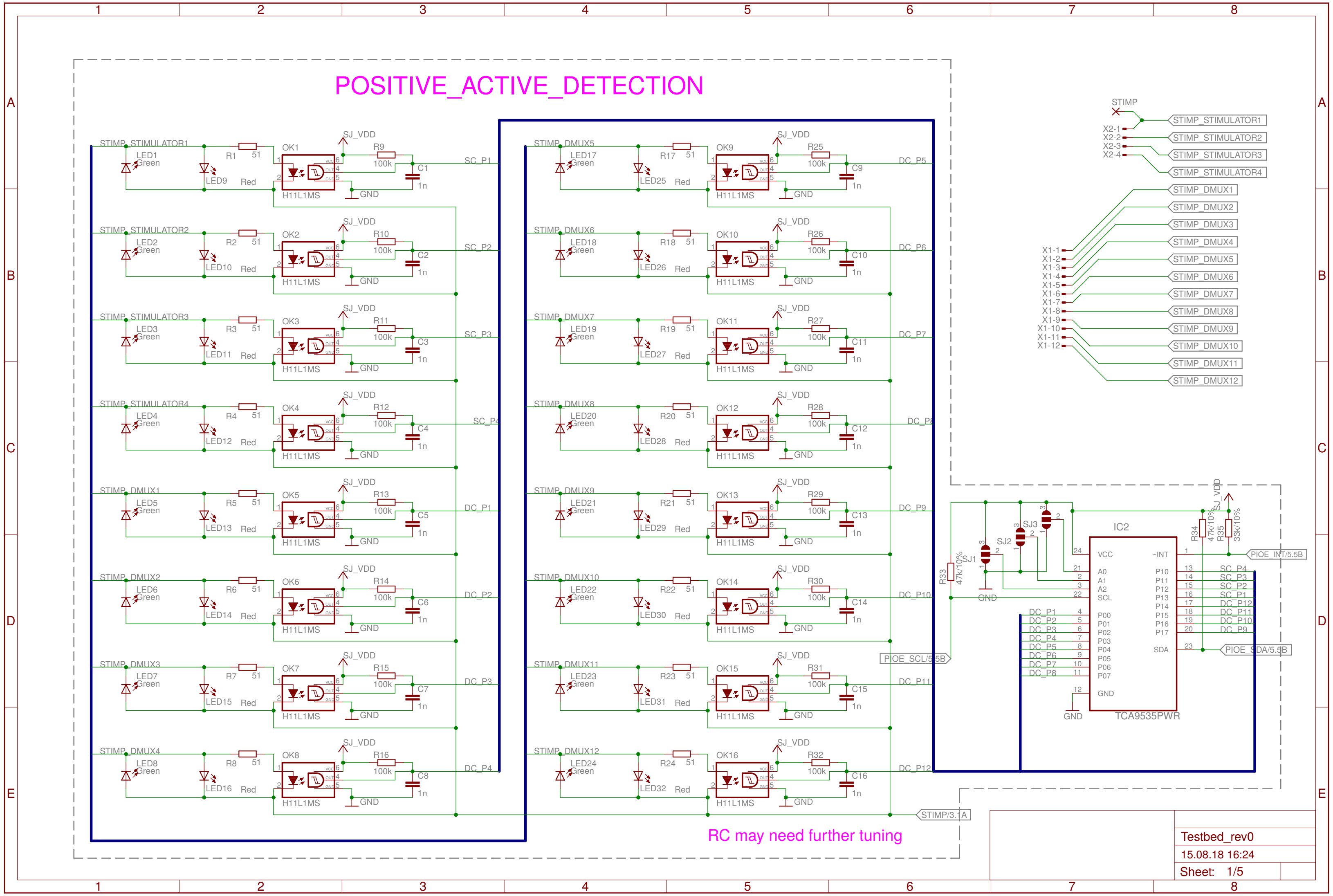
[10] *LtSpice*, 2018. http://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html.

[11] Analog Devices. *LT1818/LT1819 - 400MHz, 2500V/us, 9mA Single/Dual Operational Amplifiers*, 2002. http://www.analog.com/media/en/technical-documentation/data-sheets/18189fb.pdf.

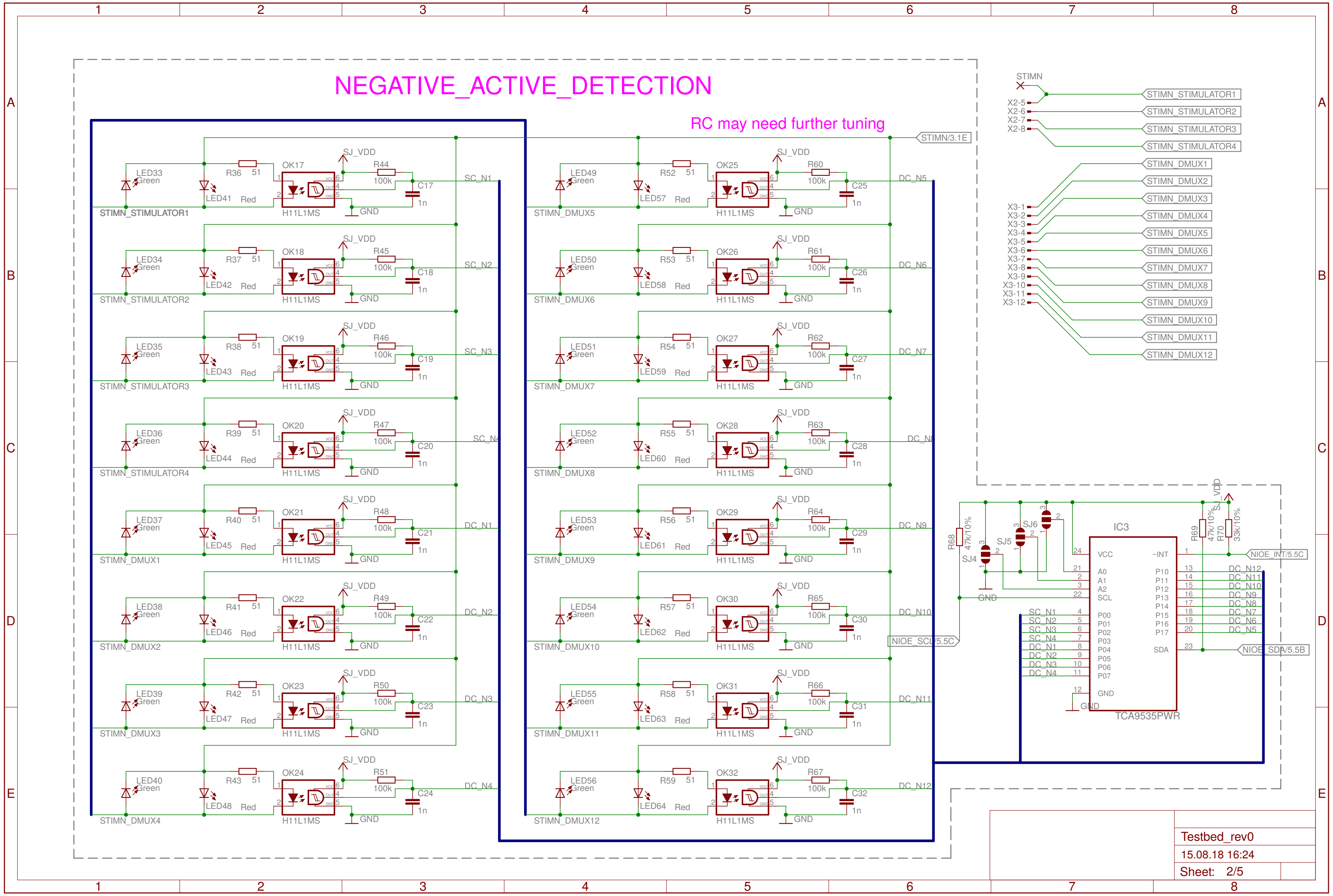
[12] Analog Devices. *Full/Low Speed 5 kV USB Digital Isolator - ADUM4160*, 2009-2012. http://www.analog.com/media/en/technical-documentation/data-sheets/ADuM4160.pdf.

Annex

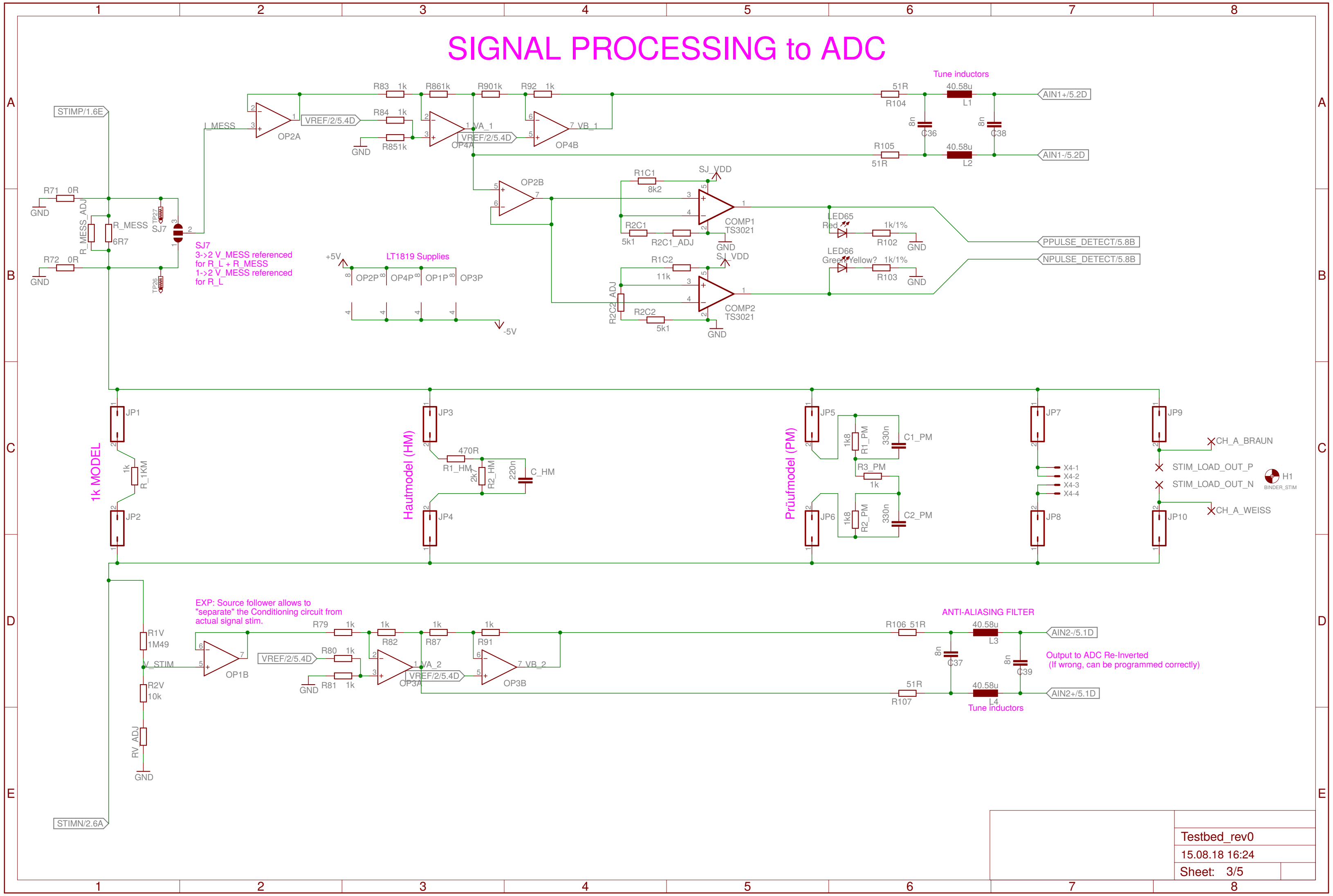
A. EAGLE Schematic



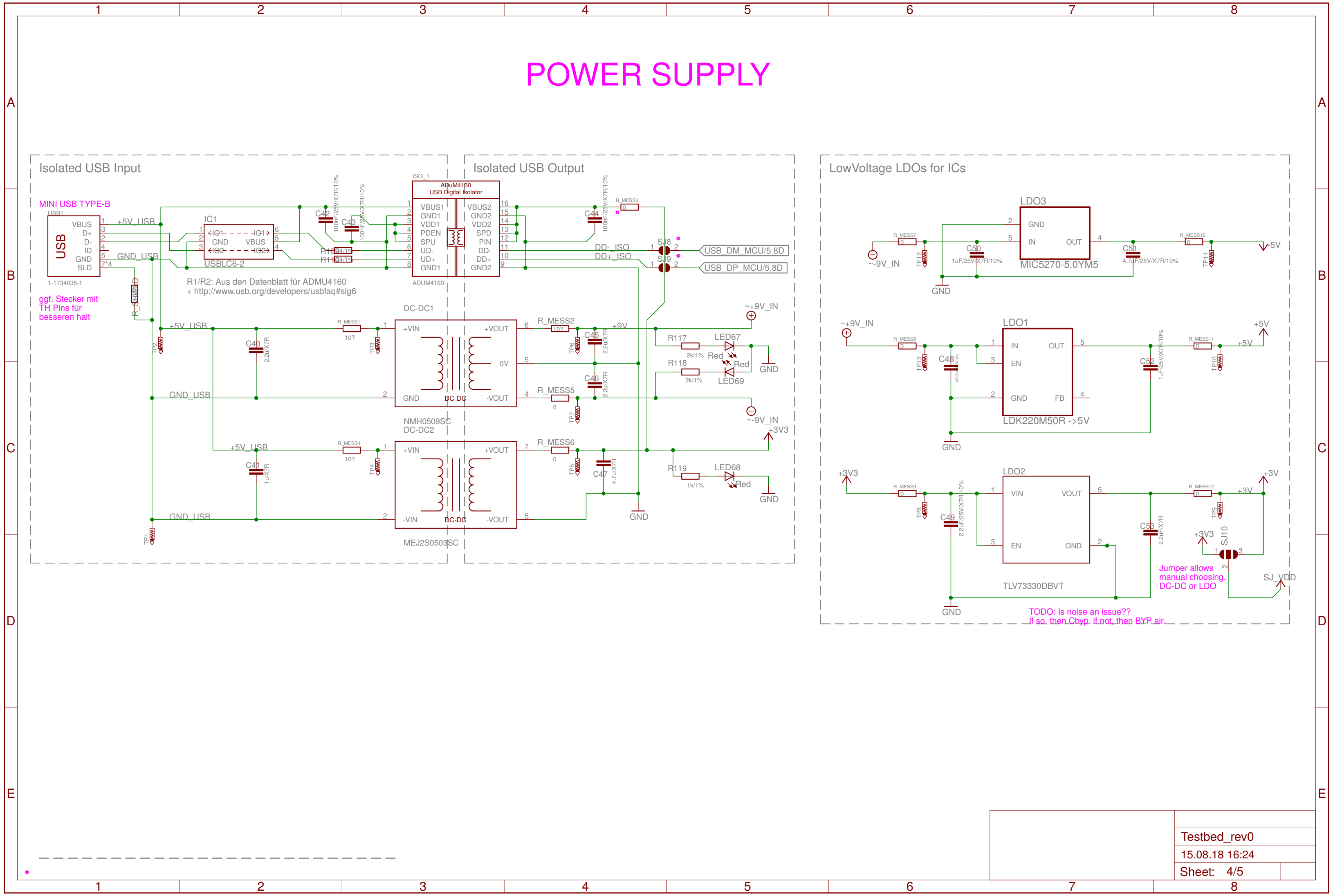
15.08.18 16:27 f=0.75 /home/russ/BA/ba_testbed/HW/EAGLE/Testbed/Testbed_rev0.sch (Sheet: 1/5)



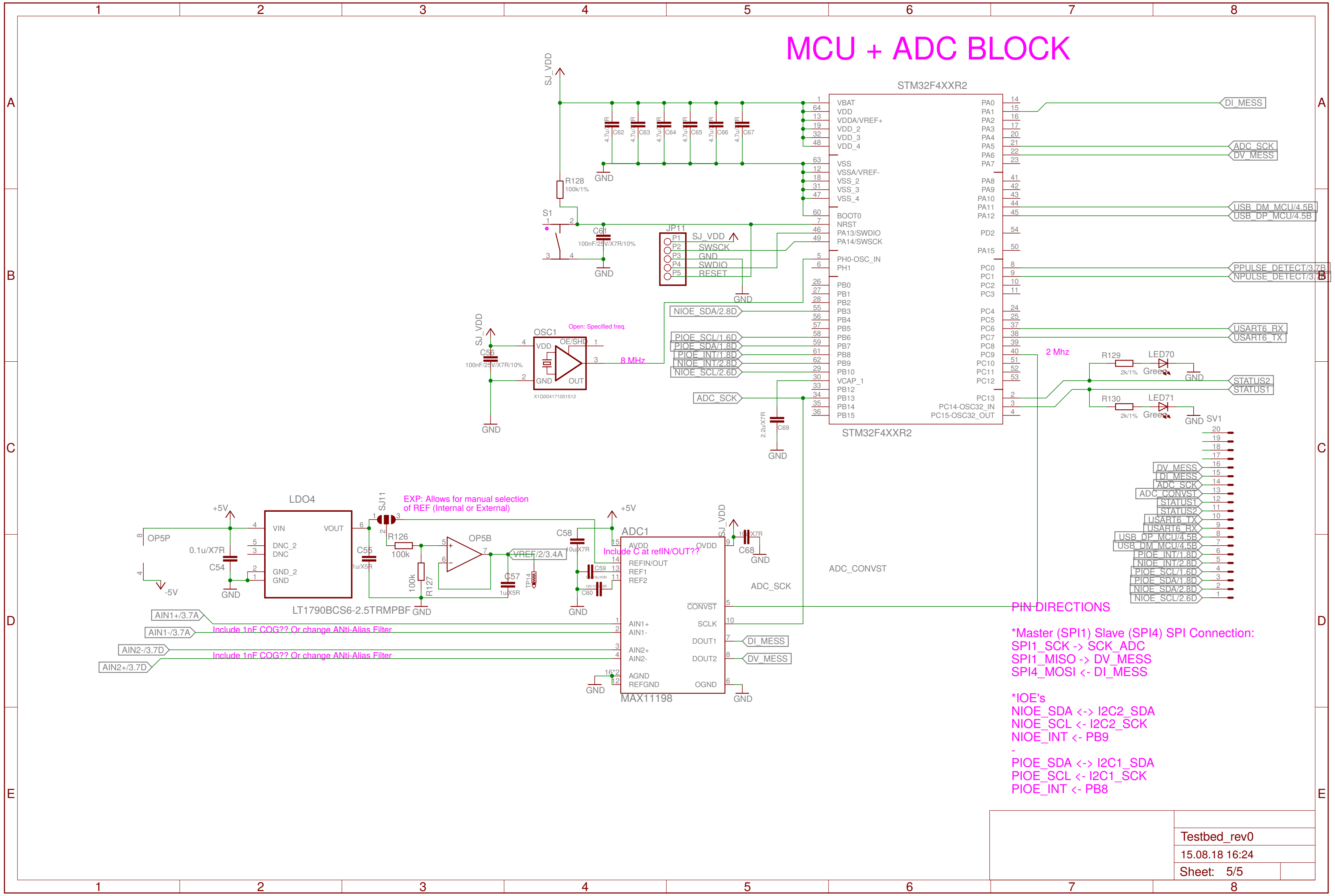
15.08.18 16:27 f=0.75 /home/russ/BA/ba_testbed/HW/EAGLE/Testbed/Testbed_rev0.sch (Sheet: 2/5)



15.08.18 16:27 f=0.75 /home/russ/BA/ba_testbed/HW/EAGLE/Testbed/Testbed_rev0.sch (Sheet: 3/5)



15.08.18 16:27 f=0.75 /home/russ/BA/ba_testbed/HW/EAGLE/Testbed/Testbed_rev0.sch (Sheet: 4/5)



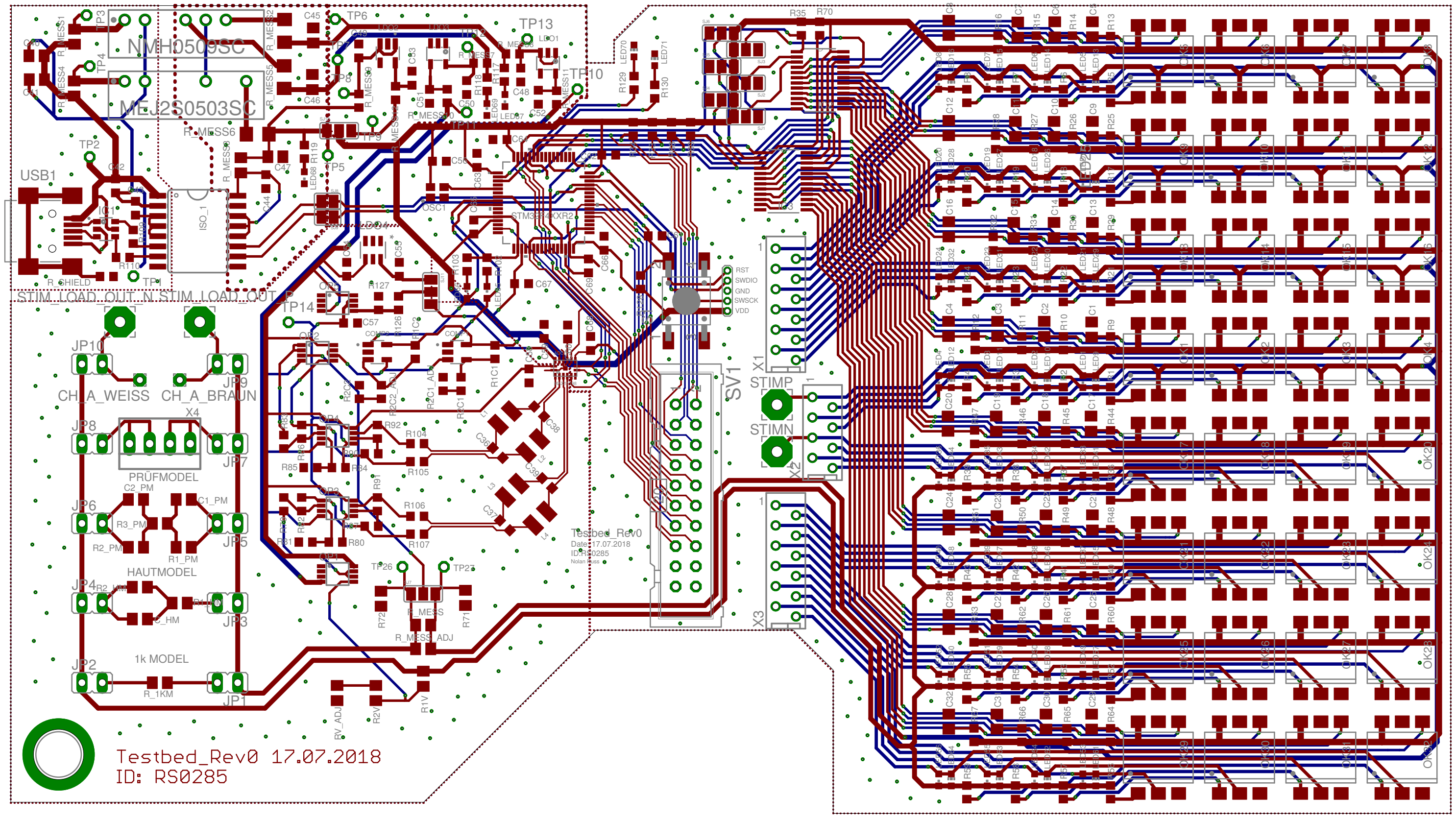
PIN DIRECTIONS

*Master (SPI1) Slave (SPI4) SPI Connection:
SPI1_SCK -> SCK_ADC
SPI1_MISO -> DV_MESS
SPI4_MOSI <- DI_MESS

*IOE's
NIOE_SDA <-> I2C2_SDA
NIOE_SCL <- I2C2_SCK
NIOE_INT <- PB9
-
PIOE_SDA <-> I2C1_SDA
PIOE_SCL <- I2C1_SCK
PIOE_INT <- PB8

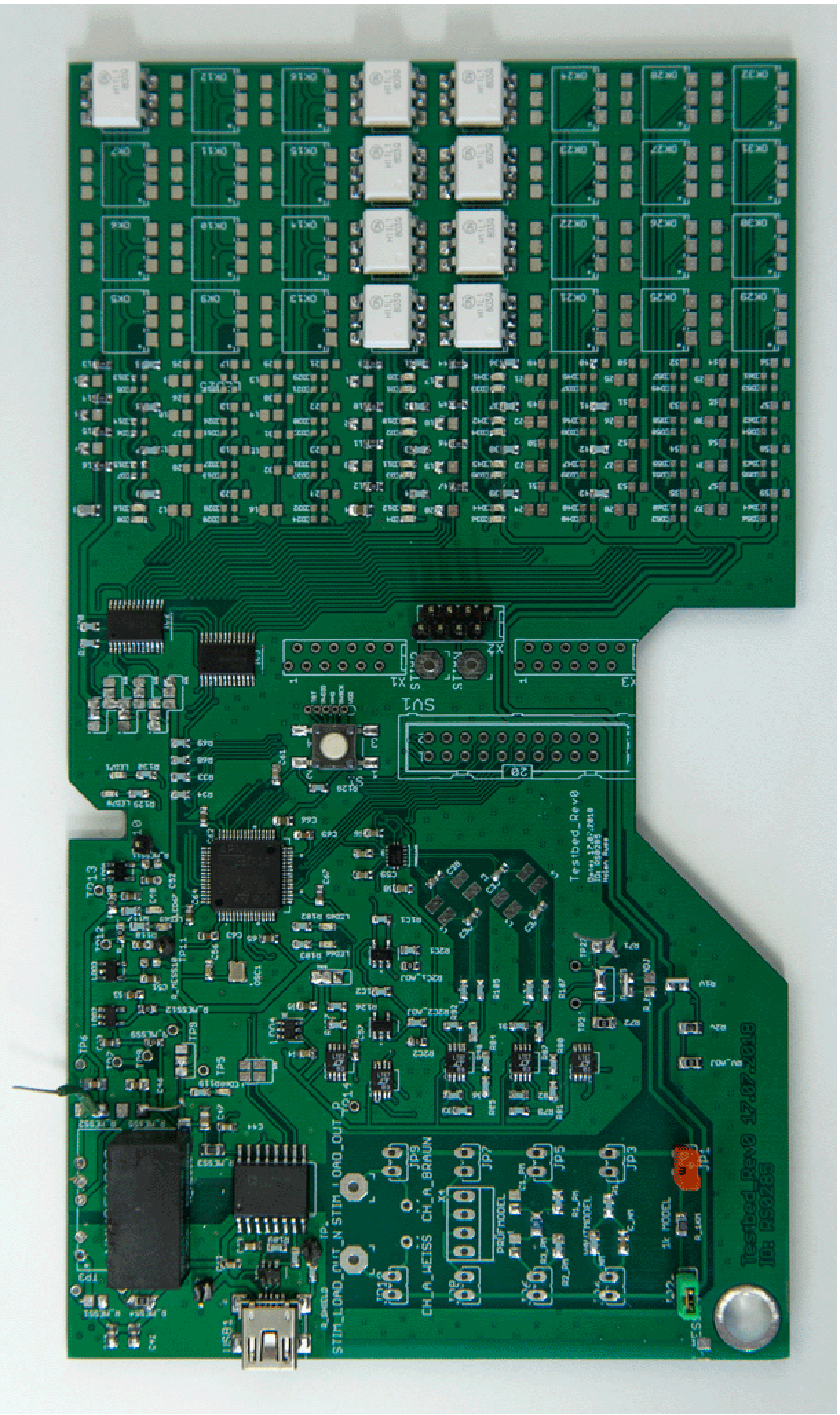
15.08.18 16:27 f=0.75 /home/russ/BA/ba_testbed/HW/EAGLE/Testbed/Testbed_rev0.sch (Sheet: 5/5)

B. EAGLE PCB layout



15.08.18 16:28 f=1.34 /home/russ/BA/ba_testbed/HW/EAGLE/Testbed/Testbed_rev0.brd

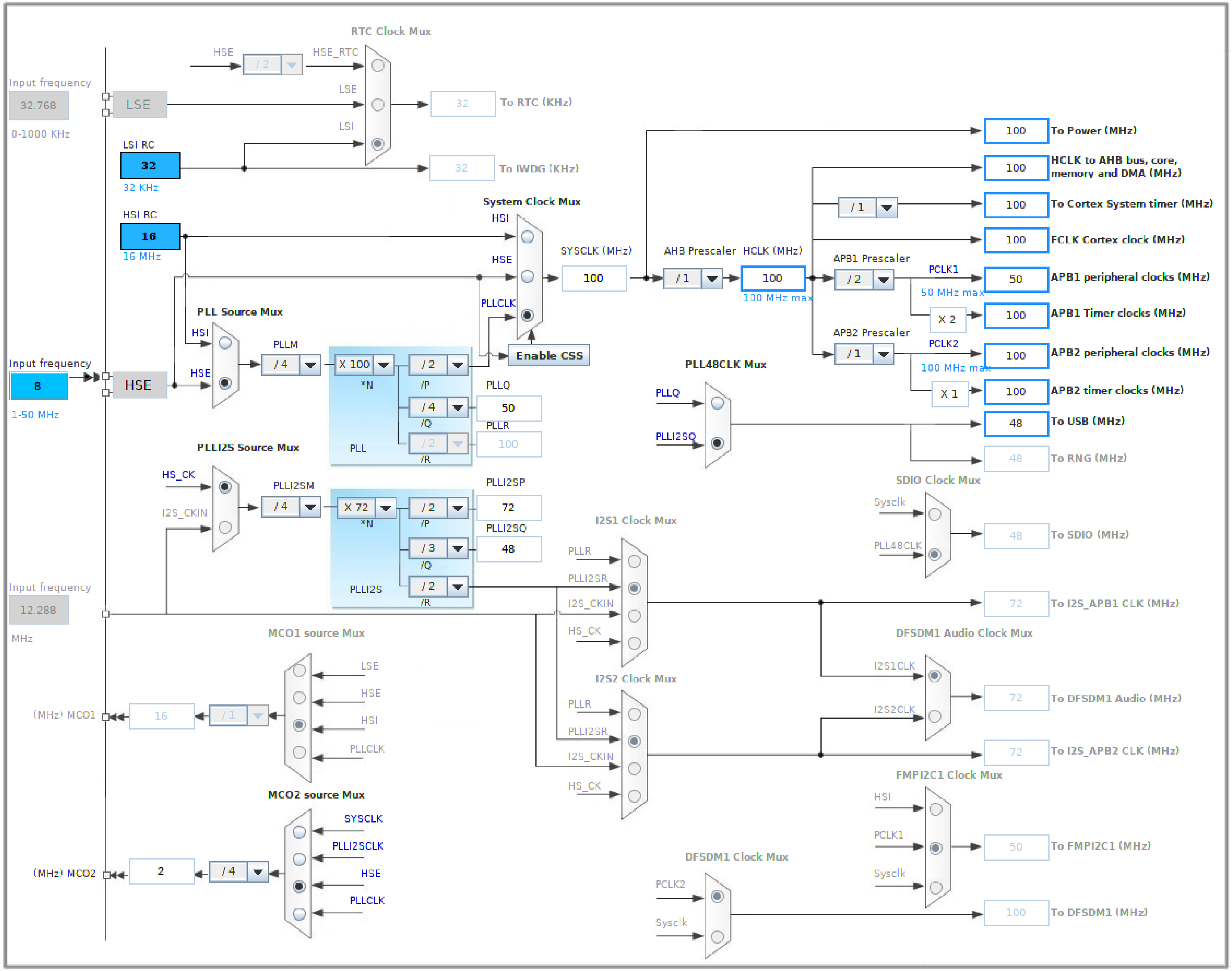
C. Built prototype



D. Bill Of Materials

Part	Value	Device	Package	Description	Quantity	Stocked	To Order	Status	DIGI-KEY_PART_NUMBER	DIGI-KEY_PURCHASE_URL	MANUFACTURER_PART_NU	MF	UNIT PRICE ($)	PRICE ($)
LED10	Red	LEDCHIP-LED0603	CHIP-LED0603	LED	38	10	10	Active	754-1117-1-ND	https://www.digikey.com/pı	APT1608EC	KINGBRIGHT	0.37	3.7
COMP1	TS3021	TS3021	SOT95P280X145-5N		2	0	2	Active	497-17815-1-ND	https://www.digikey.com/pı	TS3021AILT	STMicroelectronics	1.48	2.96
DC-DC1	NMH0509SC	NMH0509SC	NMH		1	0	1	Active	811-1507-5-ND	https://www.digikey.com/pı	NMH0509SC	Murata	13.2	13.2
DC-DC2	MEJ2S0503SC	MEJ2S0503SC	MEJ2		1	0	1	Active	811-2398-ND	https://www.digikey.com/pı	MEJ2S0503SC	Murata	10.15	10.15
IC1	USBLC6-2	USBLC6-2	SOT-23-6	USBLC6-2 Very low capacitance ESD protection	1	0	1	Active	497-5235-1-ND	https://www.digikey.com/pı	USBLC6-2SC6	STMicroelectronics	0.49	0.49
ISO_1	ADUM4160	ADMU4160	SOIC16WIDE	ADuM4160 Digital USB Isolator	1	0	1	Active	ADUM4160BRWZ-RLCT-ND	https://www.digikey.com/pı	ADUM4160BRWZ-RL	Analog Devices	13.03	13.03
J1	1-1734035-1	1-1734035-1	TE_1-1734035-1	CONN MINI B USB R/A SMD	1	0	1	Active	A123089-ND	https://www.digikey.de/pro	1734035-1	TE Connectivity	1.24	1.24
LDO1	LDK220M50R +5V	LDK220	SOT-23-5	ST 200 mA low quiescent current and low noise LDO	1	0	1	Active	497-14827-1-ND	https://www.digikey.com/pı	LDK220M50R	STMicroelectronics	0.93	0.93
LDO2	TLV73330DBVT	TLV73330DBVT	SOT95P280X145-5N		1	0	1	Active	296-42284-1-ND	https://www.digikey.com/pı	TLV73330PDBVT	Texas Instruments	0.4	0.4
LDO3	MIC5270-5.0YM5	MIC5270-X.X	SOT-23-5	Negative LDO Voltage Regulator - MIC270-x.x	1	0	1	Active	576-1294-1-ND	https://www.digikey.com/pı	MIC5270-5.0YM5-TR	Micrel	1.59	1.59
OSC1	X1G004171001512	SG-210STF-IQXO-794	2.5X2-4-PAD	OSCILLATOR WITH OE	1	0	1	Active	SER3858-ND	https://www.digikey.com/pı	SG-210STF 8.0000MY	Epson Timing	1.49	1.49
U3	LT1790BCS6-2.5TRMPBF	LT1790BCS6-2.5TRMPBF	TSOT-23_S6		1	0	1	Active	LT1790ACS6-2.5#TRMPBFDKR-ND	https://www.digikey.com/pı	LT1790ACS6-2.5#TRMPBF	Analog Devices	7.15	7.15
U4	STM32F4XXR2	STM32F4XXR2	LQFP64-10X10MM		1	0	1	Active	497-16764-ND	https://www.digikey.com/pı	STM32F412RGT6	STMicroelectronics	9.03	9.03
OK1	H11L1M	H11L1SR2M	6-SMD		32	10	0	Active	H11L1SR2MCT-ND	https://www.digikey.com/pı	H11L1SR2M	FAIRCHILD SEMICONDU	1.12	0
LED1	Green	LEDCHIPLED_0603	CHIPLED_0603	LED	33	20	0	OBSOLETE			HT-193UYG-5592	INOLUX	0	0
ADC1	MAX11198	MAX11198	T1623CN+1	Dual ADC MAX11198	1	1	0	Active	MAX11198ATE+-ND	https://www.digikey.com/pı	MAX11198ATE+	Maxim Integrated	16.06	0
BINDER1	BINDER_STIM	BINDER_STIM	5,5-PAD	Hole for the Binder Stimulation Connector	1	1	0						0	0
C1	1n	C-EUC0805	C0805	CAPACITOR, European symbol	32	32	0						0	0
C33	220n	C-EUC0805	C0805	CAPACITOR, European symbol	1	1	0						0	0
C34	330n	C-EUC0805	C0805	CAPACITOR, European symbol	2	2	0						0	0
C36	8n	C-EUC0603K	C0603K	CAPACITOR, European symbol	4	4	0						0	0
C40	2.2u/25V/X7R/10%	C-EUC0805	C0805	CAPACITOR, European symbol	3	3	0						0	0
C41	1u/X7R	C-EUC0805	C0805	CAPACITOR, European symbol	1	1	0						0	0
C42	100nF/25V/X7R/10%	C-EUC0603K	C0603K	CAPACITOR, European symbol	6	6	0						0	0
C47	4.7u/X7R	C-EUC0805	C0805	CAPACITOR, European symbol	1	1	0						0	0
C48	1uF/25V/X7R/10%	C-EUC0603K	C0603K	CAPACITOR, European symbol	3	3	0						0	0
C49	2.2uF/25V/X7R/10%	C-EUC0603K	C0603K	CAPACITOR, European symbol	3	3	0						0	0
C51	4.7uF/25V/X7R/10%	C-EUC0603K	C0603K	CAPACITOR, European symbol	6	6	0						0	0
C55	1u/X5R	C-EUC0603K	C0603K	CAPACITOR, European symbol	2	2	0						0	0
C58	10u/X7R	C-EUC0603K	C0603K	CAPACITOR, European symbol	2	2	0						0	0
C59	16V/10u/X5R	C-EUC0603K	C0603K	CAPACITOR, European symbol	2	2	0						0	0
JP1		JP1Q	JP1	JUMPER	10	10	0						0	0
JP11		PINHD-1X5X1.27	1.27_1X05_S	PIN HEADER	1	1	0						0	0
L1	40.58u	L-EUL2825P	L2825P	INDUCTOR, European symbol	4	4	0						0	0
OP1		LT1819	MSOP8	LT1819 400MHz OPV	5	5	0	Active	LT1819IMS8#PBF-ND	https://www.digikey.com/pı	LT1819IMS8#PBF	Analog Devices	3.92	0
PAD1	WIREPAD3,81/1,4	WIREPAD3,81/1,4	3,81/1,4	Wire PAD connect wire on PCB	2	2	0						0	0
PAD3		3,81/1,4	3,81/1,4	THROUGH-HOLE PAD	4	4	0						0	0
R1	51	R-EU_R0603	R0603	RESISTOR, European symbol	36	36	0						0	0
R9	100k	R-EU_R0603	R0603	RESISTOR, European symbol	32	32	0						0	0
R68	47k/10%	R-EU_R0603	R0603	RESISTOR, European symbol	4	4	0						0	0
R70	33k/10%	R-EU_R0603	R0603	RESISTOR, European symbol	2	2	0						0	0
R71	0R	R-EU_R0603	R0603	RESISTOR, European symbol	2	2	0						0	0
R73	ADJ	R-EU_R0805	R0805	RESISTOR, European symbol	2	2	0						0	0
R74	6R7	R-EU_R0805	R0805	RESISTOR, European symbol	1	1	0						0	0
R75	1k	R-EU_R0805	R0805	RESISTOR, European symbol	2	2	0						0	0
R76	1M49	R-EU_R0805	R0805	RESISTOR, European symbol	1	1	0						0	0
R77	10k	R-EU_R0805	R0805	RESISTOR, European symbol	1	1	0						0	0
R79	1k	R-EU_R0603	R0603	RESISTOR, European symbol	13	13	0						0	0
R88	470R	R-EU_R0805	R0805	RESISTOR, European symbol	1	1	0						0	0
R89	2k7	R-EU_R0805	R0805	RESISTOR, European symbol	1	1	0						0	0
R94	5k	R-EU_R0603	R0603	RESISTOR, European symbol	2	2	0						0	0
R95	7k	R-EU_R0603	R0603	RESISTOR, European symbol	1	1	0						0	0
R97	ADJ	R-EU_R0603	R0603	RESISTOR, European symbol	2	2	0						0	0
R99	1k8	R-EU_R0805	R0805	RESISTOR, European symbol	2	2	0						0	0
R102	1k/1%	R-EU_R0603	R0603	RESISTOR, European symbol	3	3	0						0	0
R108	10/5%	R-EU_R0603	R0603	RESISTOR, European symbol	1	1	0						0	0
R109	24/1%	R-EU_R0603	R0603	RESISTOR, European symbol	2	2	0						0	0
R111	10?	R-EU_R0805	R0805	RESISTOR, European symbol	3	3	0						0	0
R114	0	R-EU_R1206	R1206	RESISTOR, European symbol	2	2	0						0	0
R116	0	R-EU_R0805	R0805	RESISTOR, European symbol	1	1	0						0	0
R117	2k/1%	R-EU_R0603	R0603	RESISTOR, European symbol	4	4	0						0	0
R120	0	R-EU_R0603	R0603	RESISTOR, European symbol	6	6	0						0	0
R126	100k/1%	R-EU_R0603	R0603	RESISTOR, European symbol	3	3	0						0	0
S1		TASTERSMD	TAST_SMD	Taster - 6mm x 6mm	1	1	0						0	0
SV1		ML20	ML20	HARTING	1	1	0						0	0
TP1	TESTPOINT_TH	TESTPOINT_TH	TP1-09	Testpoint	16	16	0						0	0
U1	TCA9535PWR	TCA9535PWR	SOP65P640X120-24N	I/O EXPANDER	2	2	0	Active	296-25129-1-ND	https://www.digikey.com/pı	TCA9535PWR	Texas Instruments	1.5	0
X1		MTOP-12	MTOP-12	AMP connector	2	2	0						0	0
X2		MTOP-8	MTOP-8	AMP connector	1	1	0						0	0
X4	22-23-2041	22-23-2041	22-23-2041	.100 (2.54mm) Center Header - 4 Pin"	1	1	0					MOLEX	0	0
														$65.36

E. CubeMX Clock Configuration


RTC Clock Mux
HSE
/ 2
HSE_RTC
LSE
LSI
32
To RTC (KHz)
32
To IWDG (KHz)
Input frequency
32.768
0-1000 KHz
LSE
LSI RC
32
32 KHz
HSI RC
16
16 MHz
System Clock Mux
HSI
HSE
PLLCLK
SYSCLK (MHz)
100
AHB Prescaler
/ 1
HCLK (MHz)
100
100 MHz max
Enable CSS
PLL Source Mux
HSI
HSE
PLLM
/ 4
X 100
*N
/ 2
/P
/ 4
/Q
/ 2
/R
PLL
PLLQ
50
PLLR
100
Input frequency
8
1-50 MHz
HSE
PLLI2S Source Mux
HS_CK
I2S_CKIN
PLLI2SM
/ 4
X 72
*N
/ 2
/P
/ 3
/Q
/ 2
/R
PLLI2S
PLLI2SP
72
PLLI2SQ
48
Input frequency
12.288
MHz
100
To Power (MHz)
100
HCLK to AHB bus, core, memory and DMA (MHz)
/ 1
100
To Cortex System timer (MHz)
100
FCLK Cortex clock (MHz)
APB1 Prescaler
/ 2
PCLK1
50 MHz max
50
APB1 peripheral clocks (MHz)
X 2
100
APB1 Timer clocks (MHz)
APB2 Prescaler
/ 1
PCLK2
100 MHz max
100
APB2 peripheral clocks (MHz)
X 1
100
APB2 timer clocks (MHz)
PLL48CLK Mux
PLLQ
PLLI2SQ
48
To USB (MHz)
48
To RNG (MHz)
SDIO Clock Mux
Sysclk
PLL48CLK
48
To SDIO (MHz)
I2S1 Clock Mux
PLLR
PLLI2SR
I2S_CKIN
HS_CK
72
To I2S_APB1 CLK (MHz)
DFSDM1 Audio Clock Mux
I2S1CLK
I2S2CLK
72
To DFSDM1 Audio (MHz)
I2S2 Clock Mux
PLLR
PLLI2SR
I2S_CKIN
HS_CK
72
To I2S_APB2 CLK (MHz)
FMPI2C1 Clock Mux
HSI
PCLK1
Sysclk
50
To FMPI2C1 (MHz)
DFSDM1 Clock Mux
PCLK2
Sysclk
100
To DFSDM1 (MHz)
MCO1 source Mux
LSE
HSE
HSI
PLLCLK
(MHz) MCO1
16
/ 1
MCO2 source Mux
SYSCLK
PLLI2SCLK
HSE
PLLCLK
(MHz) MCO2
2
/ 4

F. ATES.h

```c

#ifndef ATES_H_
#define ATES_H_

#include "spi.h"
#include "i2c.h"

#define ATES__NUMBER_OF_MEASUREMENTS   2
#define ATES__SPI_BUFFER_SIZE          16 //ADC's Resolution
#define ATES__V_hSPI                hspi1
#define ATES__I_hSPI                hspi4

#define ATES__PIOE_ADDRESS          0b01001000
#define ATES__NIOE_ADDRESS          0b01001100
#define ATES__I2C_NUMBER_OF_PORTS    2
#define ATES__I2C_BUFFER_SIZE       ATES__I2C_NUMBER_OF_PORTS //Buffer is 8 bit array structure, (16 inputs for 1 I/O Expander -> 2 bytes)
#define ATES__PIOE_hi2c             hi2c1
#define ATES__NIOE_hi2c             hi2c2


extern uint8_t ATES_V_Buffer[ATES__SPI_BUFFER_SIZE];
extern uint8_t ATES_I_Buffer[ATES__SPI_BUFFER_SIZE];

extern uint16_t ATES_PIOE_Address;
extern uint16_t ATES_NIOE_Address;
extern uint8_t ATES_PIOE_Buffer[ATES__I2C_BUFFER_SIZE]; //For Debugging
extern uint8_t ATES_NIOE_Buffer[ATES__I2C_BUFFER_SIZE]; //For Debugging
extern uint8_t ATES_PIOE_Port1;
extern uint8_t ATES_PIOE_Port0;
extern uint8_t ATES_NIOE_Port1;
extern uint8_t ATES_NIOE_Port0;


void ATES_ReadADC(void);

void ATES_ReadPIOE(void);
void ATES_ReadNIOE(void);



#endif /* ATES_H_ */
```

G. ATES.c

```c

#include "ATES.h"

uint8_t ATES_V_Buffer[ATES__SPI_BUFFER_SIZE];
uint8_t ATES_I_Buffer[ATES__SPI_BUFFER_SIZE];

uint16_t ATES_PIOE_Address = ATES__PIOE_ADDRESS >> 1;
uint16_t ATES_NIOE_Address = ATES__NIOE_ADDRESS >> 1;
uint8_t ATES_PIOE_Buffer[ATES__I2C_BUFFER_SIZE];
uint8_t ATES_NIOE_Buffer[ATES__I2C_BUFFER_SIZE];
uint8_t ATES_PIOE_Port1;
uint8_t ATES_PIOE_Port0;
uint8_t ATES_NIOE_Port1;
uint8_t ATES_NIOE_Port0;

void ATES_ReadADC(void){
  HAL_SPI_Receive_DMA(&ATES__V_hSPI, ATES_V_Buffer, ATES__SPI_BUFFER_SIZE);
  HAL_SPI_Receive_DMA(&ATES__I_hSPI, ATES_I_Buffer, ATES__SPI_BUFFER_SIZE);
}

void ATES_ReadPIOE(void){ //TODO: Re-order Data Structure and optimize
  /* Data Structure
   * Port 0:
   *  DC_P8 (MSB)
   *  DC_P7
   *  DC_P6
   *  DC_P5
   *  DC_P4
   *  DC_P3
   *  DC_P2
   *  DC_P1 (LSB)
   *
   * Port 1:
   *  DC_P9 (MSB)
   *  DC_P10
   *  DC_P11
   *  DC_P12
   *  SC_1
   *  SC_P2
   *  SC_P3
   *  SC_P4 (LSB)
   */

  //The device 7 bits address value in datasheet must be shift at right before call interface

  //Write Register Addresses to be Read

  ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-2] = 0x00; //Port 0 Set to Input
  ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-1] = 0x01; //Port 1 Set to
```

```
    Input
  HAL_I2C_Master_Transmit_DMA(&ATES__PIOE_hi2c, ATES_PIOE_Address, ATES_PIOE_Buffer, ATES__I2C_BUFFER_SIZE);

  //Read Data from I/O Expander
  HAL_I2C_Master_Receive_DMA(&ATES__PIOE_hi2c, ATES_PIOE_Address, ATES_PIOE_Buffer, ATES__I2C_BUFFER_SIZE);
  ATES_PIOE_Port1 = ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-1]; //TODO: Expand for more than two ports for 1 I/O Expander
  ATES_PIOE_Port0 = ATES_PIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-2];
}

void ATES_ReadNIOE(void){
  /* Data Structure
   * Port 0:
   *  DC_N4 (MSB)
   *  DC_N3
   *  DC_N2
   *  DC_N1
   *  SC_N4
   *  SC_N3
   *  SC_N2
   *  SC_N1 (LSB)
   *
   * Port 1:
   *  DC_N5 (MSB)
   *  DC_N6
   *  DC_N7
   *  DC_N8
   *  SC_N9
   *  SC_N10
   *  SC_N11
   *  SC_N12 (LSB)
   */

  //The device 7 bits address value in datasheet must be shift at right before call  interface

  //Write Register Addresses to be Read

  ATES_NIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-2] = 0x00; //Port 0 Set to Input
  ATES_NIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-1] = 0x01; //Port 1 Set to Input
  HAL_I2C_Master_Transmit_DMA(&ATES__NIOE_hi2c, ATES_NIOE_Address, ATES_NIOE_Buffer, ATES__I2C_BUFFER_SIZE);

  //Read Data from I/O Expander
  HAL_I2C_Master_Receive_DMA(&ATES__NIOE_hi2c, ATES_NIOE_Address, ATES_NIOE_Buffer, ATES__I2C_BUFFER_SIZE);
  ATES_NIOE_Port1 = ATES_NIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-1]; //TODO: Expand for more than two ports for 1 I/O Expander
  ATES_NIOE_Port0 = ATES_NIOE_Buffer[ATES__I2C_NUMBER_OF_PORTS-2];
```

```
}
```

H. main.c

```c
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "stm32f4xx_hal.h"
#include "dma.h"
#include "i2c.h"
#include "spi.h"
#include "usart.h"
#include "usb_otg.h"
#include "gpio.h"

/* USER CODE BEGIN Includes */
#include <string.h>
#include "./../App/GSBP_F4.h" // relative path
#include "./../App/ATES.h"
/* USER CODE END Includes */

/* Private variables ---------------------------------------------------------*/

/* USER CODE BEGIN PV */
/* Private variables ---------------------------------------------------------*/
uint8_t DoMeasurments;
uint16_t MeasTestDelay;
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----------------------------------------------*/

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  *
  * @retval None
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration----------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_I2C1_Init();
  MX_I2C2_Init();
  MX_SPI1_Init();
  MX_SPI4_Init();
  MX_USART6_UART_Init();
  MX_USB_OTG_FS_PCD_Init();
  /* USER CODE BEGIN 2 */
  GSBP_Init();
   // enable all already available interfaces
   GSBP_Handles.DoEnableInterface |= GSBP_Interface_2; // Change interface when using USB.
// GSBP_Handles.DoEnableInterface |= GSBP_Interface_1;

   // start the communication interfaces
   GSBP_ManageInterfaces(GSBP_InterfaceEnable);

   uint32_t GSBP_NextCall = HAL_GetTick() + GSBP_CONFIG__CALLBACK_PERIOD_IN_MS;


   GSBP_Handles.ACK.CommandID = DebugCMD;
   sprintf((char*)GSBP_Handles.ACK.Data, "System reseted -> timer = %lu", GSBP_NextCall);
   GSBP_Handles.ACK.DataSize = 30;
   GSBP_SendPackage(&GSBP_UART, &GSBP_Handles.ACK); //Change pointer when using USB
// GSBP_SendPackage(&GSBP_USB, &GSBP_Handles.ACK);
   DoMeasurments = 0;
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
    /*
         *  check for new packages every GSBP_CONFIG__CALLBACK_PERIOD_IN_MS ms
         */
        if (HAL_GetTick() >= GSBP_NextCall) {
          GSBP_EvaluatePackage(&GSBP_UART);
          //GSBP_EvaluatePackage(&GSBP_USB);
          GSBP_NextCall += GSBP_CONFIG__CALLBACK_PERIOD_IN_MS;
        }

        if (DoMeasurments) {

          GSBP_Handles.ACK.CommandID = MeasurmentDataACK;
          sprintf((char*)GSBP_Handles.ACK.Data, "Data package ... -> timer = %lu", GSBP_NextCall);
          GSBP_Handles.ACK.DataSize = strlen((const char*)GSBP_Handles.ACK.Data);
          GSBP_SendPackage(&GSBP_UART, &GSBP_Handles.ACK);

          HAL_Delay(MeasTestDelay);
        }

    if(!PIOE_INT_Pin){ //Low state means active for INT pin
      ATES_ReadPIOE();
    }
    if(!NIOE_INT_Pin){ //Low state means active for INT pin
      ATES_ReadNIOE();
    }
/*
 * Hello World!! test.
 */
    HAL_Delay(500);
    HAL_GPIO_TogglePin(GPIOC, STATUS2_Pin);


//TODO:Call ATES_ReadADC at rising edge of PC9 pulse (MCO2).

  }
  /* USER CODE END 3 */

}

/**
  * @brief System Clock Configuration
  * @retval None
  */
```

```
void SystemClock_Config(void)
{

  RCC_OscInitTypeDef RCC_OscInitStruct;
  RCC_ClkInitTypeDef RCC_ClkInitStruct;
  RCC_PeriphCLKInitTypeDef PeriphClkInitStruct;

    /**Configure the main internal regulator output voltage
    */
  __HAL_RCC_PWR_CLK_ENABLE();

  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_BYPASS;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 100;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  RCC_OscInitStruct.PLL.PLLR = 2;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

    /**Initializes the CPU, AHB and APB busses clocks
    */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_3) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

  PeriphClkInitStruct.PeriphClockSelection = RCC_PERIPHCLK_CLK48;
  PeriphClkInitStruct.PLLI2S.PLLI2SN = 72;
  PeriphClkInitStruct.PLLI2S.PLLI2SM = 4;
  PeriphClkInitStruct.PLLI2S.PLLI2SR = 2;
  PeriphClkInitStruct.PLLI2S.PLLI2SQ = 3;
  PeriphClkInitStruct.Clk48ClockSelection = RCC_CLK48CLKSOURCE_PLLI2SQ;
  PeriphClkInitStruct.PLLI2SSelection = RCC_PLLI2SCLKSOURCE_PLLSRC;
  if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInitStruct) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

  HAL_RCC_MCOConfig(RCC_MCO2, RCC_MCO2SOURCE_HSE, RCC_MCODIV_4);

    /**Configure the Systick interrupt time
    */
  HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

    /**Configure the Systick
    */
  HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

  /* SysTick_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}
```