Politecnico di Torino

DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING

Master Course in Mechatronic Engineering

Master Graduation Thesis

Eye Safe Lidar Prototype

Design, implementation and test of the electronic system

Candidate

Amedeo Perotto

s255618

Relator

Professor Janner Davide

Academic Year 2019/2020

Introduction

This project focuses on the development of a receiver circuit of a lidar system and the measurement of the time of flight. In particular, it will focus on the electrical design of such a circuit, searching for the best tradeoff between simplicity and performance. Moreover, the entire lidar system is studied and the experimental setup is established according to the system specifications.

The motivations which prompted the study of this topic have a dual nature. The passion in designing electrical circuits, enriched over the years, has been influenced and enhanced by experiences lived during the university courses. The second one, but not secondary, is the increasing use of lidar systems in a wide range of engineering fields, spacing from the distance evaluation to the environmental analysis.

The objective of this project is to design, evaluate and assess the receiving circuit of a lidar system and the used experimental setup which constitute the prototype. In this way, the paper will provide a detailed design procedure of the circuit and the explanation of the setup used to test the performances of the board. It can be the basis for further studies of this kind of system; evolutions and updates are always possible.

The thesis is structured in chapters: The first introduces the lidar architectures and the main advantages and disadvantages of each one. The second provide an overview of the photodetector used in this prototype. The third concerns some safety aspects that must be well-know before working with system containing lasers. In the fourth chapter, the device used to direct the laser beam towards the target and all the components necessary to operate this device are analyzed and explained. The fifth and sixth, expound and analyze the digital-to-analog converter and the analog-to-digital converter used in this project and the relative implementation in the system. The detailed design procedure of the receiving circuit is explained in chapter seven with the analysis of the main errors that could affect the designed circuit.

Chapter eight contains the PCB realization procedure and the soldering activity performed to make the circuit. In chapter nine, is expounded the software and code written to show the acquired data. Chapter ten contains the experimental setup implemented and the obtained results. In chapter eleven it is proceeded to comment on the results obtained by processing the obtained data.

Thanks to this research it was possible to analyze some important feature of these systems and to provide a well-performing architecture that can be used to perform distance measurements. The performances of the studied prototype are deeply analyzed in the conclusions.

Acknowledgment

I would like to thank the following people, without whom I would not have been able to complete this research, and without whom I would not have made it through my master's degree!

My supervisor Professor Janner D., whose insight and knowledge into the subject matter steered me through this research. And special thanks to Franco G., whose support allowed my studies to go the extra mile (sorry for all the extra work Giuseppe!).

The staff of ISMB, who supported my work and allowed me to use their facilities and instrumentations without which I would have no content for my thesis.

I would like to thank my parents whose love and guidance are with me in whatever I pursue. I must admit, even my brothers Zeno and Filippo helped me making the days more fun. A special thanks go to my brother and my best friend, Gabriele, who, even if he is on the other side of the world, cheers me on and supports me, moreover he selected excellent "creation"; I would like to tell him: "Si, la radio a gallina funziona!".

Most importantly, heartfelt thanks to my loving and supportive girlfriend, Marina, who has been by my side throughout this thesis and who provide unending inspiration and for believing in me and encouraging me to follow my dreams.

Summary

List of Tables

List of Figures

1 LIDAR architectures

This chapter is intended to provide an overview of the relevant architectures available for this project and to discuss advantages and disadvantages of each of them.

A preliminary consideration, which is valid for all receiver architectures presented, concerning the emitted and received optical power of the laser, is necessary. Since the laser must operates according to IEC 60825, which is applicable to safety of laser products emitting laser radiation, the optical output power of the laser is limited and thus, the received power. This affects and limits the maximum measuring distance.

1.1 Pulsed Time of Flight with Phase-shift based estimation algorithm

This architecture implements a pulsed time of flight stimulus with a phase-shift algorithm. The laser driver circuit uses a continuous wave of frequency f to modulate the amplitude of the emitted laser beam.

The design relies on the estimation of the phase of the reference and the return signals using the discrete Fourier transform. These two signals are processed using a non-windowed DFT to obtain the beat frequency and then, the distance is calculated using *Equation* (1).

$$d = \frac{\mathrm{c} \times \Delta\Theta}{4\pi f} \tag{1}$$

Where

- d = distance [m]
- c = speed of light [m/s]
- $\Delta\Theta$ = phase angle in radians
- f = frequency of the continuous wave of the reference signal

The advantages of this technology are that both the phase difference and the reflectance angle are obtained. These data can be used for identification and classification purposes. Moreover, it ensures fast computations and it is possible to obtain high measurement rates since the algorithm is based on the phase information of continuous waves that depend on the modulation frequency *f.*

The drawback of this technique is that the distance measured varies periodically depending on the modulation frequency, indeed, the phase shift between the two signals, increases as the distance increases and so, when the phase shift reaches 360°, from the receiver point of view, it is equivalent to the 0° phase shift. To solve this problem, it should be possible to implement a scheme in which a combination of two different modulation frequencies, such that they are not harmonics of each other, is used. In such a way, a lower frequency would be used to provide the distance approximation while the higher modulation frequency would provide the accuracy.

In the face of good performances and accuracy, the system is complex and requires the use of an FPGA and fast ADCs, which makes the system expensive.

1.2 Time to Digital Converter Based System

It is possible to estimate the distance by applying a short pulse to the laser and measuring the time from when it is emitted until it is received by the detector. The time-of-flight can be sensed using a time-to-digital converter (TDC); it acts as a stopwatch directly measuring the time between a start signal and a stop one. Making use of this measure, the distance can be computed using the *Equation* (2).

$$d = \frac{\mathrm{c} \times \mathrm{t}}{2} \quad (2)$$

Where

- d = distance [m]
- c = speed of light [m/s]
- t = ToF [s]

To guarantee accuracy and resolution over long distances, the system should present fast rise time, low noise detector and amplifiers [1].

Since the return signal amplitude is inversely proportional to the square of the measured distance, the system should manage the return signal level and, thus, the receiver requires automatic control gain. Moreover, a time discriminator is needed to guarantee that the triggering event occurs at a constant point in the rise of the return signal.

This method has the advantage that is easy to implement, and it can achieve good performances.

The main disadvantage is given by the fact that one pulse at a time is send, limiting the measuring rate. However, good rates are still possible thanks to the high speed of light.

1.3 High Speed ADC based system

The system is the same mentioned in the previous section except for the integrated circuit used to measure the time-of-flight. In this case the ToF is measured using a high-speed ADC and the resulting distance is computed using the *Equation 2*.

In a high-speed ADC based system, the ADC digitizes the return signal and so, signal processing can be employed to obtain additional information for target identification. This allows to relax the constraint of the laser rise time necessary for TDC based system and the wide dynamics of the ADC eases the ACG requirement in the receiving circuit.

Using high-speed ADCs, it is possible to obtain very accurate measures far from the purposes of this project [2].

1.4 Coherent Detection

This method can be used both in a phase detection system and when chirped laser pulses are applied. It consists in a system in which the mixing operation between the transmitted and received signal, is performed in the optical domain. Such configuration is affected by shot noise and thus characterized by a better Signal-to-Noise ratio than the architectures presented above [3].

More optical components are needed, increasing the cost and, furthermore, problems could arise when mixing two optical signals, producing random errors.

1.5 Chosen Architecture

An objective of this project is to maintain the system as simple as possible ensuring thorough, repeatable and reliable measures. Coherent mixing and phase-shift based architecture are rejected since the design would be too complex and costly. Even high-speed ADC based systems, are discharged because they are more appropriate for measuring very narrow distances.

Therefore, it was decided to develop the LIDAR receiving circuit using a TDC based architecture which ensures simplicity and reliable measurements.

2 Safety Aspects

In a LIDAR system, the main possible source of harmfulness is the laser beam.

When dealing with laser, it is necessary to minimize the risk of accidents. Due to the high intensity of the laser beam, they can be hazardous especially for the eyes.

In this chapter the relevant aspects related to laser safety, ranging from the structure of the human eye and the possible damage that affects it, up to the current international legislation, are analyzed.

2.1 Structure of human eye

To understand the bioeffects of laser-induced eye injuries, the structure of human eye must be investigated.

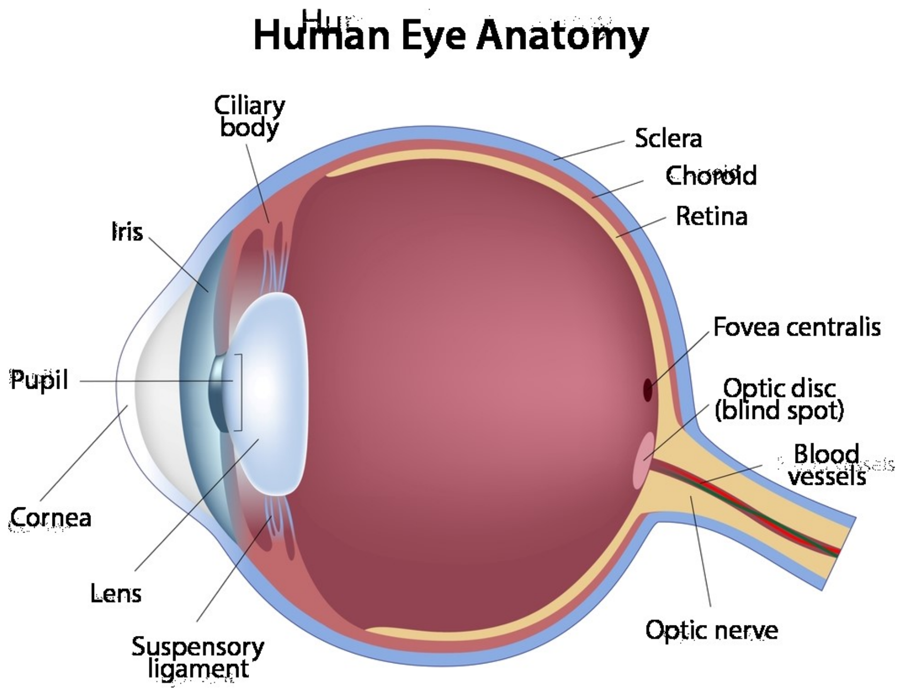


Fig. 1 Structure of Human Eye

The human eye acts as a camera:
the light enters the eye through the cornea, the amount of light that actually enters the eye is regulated by the iris, which controls the size of the pupil. The lens focuses the light that, passing through the vitreous humor, reaches the retina. Rods and cones, located in the retina, convert the light into electrical signals that travelling through the optic nerve reaches the brain which translates them in images.

It is therefore clear that the structures most prone to damage by laser beams are the cornea and retina.

2.2 Laser Beam: Possible Eye Damages

The cornea and the lens can focus the laser radiation to a small spot on the retina, this could cause permanent damage due to the high optical intensity of the laser beam, even after propagation over long distances.

The damage can present a thermal or photoelectrical nature, moreover, these damages could not be immediately noticed, for example, it is possible to burn peripheral regions of the retina which could be noticed years later.

The harmfulness depends on exposure parameters such as wavelength, exposure time and pulse energy.

In Table 1, it is presented an overview of some wavelengths range and the part of the eye each affects.

Table 1 Wavelength and Eye Damages

	Lengths range	Eye Parts affected	Damages
Ultraviolet	*180 to 400 nm*	Cornea and Lens	• Long term exposure can cause cataracts • Corneal flash burns • Photokeratitis
Visible Light	*400 to 700 nm*	Cornea, Lens and Retina	• Temporary Impaired Vision • Blurred Vision • Glare Responses
Near-Infrared	*700 to 1400 nm*	Cornea, Lens and Retina (Macula/Fovea)	• Blind Spots • Retinal Thermal Injury • Infrared Cataract
Mid-and-Far Infrared	*1400 nm to 1 mm*	Cornea and Lens	• Lens Damages • Corneal Thermal Injury

2.3 International Standards and Classification

There are different regulation schemes that specify exposure limits for human eye safety. Particularly important standards are:

- IEC 60825-1 international laser safety standard of Electrotechnical Commission (IEC)
- US user standard ANSI-Z-136 based one

The European standardization organization adopted the IEC standard as EN 60825-1. In this regulation, the lasers, are classified in different *safety classes*, with class 1 being the least dangerous and class 4 the most impactful one. The classification is based on the laser power, beam quality, laser wavelength and the accessibility of hazardous areas.

Table 2 provides the international lase safety classes classification with a simplified description of each class, for detail consult the IEC 60835-1.

Table 2 Safety Classes with simplified description [4]

Safety Classes	Simplified Description
1	The accessible laser radiation is not dangerous under reasonable conditions of use. Examples: 0.2-mW laser diode, fully enclosed 10-W Nd:YAG laser
1M	The accessible laser radiation is not hazardous, provided that no optical instruments are used, which may e.g. focus the radiation.
2	The accessible laser radiation is limited to the visible spectral range (400–700 nm) and to 1 mW accessible power. Due to the blink reflex, it is not dangerous for the eye in the case of limited exposure (up to 0.25 s). Example: some (but not all) laser pointers
2M	Same as class 2, but with the additional restriction that no optical instruments may be used. The power may be higher than 1 mW, but the beam diameter in accessible areas is large enough to limit the intensity to levels which are safe for short-time exposure.
3R	The accessible radiation may be dangerous for the eye but can have at most 5 times the permissible optical power of class 2 (for visible radiation) or class 1 (for other wavelengths).
3B	The accessible radiation may be dangerous for the eye, and under special conditions also for the skin. Diffuse radiation (as e.g. scattered from the some diffuse target) should normally be harmless. Up to 500 mW is permitted in the visible spectral region. Example: 100-mW continuous-wave frequency-doubled Nd:YAG laser
4	The accessible radiation is very dangerous for the eye and for the skin. Even light from diffuse reflections may be hazardous for the eye. The radiation may cause fire or explosions. Examples: 10-W argon ion laser, 4-kW thin-disk laser in a non-encapsulated setup

All these parameters in addition with the spatial distribution of the light from the laser, concur to define the safe exposure limits, known as *Maximum Permissible Exposure MPE*, for a given situation. The *MPE* is the highest power density that is considered safe for the eye. It is about 10% of the dose that has the 50% chance of creating damage under worst case conditions [5]. In this scenario, the eye lens and cornea focus the light at the smallest possible point on the retina for the wavelength and the pupil is fully open (0.39 cm^2). The computation results are pertinent for laser beams characterized by a cross-section that can pass through the fully open pupil. The international standard includes methods to compute the *MPE*.

The *MPE* is measured at the cornea of the human eye for a given wavelength and exposure time; the following graphs show the trend of the *MPE* for different exposure time and wavelength.

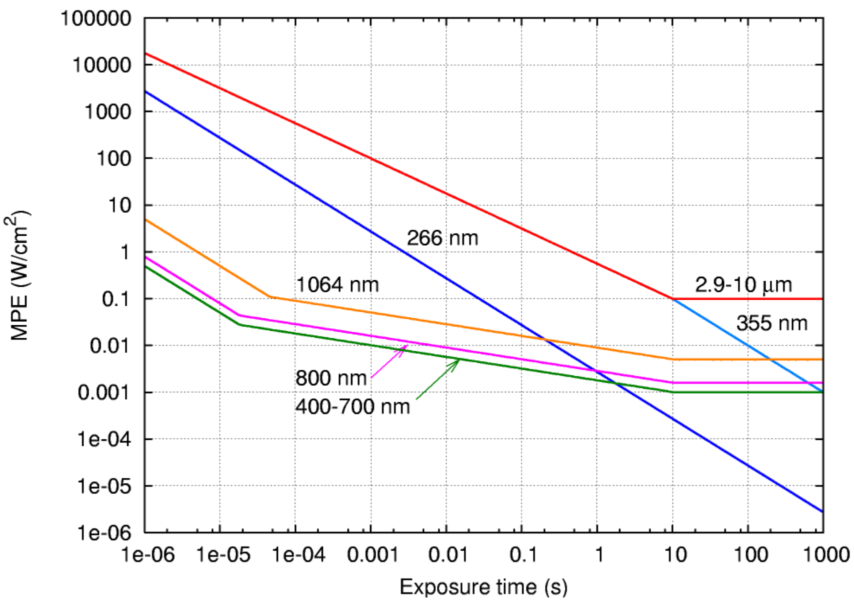


Fig. 2 MPE vs Exposure Time [6]

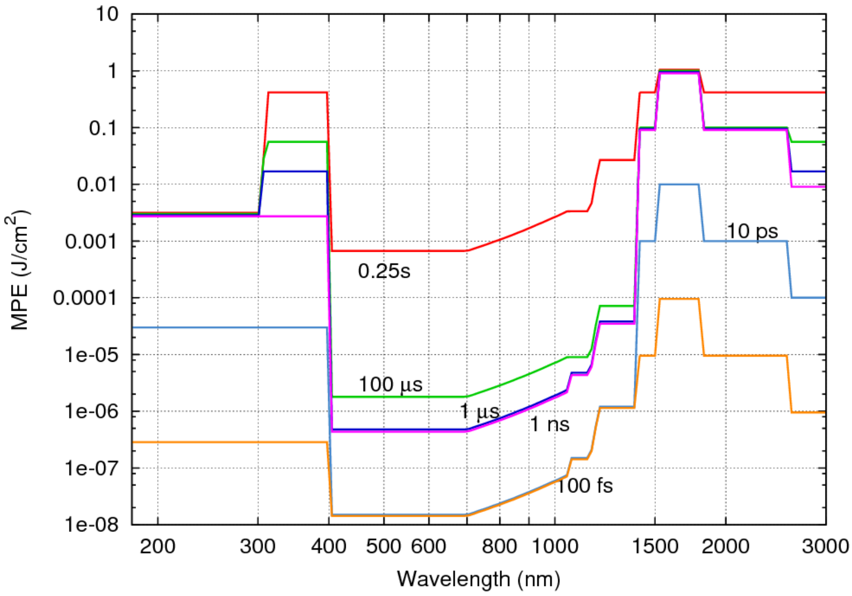


Fig. 3 MPE vs Wavelength [6]

This standard cover much more than solely the laser classification, it also prescribes the measures to be taken in order to work safely with laser products.

2.4 Nominal Hazard Zone

Since the international standards classify the laser product itself and do not evaluate a particular setup containing a laser, to assess the safety of a particular setup may be necessary to define a *"Nominal Hazard Zone".*
It is *the space within which the level of the direct, reflected or scattered radiation may exceed the applicable MPE.* [7]

In this region it is mandatory to apply much more restricting measures to guarantee the safety of the operators.

2.5 Laser Safety Precautions

The safety precautions necessary when using lasers, can be divided into two categories:

1. Technical Safety Precautions
2. Non-Technical Safety Precautions

The first concerns all the technical or, in other words, material solutions to avoid injuries when using a laser. These include, for example:

- The use of strongly absorbing radiation glosses
- Protective housing around the dangerous working areas
- Full or partial encapsulation of the laser system
- Beam Stoppers

Without an appropriate training of the operators, the technical precautions are not sufficient. The latter category includes some guidelines that should be well known by all the workers involved.

Some of these rules of thumb are:

- To assess the risks before any operation
- Establishing adequate working practices
- Safety education must be ensured
- The responsibility must be clearly assigned and defined

2.6 Conclusions

Since in this project a stock laser is used, and it is the duty of the manufacturer of the laser product to classify the product itself, the laser classification is not assessed by the author. However, knowing how to interpret the laser classification is necessary in order to establish the correct safety measures.

A complete safety assessment is, therefore, the details of the whole setup and the way it is used. From this point of view, both technical and non-technical have been taken during the experiments.

3 Photodetector

3.1 Introduction

The photodetector is the sensor positioned after the optics but before the electrical processing stages, which is responsible for the conversion of the optical signal, i.e. the incident light, into an electrical signal. Ideally the conversion is instantaneous, keeping the SNR, but due to the physical characteristics of the devices, the conversion process is not ideal.

There is a huge variety of such devices available, each of them suitable for a specific application. It is then necessary to compare the available devices with respect to certain parameters to find the best solution for the problem under analysis.

A brief introduction on the semiconductor physics is provided in this chapter, since it is considered necessary to discuss the properties of different detectors. At the end of this chapter, the selected device is presented and commented.

3.2 Photodetector Parameters

This section is intended to provide some common properties by which the photodetectors are characterized and used to compare their performances.

3.2.1 Responsivity

The responsivity of a photodetector is defined as a ratio of generated photocurrent to the incident light power at a given wavelength:

$$R(\lambda) = \frac{I_{PD}}{P} \left[\frac{A}{W}\right] \quad \textbf{(3)}$$

This property describes the response of the photodetector when exposed to light. Moreover, it is not constant over the entire spectrum and thus it is typically presented graphically versus wavelength. The choice of the photodetector depends on the wavelength of light, which is to be detected, since it is desirable to use a photodetector which presents peak responsivity at that wavelength. The responsivity is also related with the *quantum efficiency*. [8]

3.2.2 Frequency Response

The frequency response, also known as bandwidth, describes the behavior of the sensor when stressed by a time dependent optical signal. A fast detector is characterized by having a high cutoff frequency and thus, it is able to follow fast variation of the input signal.

This property is influenced by the underlying process by which the light is converted and might be altered by changing the operating conditions.

3.2.3 NEP: Noise-Equivalent-Power

All photodetectors output noise as well as the output signal. The amount of produced noise hinge on manufacturing as well as operating conditions. Since when the level of the signal equals the level of the noise it is not possible to distinguish the signal from the noise, this set the lower bound for the detectable optical signal power. It means that the optical input has to be strong enough to overcome the noise of the detector to have a measurable signal output.

The Noise Equivalent Power (NEP) is a measure of its responsivity with respect to the minimum optical signal power at a certain wavelength and signal bandwidth it can detect. It is defined as the input signal power that results in a signal-to-noise ratio (S/R) of 1 in a 1 Hz output bandwidth; its unit of measure is W/√Hz. A sensor with lower NEP can detect weaker signal than a detector with higher NEP.

To compute the NEP at different wavelength λ it is possible to use the following formula.

$$NEP(\lambda) = NEP_{min} \times \frac{R_{max}}{R(\lambda)} \quad \textbf{(4)}$$

Where NEP_{min} is the NEP as given in the specifications, while R_{max} is the maximum responsivity of the detector and $R(\lambda)$ is the responsivity of the detector at wavelength λ.

It is then possible to compute the minimum detectable power P_{min} using the following formula:

$$P_{min} = NEP(\lambda) \times \sqrt{BW} \quad \textbf{(5)}$$

3.3 Different Types of Photodetectors

Among the great variety of photodetectors available it is essential to choose the more adapted sensor for a given application. Several factors besides those mentioned in the previous paragraph, influence the choice; most often, these criteria limit the options.

Table 3 resumes the key characteristics of different semiconductor photodetectors while *Fig. 4* shows their absolute spectral response.

Table 3 Comparison of Different Photodetector [9]

Photodetector Type	Gain	Useful Spectral Range (nm)	Quantum Efficiency %	Response Time (ns)	Multiplied Dark Current (nA)
Si PIN	1	400-1150	60 - 90	0.3 - 3	-
Si APD	50 - 300	400-1150	70 - 80	0.5 - 5	0.1 – 1.0
Ge APD	10 - 100	800-1750	50 - 80	0.3 - 3	5 - 100
InGaAs Pin	1	900-1700	70 - 90	0.05 - 1	-
InGaAs APD	10 - 40	900-1700	60 - 90	0.1 - 1	0.5 - 5
Photomultiplier	105 - 107	160-850	30	0.15 - 13	10 - 200

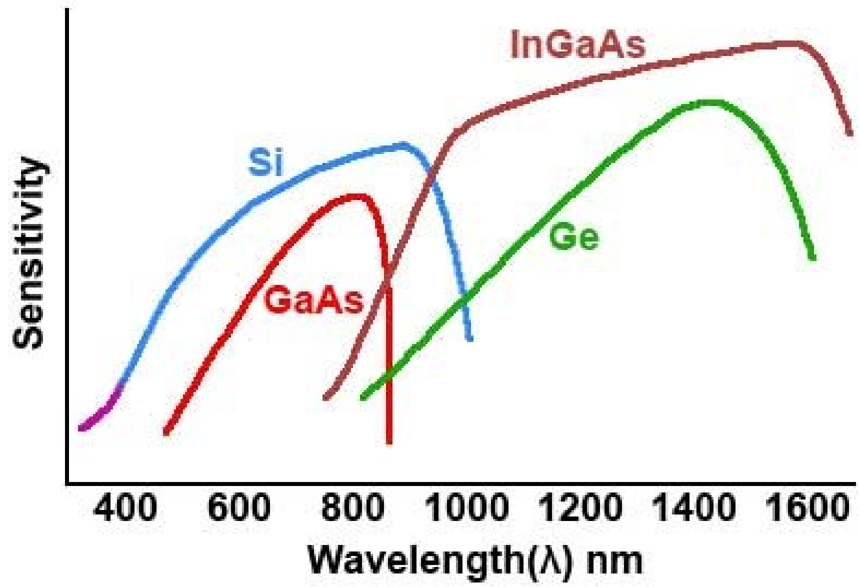


Fig. 4 Absolute Spectral Response for Different Photodetectors [10]

The detailed description of each photodetector will be left to the reader for exploration. [11]

Considering the detector demands for LIDAR applications, such as high sensitivity, wide dynamic range, usable under strong light condition, in this project the chosen photodetector is an InGaAs PIN diode, since is the simplest light-receiving element. Moreover, its sensitivity is stable, presents a wide dynamic range and it can be used under strong background light. The noise of the readout circuit determines the minimum receive level.

The table below resumes the characteristics of different photodetectors with respect to the demanded properties for LIDAR application.

Table 4 Photodetector Comparison Table [12]

Parameter	MPPC	APD	PIN photodiode
Range	Long	Long	Short
Accuracy	High	High	High
Readout circuit	Simple	Complex	Complex
Operation voltage	to several tens of V	100 to 200 V	to 10 V
Gain	10^5	10 to 100	1
Temperature sensitivity	Middle	High	Low
Response time	Fast	Medium	Medium
Ambient light immunity	Medium	Medium	High
Array	Suitable	Suitable	Suitable
Gap	Narrow	Wide	Wide
Uniformity	Good	Depends on the size	Good

3.4 The Chosen Photodetector

As mentioned before, an InGaAs photodetector is chosen for this project. The selected product is a PDA10CS InGaAs Switchable Gain Detector manufactured by ThorLabs [13]. It includes a reverse-biased PIN photo diode, mated to a switchable gain transimpedance amplifier; *Fig. 5* shows the schematics.

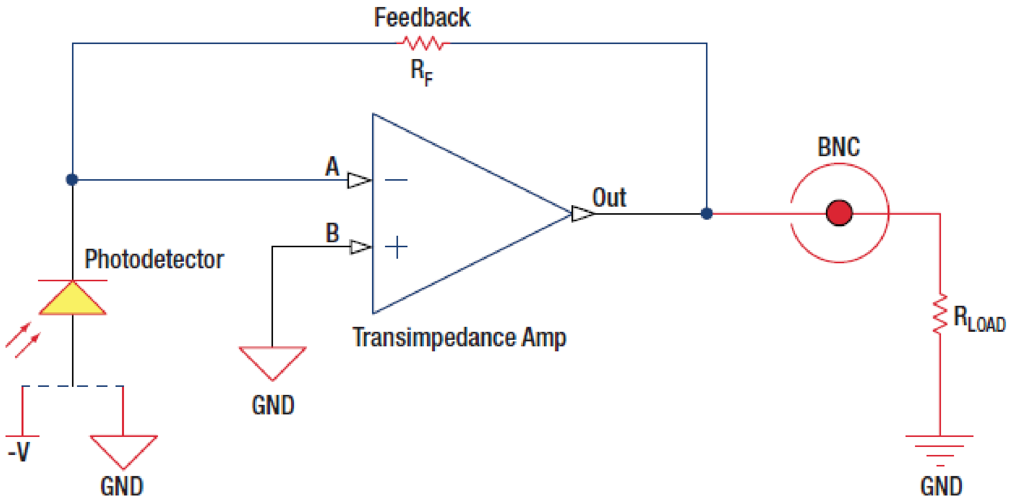


Fig. 5 PDA10CS Schematics [13]

This device implements a low noise, low offset, high gain transimpedance amplifier that allows gain setting. There are eight gain positions incremented in 10dB steps.

4 Scanning Galvo System

The laser scanning operation is performed using the GVS012 galvanometer system manufactured by Thorlabs. It is composed by two motor and mirrors, a mounting bracket, two driver cards and a post mounting plane.

4.1 The system

4.1.1 Introduction

A galvanometer is a current controlled motor with a limited travel. When a current is applied to the motor coils, the motor shaft rotates with an acceleration directly proportional to the supplied current. To stop the motion, a reverse polarity current have to be applied, while if the current is removed, the motor comes to rest under friction.

The following sections contain a detailed description of each component in the system.

4.1.2 The Galvanometer

The galvanometer is composed by a motor that moves the mirror and a detector that feeds back the position information.

Fig. 6 Galvo/Mirror Assembly [14]

The galvo motor used in the GVS012 system is realized in such a way that the magnet is part of the roto and the coil is part of the stator. This configuration guarantees faster response and higher system resonant frequencies when compared to moving coil configuration [14]. The feedback position system is composed by an optical detector which consists of two pairs of photodiodes and a light source. Different amounts of light are detected by the photodiode during the mirror motion and so, the current produced by the photodiodes is relative to the galvo actuator position.

4.1.3 The Mirror

The mirror assembly deflects the light beam according to the motor angular position.

The GVS012 implements protected silver-coated mirrors that are suitable for applications from 400 nm to near infra-red (2000 nm), and can support powers from 100 to 150 W/cm^2

Fig. 7 Motor/Mirror Assembly [14]

4.1.4 Servo Driver Board

To drive the actuator to the desired position, the system uses a servo circuit that interprets the signal from the position sensor and, making use of the positional error, speed and integral of current terms, outputs the control voltages.

The control scheme, represented by the diagram below, ensure excellent dynamic performance. Moreover, it includes a current factor to obtain stability at high acceleration, resulting in a higher system speeds compared to integrating servo systems. [14]

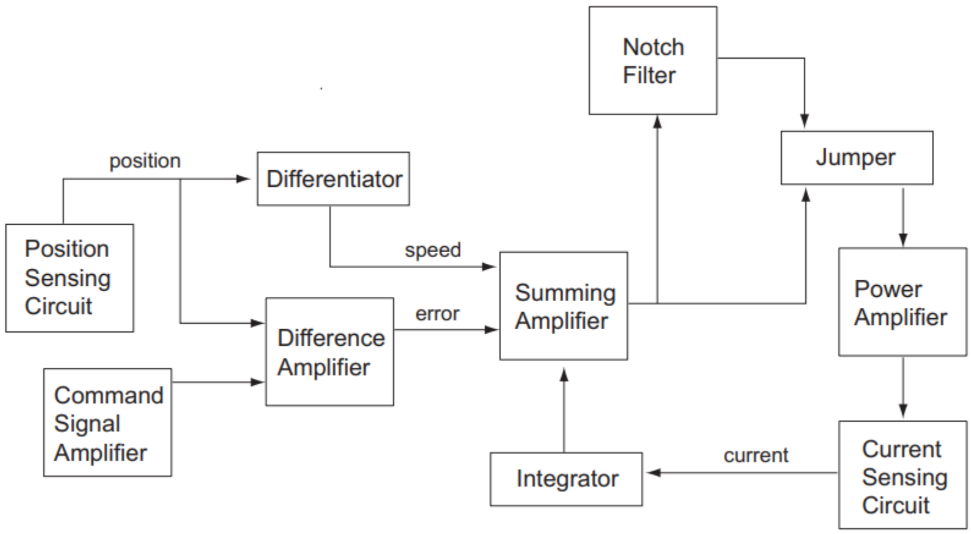


Fig. 8 Servo Board Diagram

Fig. 9 Servo Driver Circuit [14]

4.2 Installation

4.2.1 Mechanical Installation

Two preliminary considerations must be done:

1. Both the motor and the driver card must carry the same serial number. The serial number location is shown in Fig. 10.
2. It is mandatory to mount heatsinks to the driver board otherwise the devices will overheat, and permanent damage may occur.

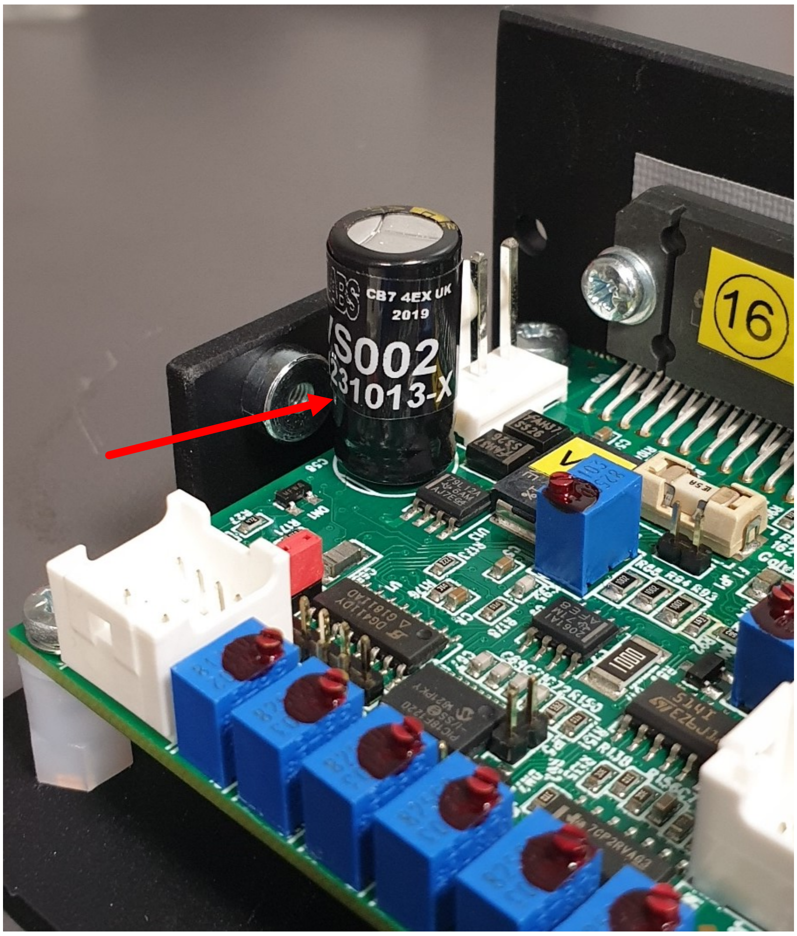


Fig. 10 Serial Number Location

4.2.1.1 Heatsinks

Thermal management is of the greatest importance to the correct operation of the galvo system. The GVS012 implements a heatsink for the servo driver board, that has to be mounted using the two M3 x 8 screw to secure the heatsink bracket to the heat sink as shown in *Fig. 11*.

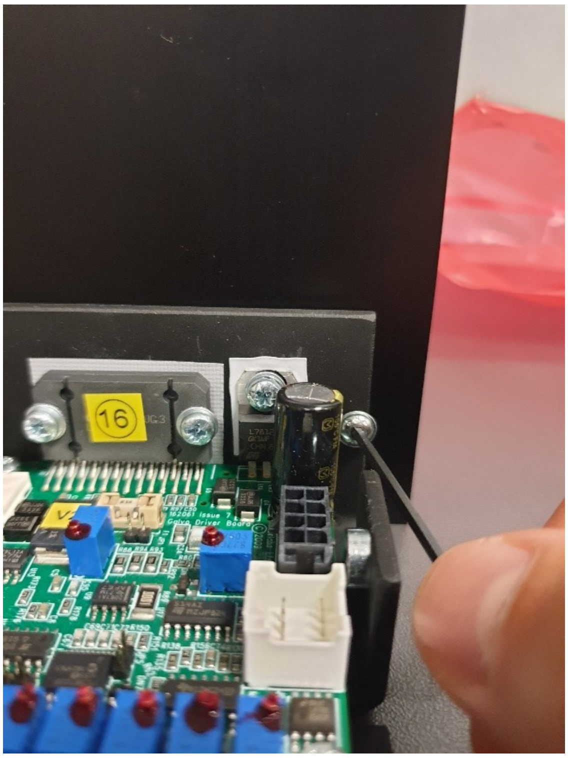


Fig. 11 Servo Driver with Heat Sink

Moreover, the galvo motors must be kept cool, their temperature must be < 50 °C. To ensure the correct working temperature, the galvo system is provided of a mounting bracket able to provide adequate heat sinking for most applications shown in picture 6.

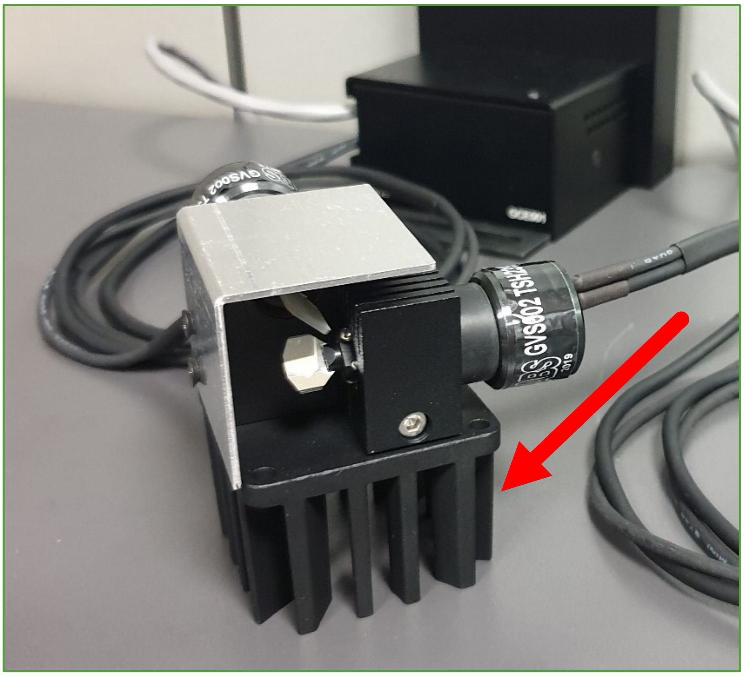


Fig. 12 Galvo Motors with Heatsink (red arrow)

4.2.1.2 System Set Up

The setup used in this project, shown in *Fig. 13*, is obtained by arranging a beam steering system so that the laser beam hits the mirror of the X axis and is therefore reflected on the screen, at right angles to the frame.

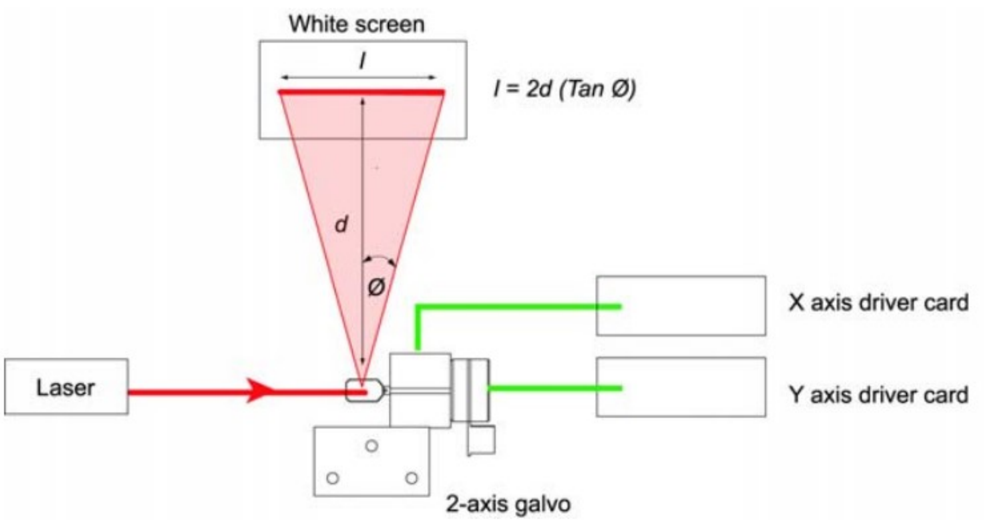


Fig. 13 Beam Steering System [14]

4.2.2 Electrical Installation

4.2.2.1 Power Supply

Thorlabs has been designed a recommended power supply (GPS011) to power the galvo controller boards represented in *Fig. 14*

Fig. 14 GPD011 [14]

However, custom solution can be used but care must be taken to ensure that the voltage and current are with the limits. The galvo controller boards require a split rail DC supply in the range ±15V to ±18V, the maximum current should not exceed 1.2 A rms on each rail and for optimum performance the supply should provide peak currents of up to 5A on each rail. [14]

To ensure that the power supply reservoir capacitors on the board are not damaged by the currents at the power-up of the power supply, it is important to limit the inrush currents.

To guarantee the fulfillment of these requirements, the GPS011 power supply is used.

4.2.2.2 Electrical Connections

J10 is used to connect the driver board to the power supply as shown in *Fig. 15.*

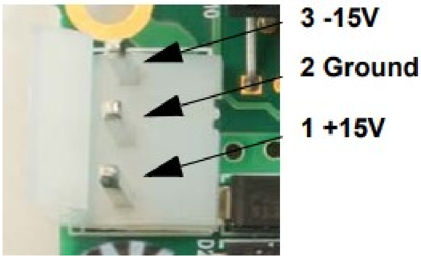


Fig. 15 Power Connection Galvo Driver Board [14]

After connecting the power supply, it is possible to connect a motor cable to the connector J9 as shown in *Fig. 16*.

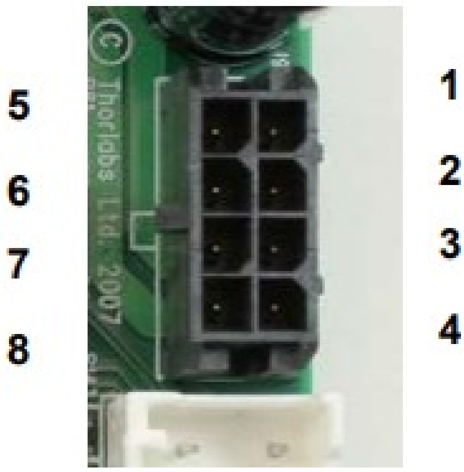


Fig. 16 Motor Connector [14]

1. Position Sensor A Current
2. Position Sensor Ground
3. Position Sensor Cable Shield
4. Drive Cable Shield
5. Position Sensor B Current
6. Position Sensor Power
7. Motor +Coil
8. Motor -Coil

Once the motor is hooked up to the driver board, the command input must be connected to the J7 connector.

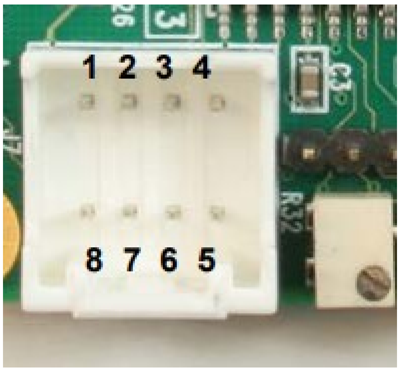


Fig. 17 Command Input Connector [14]

1. Command Input +ve
2. Command input -ve
3. DRV OK
4. External Enable
5. -12V Output
6. +12V Output
7. Ground
8. Ground

Moreover, each driver board is provided of a diagnostic connector. It is used to connect the board to a diagnostic device like an oscilloscope. It is possible to obtain information about the position of the scanner, the positioning error and also the currents flowing in the motor drive and the motor voltage. The detailed description is reported in *Table 5.*

Table 5 Diagnostic Signal Description [14]

Pin	Signal	Description
1	Scanner Position	Signal proportional to the position of the scanner mirror, with a scaling of 0.5V per degree of mechanical movement
2	Internal Command Signal	The command signal following amplification by the input stage
3	Positioning Error x5	Signal proportional to the difference between the demanded and actual positions
4	Motor Drive Current	The drive current of the motor, 2V per A
5	Not Connected	N.C
6	Test Input	N.C
7	Motor + Coil Voltage / 2	It outputs the drive voltage to the positive side of the motor coil. Scale factor = 2.
8	Ground	Ground

Fig. 18 reports the pin identification.

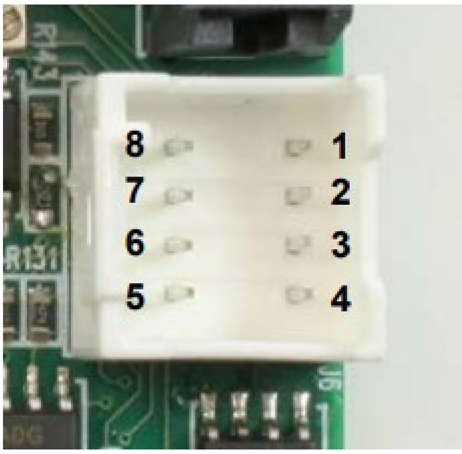


Fig. 18 Diagnostic Connector [14]

4.2.3 Operation with DAC

Once connected as expounded above, the galvo operates correctly. However, it is possible to use a DAC to drive the servo drivers supplied with the galvos. The outputs of the DAC should accomplish some specifications:

- Dual bipolar -10V to 10V DAC analogue output channels
- DAC clocking frequency of 5kS/s per channel
- 16 Bit DAC resolution

In this project a DAC8734 from Texas Instruments is used for this purpose.

4.3 Test

To test the mechanical and electrical installation, a signal generator and an oscilloscope were used.

Initially, each motor driver board was tested independently:

The signal generator was connected to the command input connector as expounded above, while, the oscilloscope was attached to the diagnostic connector pin 3 (Positioning Error x 5) and then to pin number 7 (Motor + Coil Voltage / 2).

The test consists in linearly changing the command input voltage with increment of 0.5V and annotate the difference between the actual and demanded position.

5 DAC8734

The DAC8734 is a high-accuracy, 16-bit R-2R digital-to-analog converter. It accepts supply voltages in range ±5V to ±18V in bipolar output. These characteristics, coupled with less than 1-bit error after calibration, makes this DAC ideal for driving the servo drivers. [15]

The complete list of features is reported in *Table 6*.

Table 6 DAC8734 Features [15]

Bipolar Output	Up to ±18V
Unipolar Output	0V to ±20V
16-Bit Monotonicity	
Relative Accuracy	1 LSB Max
Low zero and gain errors	• Before User Calibration: 4 LSB • After User Calibration: 0.125 LSB Zero Error , 1 LSB Gain Error
Low noise	60nV/√Hz
Setting time	6 μs
Configurable Gain	x 2 / x 4
Analog Output Monitor	
Power-down Mode	
SPI™	Up to 50MHz , 1.8V/3V/5V Logic
Daisy-Chain Mode	
Operating Temperature	-40°C to + 105°C

The internal block diagram of the DAC and a more detail of one channel are reported in the following figures.

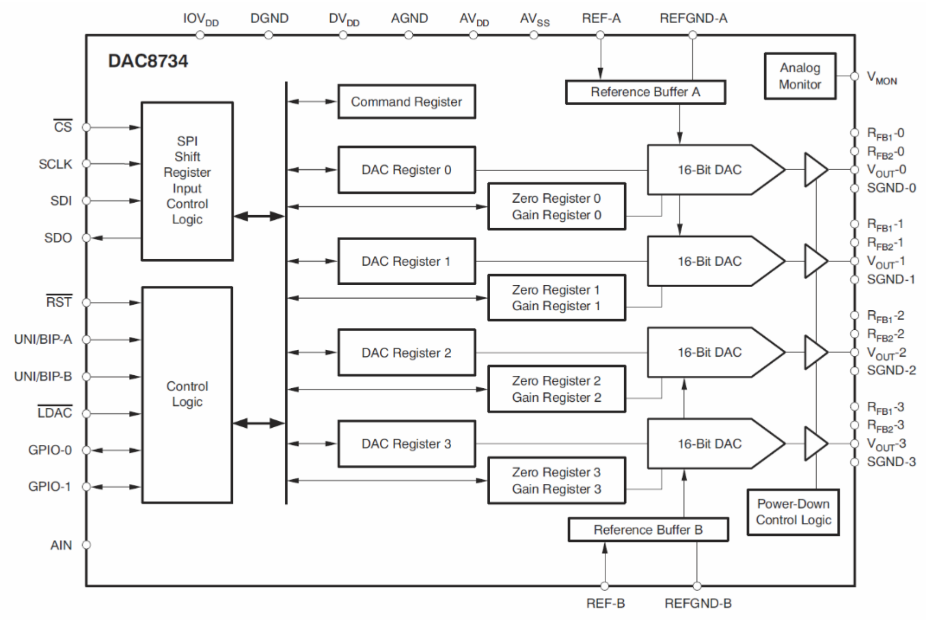


Fig. 19 Block Diagram DAC8734 [15]

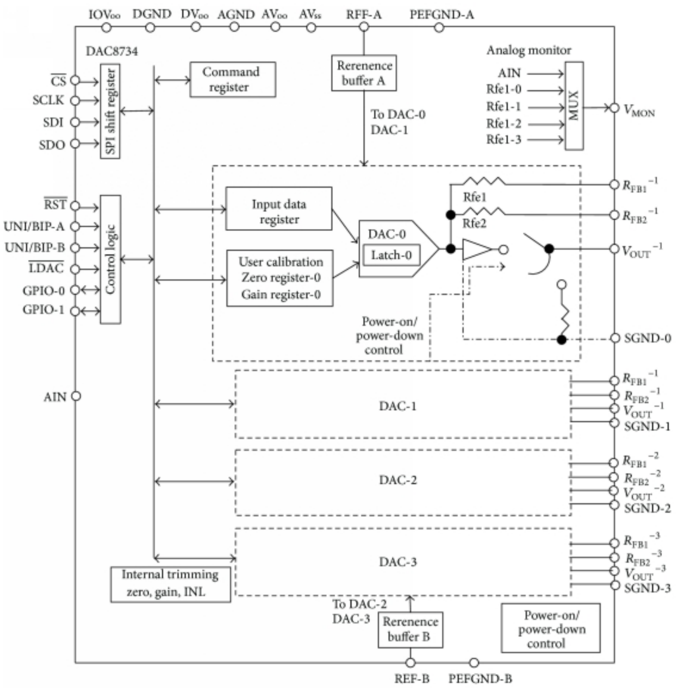


Fig. 20 Detailed Block Diagram DAC8734 [15]

The typical characteristics are reported for bipolar operation in the pictures below

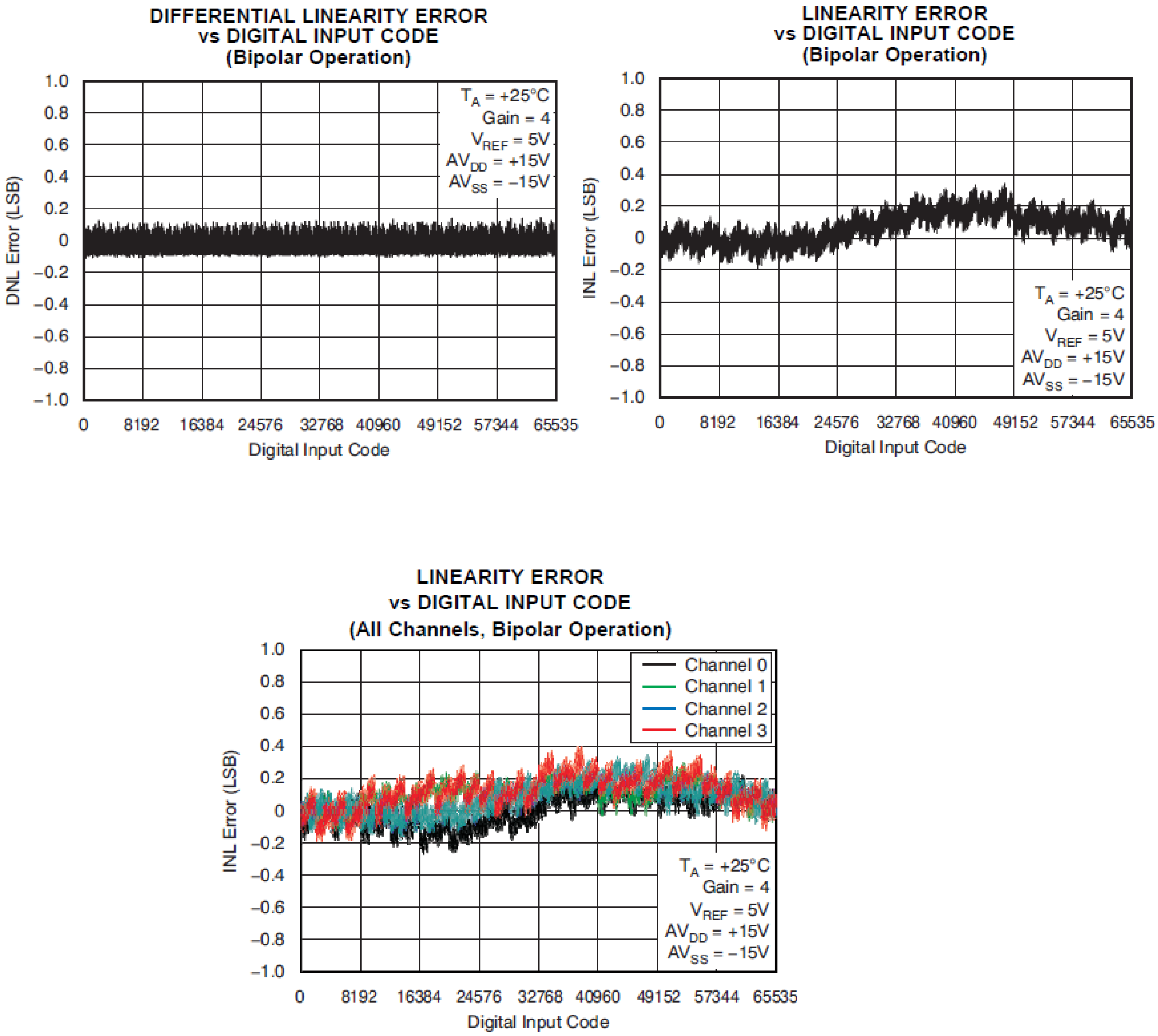


Fig. 21 DAC8734 Main Characteristics [15]

5.1 DAC Architecture

The DAC8734 is a voltage-output DAC, with internal and output buffers. Each channel consists of a R-2R ladder network, shown Fig. 22. The ladder network is driven by the output of the reference buffer; it is designed to allow user tuning, giving the DAC8734 four different output voltage range settings.

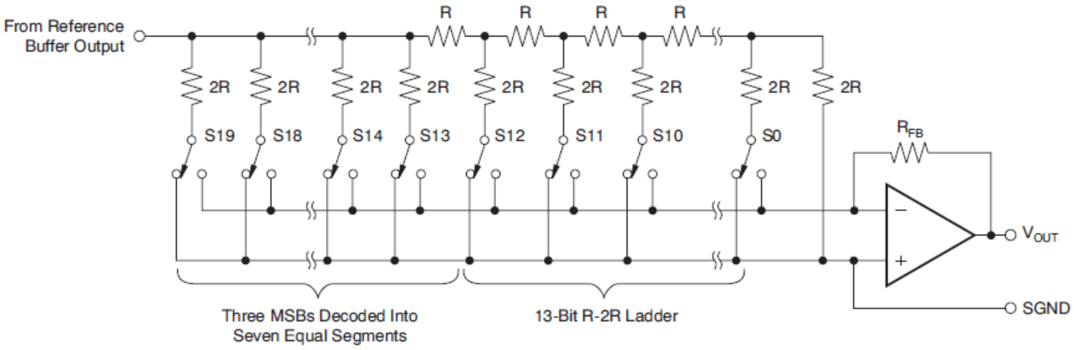


Fig. 22 DAC8734 Ladder Network [15]

The transfer function for the analog outputs for bipolar output is:

$$V_{out} = G_{Gain} \times V_{ref} \times \left(\frac{INPUT\_CODE}{65536} + \frac{ZERO\_CODE}{8 \times 65536} \right) \times \left(1 + \frac{GAIN\_CODE}{65536} \right) \quad (6)$$

Where:

- GAIN is the DAC gain
- INPUT_CODE is the decimal equivalent value of the code written into the DAC input register
- ZERO_CODE is the decimal equivalent value of the code written intro the DAC input register
- GAIN_CODE is the decimal equivalent value of the code written into the Gain Register

The output voltage cannot be greater than $(AV_{DD} - 1{,}0V)$ or less than $(AV_{SS} + 1{,}0V)$ otherwise the output could be saturated.

5.2 Output Range

Each channel of the DAC8734 is provided of an amplifier that implements a bipolar output with gain of 2 or 4. Using 5V as reference, the output range can be configured as ±10 as required by the servo drivers of the galvo system.

To perform correctly the power supply of the DAC must meet some requirements. In particular for bipolar mode $AV_{DD} \geq 2 \times V_{REF} + 1V$ and $AV_{SS} \leq -2 \times V_{REF} - 1V$; for 5V reference in bipolar operation, the supplies must be at least ± 11V.

5.3 Power-Supply Sequencing

The DAC8734 require a through initialization, the digital supplies and logic inputs must be applied before AV_{SS} and AV_{DD}. Moreover, AV_{SS} must be applied before AV_{DD} and REF-x should be applied after AV_{DD}. This initialization sequence ensures that the ESD protection circuit does not turn on.

5.4 Serial Interface

The DAC8734 makes use of a high-speed SPI interface up to 50 MHz to communicate with the Teensy 3.6 using a compatible serial interface.

5.5 DAC8734EVM

Texas Instruments developed an evaluation module that contains all the support circuitry needed for the DAC8734. This module makes available customizable options to the customer and thus, it is a reliable and ready solution to be implemented in the system. The operation mode can be controlled using an onboard switch or using the digital header. [16]

5.5.1 Analog Interface

The analog interface is available on the top of the evaluation module. The board comes with a gain of 4, to change the gain a proper jumper must be installed, and the command register must be update accordingly. To ensure an output voltage of ±10V a gain of 4 is needed, thus no variations are need in this project.

Albeit the DAC8734EVM has an external reference voltage option, the onboard reference was used.

5.5.2 Serial Data Interface

The following table reports the serial interface pins.

Table 7 DAC Serial Interface pins [16]

Pin No.	Signal Name	I/O Type	Pullup	Function
J2.1	Unused	—	—	
J2.2	GPIO-0	In/Out	High	GPIO-0
J2.3 J2.5	SCLK	In	None	DAC8734 SPI clock; pins are shorted together
J2.4	DGND	In/Out	None	Digital ground
J2.6	GPIO-1	In/Out	High	GPIO-1
J2.7 J2.9	$\overline{CS}$	In	None	SPI bus chip select; pins are shorted together
J2.8	Uni/Bip A	In	High	Output mode select of Group A
J2.10	DGND	In/Out	None	Digital ground
J2.11	SDO/SDI	In/Out	None	DAC8734 SPI data in/out
J2.12	Uni/Bip B	In	High	Output mode select of Group B
J2.13	Unused	—	—	
J2.14	RST	In	High	Input register reset
J2.15	$\overline{LDAC}$	In	High	GPIO signal to control LDAC for DAC output latch update
J2.16	Unused	—	—	
J2.17	$\overline{LDAC}$	In	High	Alternate GPIO signal to control LDAC for DAC output latch update
J2.18	DGND	In/Out	None	Digital ground
J2.19	Unused	—	—	
J2.20	Unused	—	—	

(1) Group A contains V_{OUT}-0 and V_{OUT}-1. Group B contains V_{OUT}-2 and V_{OUT}-3.

Care must be taken to the load DAC ($\overline{\text{LDAC}}$) since this controls the way in which the DAC register is updated. There are two way to update the DAC registers:

1. $\overline{\text{LDAC}}$ is tied to ground while $\overline{\text{CS}}$ pin is low, in this case the input registers are immediately updated. At the end of the data transfer into the register, the CS pin is set high. In such a way it updates both the addressed input data register and the DAC latch register. An update of the DAC latch register updates the DAC channel analog output.
2. $\overline{\text{LDAC}}$ is high and $\overline{\text{CS}}$ pin is low, at the end of the data transfer into the register, the CS pin is set high. This does not update the latch register and thus, the output does not change. The analog output is updated if the $\overline{\text{LDAC}}$ is set low or when the LD bit in the Command Register is set to '1' after the input data register is written.

The first approach is implemented in this design.

5.5.3 Power Supplies

The DAC8734 requires more than one power supply to operate correctly; some of them are optional or not strictly required.

The required power supplies are reported in the following table jointly with the requirement itself.

Table 8 DAC Power Supplies [16]

Pin Number	Pin Name	Function
J3.1	+VA	+4.75V to +24V analog supply
J3.2	-VA	-18V to -4.75V analog supply
J3.5	DGND	Digital Ground Input
J3.6	AGND	Analog Ground Input
J3.10	+5VD	+5V

The jumper JP8 allows to select the dc logic voltage for the DAC8734 (IOVDD) between +3.3VD, +1.8VD or +5VD, all referenced to the digital ground.

5.5.4 Reference Voltage

The DAC8734 can handle two different reference sources simultaneously for different output channels. The pins used to control the reference voltages are REFA and REFB. REFA is used as reference for output channels V_{OUT}-0 and V_{OUT}-1, while output channels V_{OUT}-2 and V_{OUT}-3 use REFB [16].

Three reference voltage options are available, 5V (REF5050), 2.5V (REF5025) or external reference voltage, all cherry-pickable using the switch S3 for REFA or switch S2 for REFB as shown in *Fig. 23*.

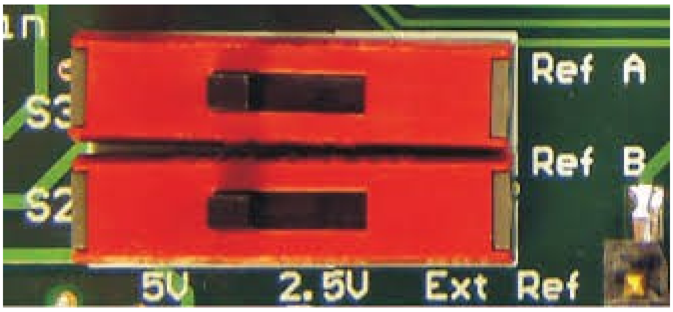


Fig. 23 DAC Reference Voltage Switches [16]

An RC filter is then used to filter these reference voltages before being supplied to the DAC8734.

In this project the onboard 5V reference voltage is used to obtain ±10V.

6 Analog Devices AD7606

To process the feedback from the galvo system, an analog to digital converter must be employed.

The chosen one is the AD7606 developed by Analog Devices. It is a simultaneous sampling, analog to digital data acquisition system with eight channels. Each channel implements an analog input clamp protection, a second order antialiasing filter, a track-and-hold amplifier, a 16-bit redistribution successive approximation analog to digital converter, a reference buffer and a high-speed serial and parallel interfaces [17].

6.1 Analog Input

6.1.1 Analog Input Clamp Protection

Each analog input employs a clamp protection circuitry to manage input over voltage of up to ±16.5 V. *Fig. 24* shows the characteristic of the clamp circuit; it is possible to observe that for input voltages above ±16.5 V the clamp circuitry turns on.

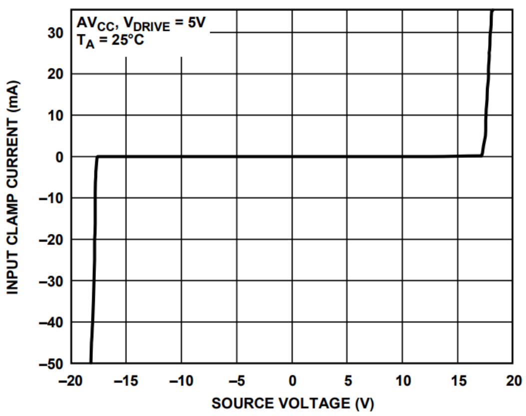


Fig. 24 Input Clamp Protection Characteristic [17]

6.1.2 Analog Input Impedance

The analog input impedance is 1MΩ and it does not vary with the sampling frequency.

This impedance eliminates the need of a driver amplifier in front of the converter, allowing for direct connection to the source [17]. Thus, it is possible to use a single supply instead of a bipolar one.

6.1.3 Analog Input Range

The analog input range is determined by the logic level on the RANGE pin: if this pin is high, the analog input range is ±10V while if it is tied low the range is ±5V [17].

It is important to send a RESET pulse after powering up the AD7606, to ensure the analog inputs has the correct range configuration.

6.1.4 Analog Input Antialiasing Filter

AD7606 implements a second order Butterworth antialiasing filter whose frequency and phase response are reported in *Fig. 25* and Fig. 26.

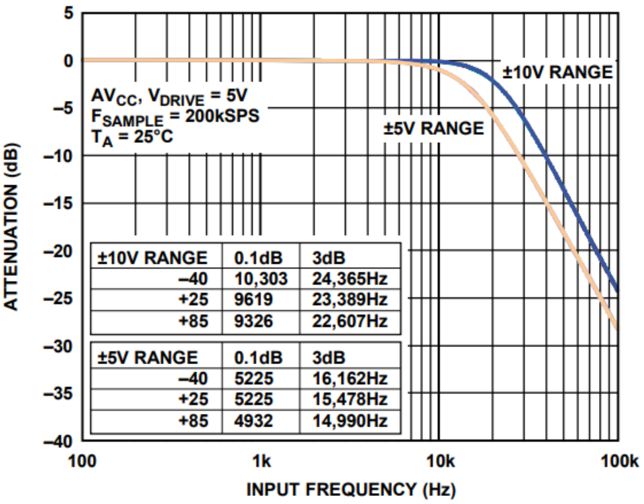


Fig. 25 Analog Antialiasing Filter Frequency Response [17]

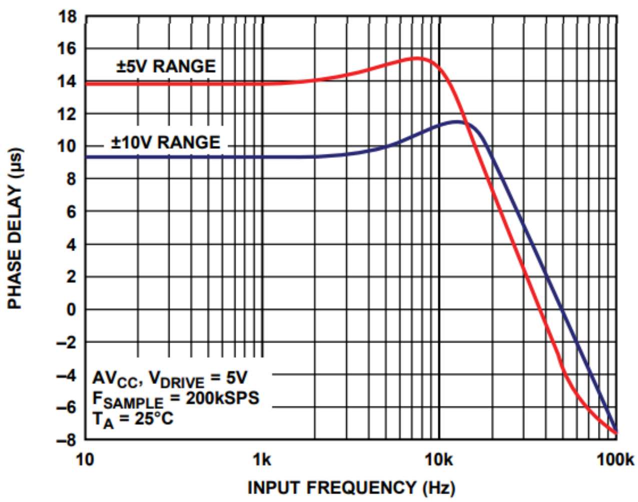
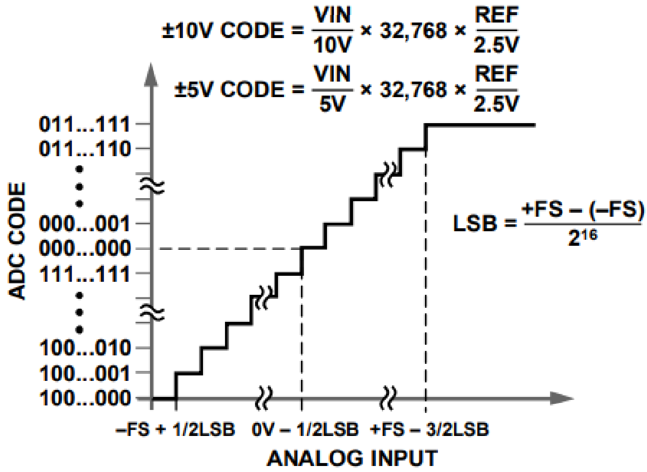


Fig. 26 Analog Antialiasing Filter Phase Response [17]

6.2 ADC Transfer Function

The LSB size depends on the analog input range selected as shown below.



	+FS	MIDSCALE	–FS	LSB
±10V RANGE	+10V	0V	–10V	305µV
±5V RANGE	+5V	0V	–5V	152µV

Fig. 27 AD7606 Transfer Characteristic [17]

6.3 Internal/External Reference

The AD7606 allows the end user to decide whether to use an internal onboard reference voltage or an external one. The internal buffer is used to gain up either the internal or external reference.

The REF SELECT pin allows the user to select between the internal reference or the external one. If this pin is tied high, the internal reference is selected.

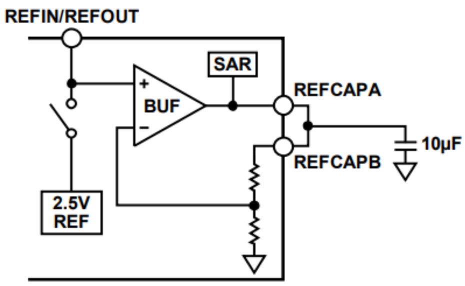


Fig. 28 Internal Reference [17]

For sake of simplicity, in this project the internal reference is used.

6.4 Conversion Control

The AD7606 allows simultaneous sampling of all analog input channels or the channels can be sampled simultaneously in two sets.

6.4.1 Simultaneous Sampling on All Analog Input Channels

To sample all the channels simultaneously both CONVST A and CONVST B pins must be tied together. This approach is exploited in this project since in such a way, it is possible to control both the CONVST pins with a single signal.

In this configuration the BUSY signal indicates the progress of the conversion process. When the rising edge of CONVST is applied, BUSY goes high and return low at the end of the conversion process and so new data can now be read [17].

6.5 Digital Interface

The AD7606 has three interface options selected via the $\overline{\text{PAR}}$ / SER / BYTE and DB15 / BYTE SEL pins as exploited in *Table 9*.

Table 9 AD7606 Digital Interface Options [17]

$\overline{\textbf{PAR}}$ / SER / BYTE	DB15	Interface Mode
0	0	Parallel Interface Mode
1	0	Serial Interface Mode
1	1	Parallel Byte Interface Mode

6.5.1 Serial Interface

To transfer the data, the AD7606 uses the $\overline{CS}$ and SCLK pins. It is possible to use two DOUT lines to access data from the AD7606, on the falling edge of the $\overline{CS}$ the output lines are taken out of the three-state and the MSB of the conversion result is clocked out. This MSB is valid on the first falling edge of the SCLK after the $\overline{CS}$ falling edge. While, the rising edge of the $\overline{CS}$ and of the SCLK clocks the subsequent data bits onto the serial data outputs. Thus, 16 clock cycles are necessary to access each conversion result.

Throughout this project, a single DOUT line is used; the disadvantage of using one line is that the throughput rate is lower if reading occurs after conversion. When exploiting just one line, the other one should be left unconnected.

Fig. 29 shows the timing diagram for reading one channel of data.

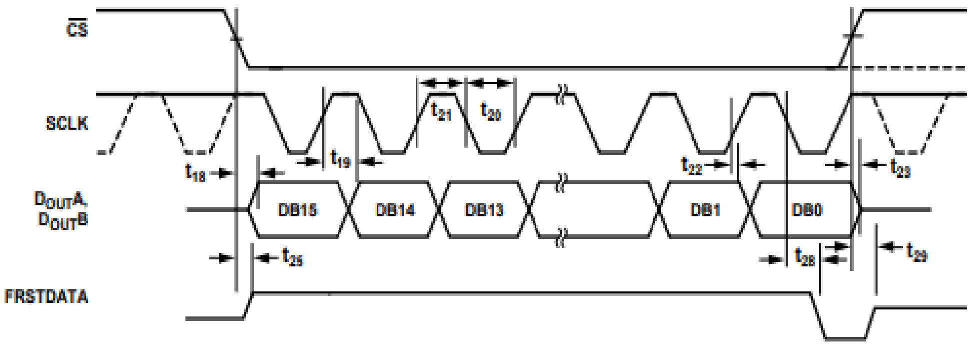


Fig. 29 Timing Diagram SPI [17]

The FRSTDATA pin indicates when the first channel is being read back on the serial interface. Since on the falling edge of the $\overline{CS}$ the MSB is clocked out is, during this phase the FRSTDATA pin is tied high. Eventually, it returns low on the 16th SCLK falling edge after the $\overline{CS}$ falling edge.

7 Receiver Circuit

This circuit implements two TDCs to perform the time measurement.

The following paragraphs contain the design procedures as well as the description of the main components that constitute the receive path.

7.1 Main Components

7.1.1 TDC7200

The TDC7200 is a time-to-digital converter that operates as a stopwatch used to measure the elapsed time between a start event and a stop one.

This device implements an internal base that is used to perform such measurement.

7.1.1.1 Block Diagram

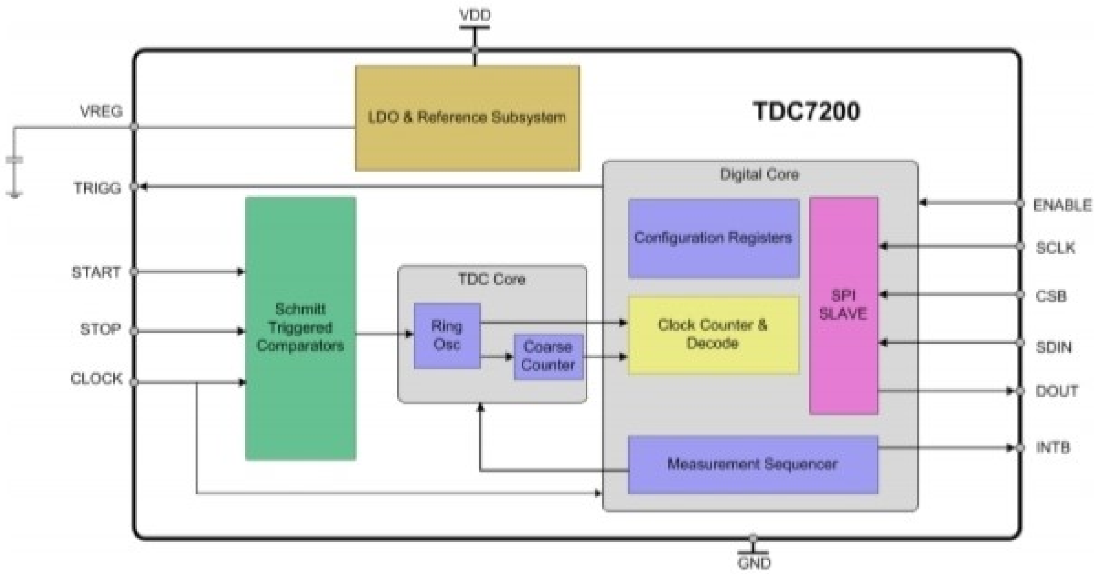


Fig. 30 TDC7200 Internal Block Diagram [18]

7.1.1.2 Feature description

This component is provided of an internal power supply voltage regulator that guarantee that no external circuitry is needed, except for the mandatory decoupling capacitor.

Besides, the TDC7200 needs an external reference clock connected to the clock pin that is used to calibrate the internal time base and by all the digital circuits.

The TDC7200 uses two counters to perform the measurements: The Coarse Counter and the Clock counter. The first counts the number of times the ring oscillator wraps and writes the result into two dedicated registers. The latter counts the number of integer clock cycles between start and stop events, it is used only when measuring times greater than 500ns [18].

The measurement accuracy is, thus, heavily dependent on the external clock accuracy. Moreover, the measurement uncertainty hinge on the clock frequency.

Fig. 31 shows the effect of the clock frequency on the measurement accuracy.

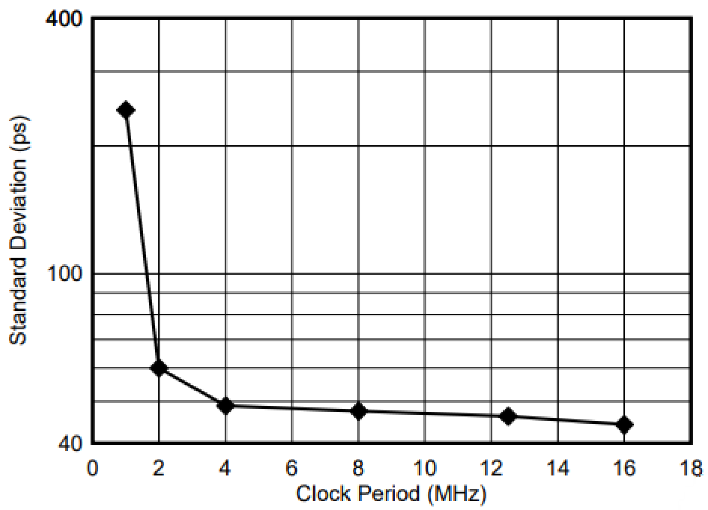


Fig. 31 Standard Deviation vs Accuracy [18]

Using a 1 MHz reference clock, the standard deviation is approximately 293 ps, while with a 16 MHz refence clock, the uncertainty decreases reaching a value slightly above 40 ps.

7.1.1.3 Functional Modes

The device provides two functional models:

1. Measurement Mode 1 ($t_{measure} < 500$ ns)
2. Measurement Mode 2 ($t_{measure} > 500$ ns)

7.1.1.3.1 Measurement Mode 1

In measurement mode 1 the TDC7200 perform the entire counting operation using its internal ring oscillator plus the coarse counter.

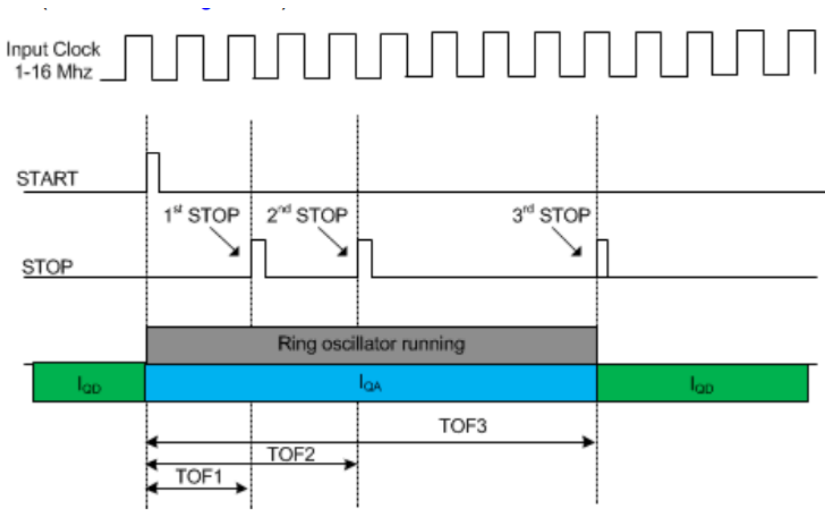


Fig. 32 Measurement Mode 1 [18]

As shown in *Fig. 33* using the measurement mode 1 for acquiring time >500 ns is not recommended since the accuracy decreases significantly.

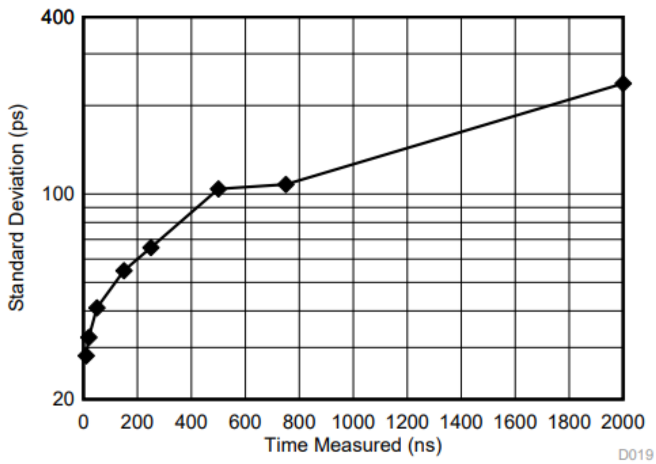


Fig. 33 Measurement Mode 1: Standard Deviation vs Time Measured [18]

7.1.1.3.2 Measurement Mode 2

Fig. 34 shows how the measurement mode 2 work. It is important to notice that the internal ring oscillator is used only to count fractional part of the total measured time, it starts counting from the START signal until the first rising edge of the CLOCK. Then, the clock counter counts the clock cycles of the external clock and the STOP signal is received [18].

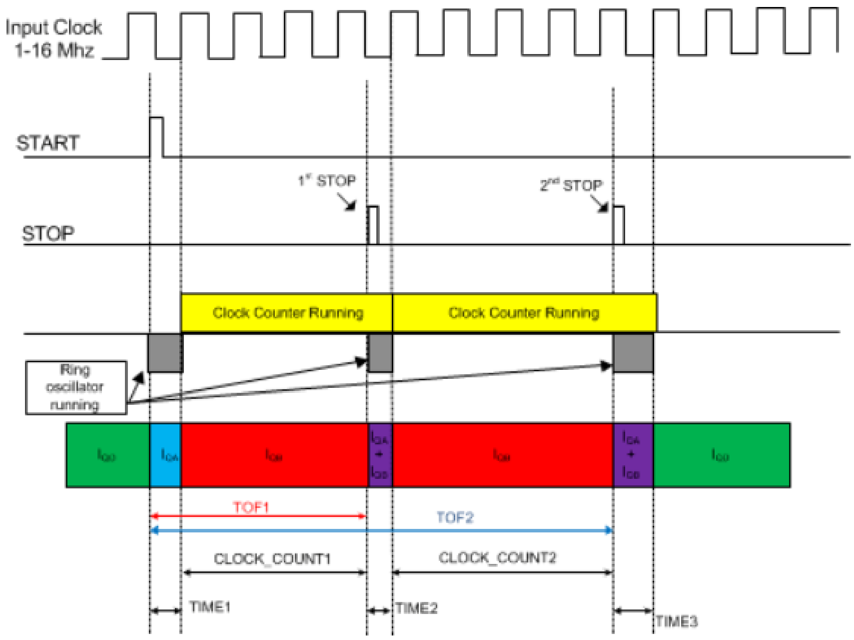


Fig. 34 TDC7200 Measurement Mode 2 [18]

Considering the scope of this project, the most suitable functional mode is the measurement mode 2. The following paragraph explains how it is possible to compute the time-of-flight using this measurement mode.

7.1.1.3.2.1 Measurement Mode 2: Time-of-Flight

$$TOF_n = normLSB(TIME1 - TIME_{n-1}) + (CLOCK_{COUNT_n})(CLOCKperiod) \tag{7}$$

$$normLSB = \frac{CLOCKperiod}{calCount} \tag{8}$$

$$calCount = \frac{CALIBRATION2 - CALIBRATION1}{(CALIBRATION2\_PERIODS) - 1} \tag{9}$$

Where:

- TOF_n = time-of-flight measured from the START to the n^{th} STOP
- TIME1 = time 1 measurement given by the register address 0x10
- $TIME_{(n+1)}$ = (n+1) time measurement, where n =1 to 5
- normLSB [s] = normalized LSB value from calibration
- $CLOCK\_COUNT_n$ = n^{th} clock count, n=1 to 5
- CLOCKperiod = external clock period
- CALIBRATION1 = TDC count for first calibration cycle
- CALIBRATION2 = TDC count for second calibration cycle
- CALIBRATION2_PERIODS = setting for the second calibration; located in register CONFIG2

7.1.1.4 Programming

The TDC7200 communicates with the microcontroller using the serial peripheral interface. It is used to configure the parameters available in the configuration registers and to read the values stored in the register after the conversion.

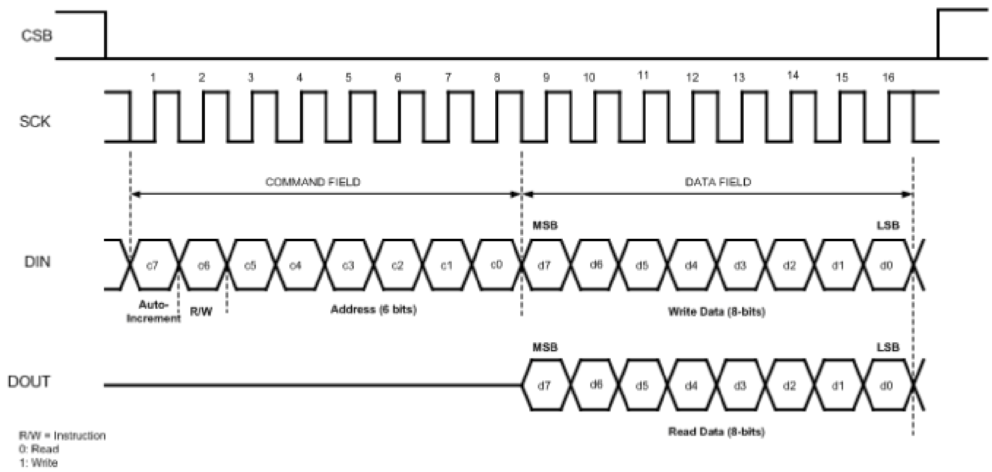


Fig. 35 TDC7200 SPI Protocol [18]

Fig. 35 shows the SPI protocol for a transaction of on byte of data.

7.1.1.5 Power Supply Recommendations

The input voltage should be in range between 2V and 3.6V. The datasheet recommends *"to place a 100nF ceramic bypass capacitor to ground as close as possible to the VDD pin. In addition, an electrolytic or tantalum capacitor with value greater than 1µF is recommended"* [18].

For this reason, this kind of capacitor were implemented in the design of the receiving circuit.

7.1.1.6 Clock Recommendations

To obtain accuracy and reliability, a stable reference clock is needed. The ability to measure time is affected by the accuracy and jitter of the chosen reference clock.

The accuracy refers to the difference between the nominal clock period value and the actual one. For example, a 16MHz clock reference may have a 5ppm accuracy.

The effect of the clock jitter is to introduce uncertainty into a time measurement. The uncertainty is a function of the clock jitter and the number of clock cycle measured: the jitter accumulates on each clock cycle as shown by *Fig. 36.*

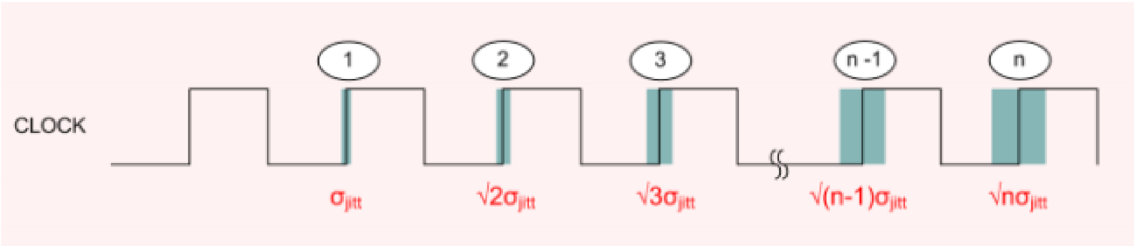


Fig. 36 Clock Jitter [18]

The chosen clock reference is ASTX-H11 described in the following section.

7.1.2 Clock Reference: ASTX-H11

The ASTX-H11 is a 16MHz TCXO oscillator with frequency stability of 2.5ppm. It means that if, for example, the time measured is 50μs, the error over this time period is:

$$\epsilon = t_{period} \times \frac{f_{stability}}{10^6} = 50 \times 10^{-6} \times \frac{2.5}{10^6} = 0.125ns \tag{10}$$

Considering the clock jitter, it is possible to assert that the clock jitter uncertainty introduced in a single measurement can be computed using *Equation 11*.

$$\sigma_{jitter} = \sqrt{n} \times \theta_{jitter} \tag{11}$$

Where:

- n = number of clock cycle counted
- θ_{jitter} = cycle-to-cycle resolution

The cycle-to-cycle jitter is defined in JEDEC Standard 65B as "*the variation in cycle time of a signal between adjacent cycles, over a random samples of adjacent cycle pairs*". Moreover, "*the sample size should be greater or equal to 1000.* " [19] .

By integrating the phase noise over a range of frequencies, is possible to obtain information about the phase jitter, as shown below [20].

$$N = Noise\ Power = \int_{12MHz}^{20\ MHz} \Theta_{noise}(f)df \tag{12}$$

The problem of clock stability and the consequently impact on measurement has been solved by D.W Allan [21]. Albeit the applicability

$$RMS_{phase\ jitter} = \sqrt{10^{\frac{N}{10}}} \times \tag{13}$$

$$RMS_{phase\ jitter}[s] = \frac{RMS_{phase\ jitter}}{2 \times \pi \times f} \tag{14}$$

of this theory is beyond the scope of this project, it represents a suitable source of information about the relevant phenomena that the designer should consider when dealing with clocking systems.

All in all, the ASTX-H11 results a consistent choice for this project, since it guarantees stability and the effect of the clock jitter is minimum.

7.1.3 TLV3502

It is a push-pull output comparator featuring a fast 4.5ns propagation delay. This component is ideal for low-voltage application thanks to the beyond-the-rails input command mode range [22].

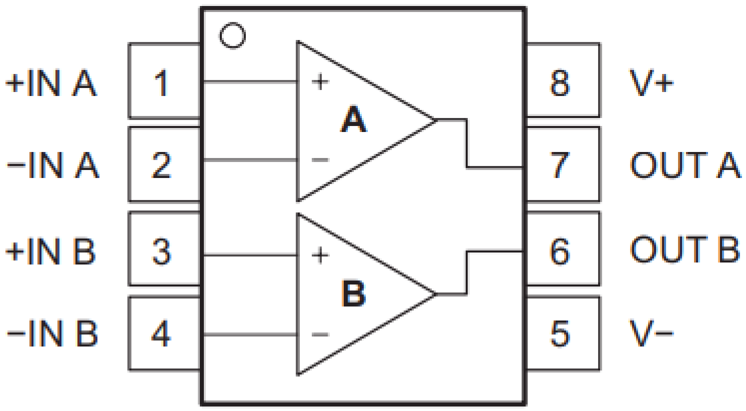


Fig. 37 TLV3502 [22]

Another important feature is that it presents an improved noise immunity including the internal 6 mV hysteresis. It is also possible to add an external hysteresis simply by connecting a small amount of feedback to the positive input as shown in Fig. 38.

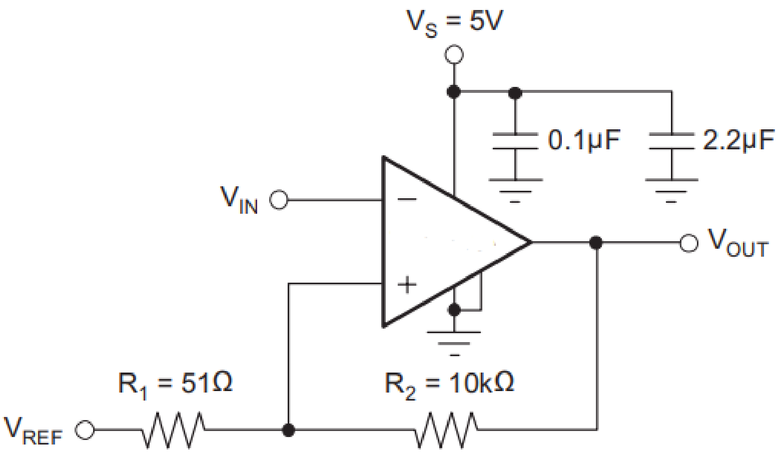


Fig. 38 Adding Hysteresis to the TLV3502 [22]

Moreover, the device inputs are protected by electrostatic discharge diodes that conduct if the input voltages exceed the power supply by mode than 300mV. By limiting the input current to 10 mA, it ensures that the device is tolerant to transitory voltages greater than 300 mV. This limiting can be accomplished with an input resistor in series with the comparator [22].

This kind of comparator is specified for use on a single supply from 2.7 V to 5.5 V. The bypass capacitor must be placed as close as possible to the power-supply pins to reduce noise coupling in from noise or high-impedance power supplies.

Cares must be taken when routing the signal source to the comparator input, since it is necessary to minimize the propagation delay of the complete circuit, the resistance of this route should be minimized.
It is suggested [22] to implement a 2.2 µF tantalum capacitor to buffer the power-supply line against ripple; however, for this specific project, the 2.2 µF capacitor is not considered necessary and then not placed.

7.1.4 SN74LVC1G08-Q1 AND Gate

The SN74LVC1G08-Q1 is a high-drive CMOS device used for implementing AND logic with high output drive. At 3.3V, it can produce up to 24 mA of drive current and thus it is ideal for high speed applications [23].

The application curves are shown in *Fig. 39*

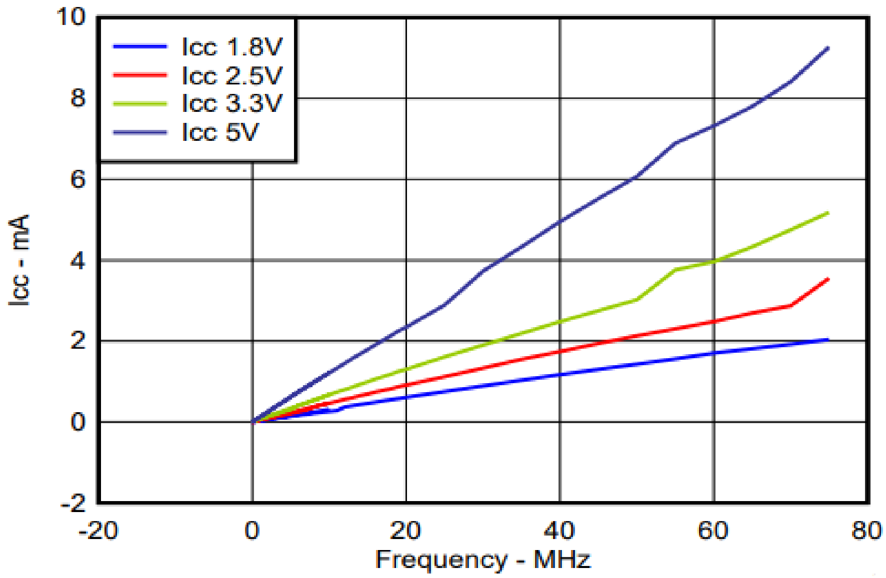


Fig. 39 Icc vs Frequency - SN74LVC1G08 [23]

The device contains one 2-input positive AND gate.

Fig. 40 AND Functional Block

It performs the Boolean function:

$$Y = A \cdot B$$

Or

$$Y = \overline{\bar{A} + \bar{B}}$$

Table 10 resumes the functional modes.

Table 10 AND Function Table

INPUTS		OUTPUT
A	B	Y
H	H	H
L	X	L
X	L	L

Since the high drive also creates fast edges into light loads, routing and load conditions should be taken into account to prevent ringing.

To prevent power disturbances, a bypass capacitor is implemented.

7.2 Design Considerations

7.2.1 Electrical Design

The scheme is quite straightforward: the received signal from the sensing element is amplified and compared to a suitable threshold, the output of the comparator triggers the STOP input of the TDC.

To work with the digital samples coming from the TDC it is necessary to consider the following parameters:

Table 11 TDC Sampling Parameters

PARAMETER	TDC SAMPLING
Conversion trigger	Level Crossing
Analog signal amplitude information	Exact but lower number of steps
Digital Output	Digital image of the time at the instant of crossing the threshold
Main noise characteristics	Jitter

According to "*at the zero crossing of h determine h within a multiplicative factor*" [24]. This means that, considering a bandpass signal, if one catches the instant at which the signal crosses zero, one has all the information to reconstruct the signal except the amplitude [25].

Thus, a TDC based system, should capture rising edges and falling edges when crossing the zero and, thanks to the previous statement, extract the ToF.

7.2.1.1 Electronics

To compute the electronics, it is necessary to estimate the amount of energy that the sensing element will receive as a function of the transmitted power. It is possible to estimate the received energy using the following equation:

$$\mathrm{Pr} = \frac{Gt \times Gr \times Pt \times \pi \times r^2}{2 \times \pi \times d^2} \tag{15}$$

Where:

- Gt = efficiency of the transmitting element
- Gr = efficiency of the receiving element
- r = half-diameter of the optical receiving element
- 2 x π = solid angle at which the light is diffracted by the object
- d = distance between the object and the sensing element

Assuming that:

- Light is perfectly collimated, and the transmitted lights hits entirely the target
- Energy density constant over $2 \times \pi \times d^2$

Then, *Equation* (15)***Errore. L'origine riferimento non è stata trovata.*** becomes:

$$\text{Prx} = \frac{Ptx \times \pi \times r^2}{2 \times \pi \times d^2} \qquad \textbf{(16)}$$

Moreover, the electronics must fulfill these requirements:

1. Recognize the smallest possible amount of energy from the echo from the background noise;
2. Highest possible accuracy on the time measure between the moment when the light pulse is emitted and the time at which it is received.

Related to the first requirement listed above, is the maximum sensing distance. Using TDCs the maximum measuring range is influenced by the TDC's register width. As explained before, the datasheet of the TDC7200 gives the maximum range as a function of the clock fed to it. By analyzing this data, it is clear that the range of the LIDAR design is not affected by the measurement range of the TDC7200 (up to 1379 km at 16 MHz).

The second requirement covers many aspects of the design of such system, in particular it is related to the resolution, accuracy and repeatability of the measure.

- **Resolution**: the resolution of the LIDAR system is the smallest difference between two reported distances. It is determined by the LSB of the TDC.
- **Accuracy**: It is altered by the timing delay introduced by the propagation delays of the devices.

- *Walk error:*

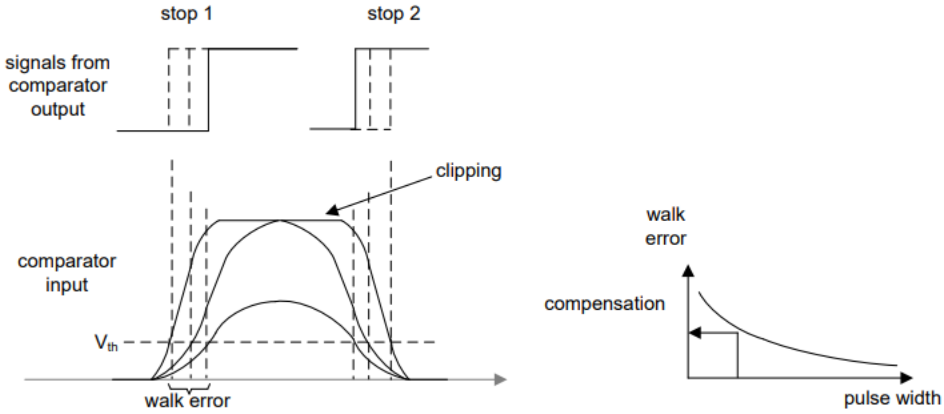


Fig. 41 Walk Error [26]

It is a particular accuracy error resulting from the combination of different parameters:

- Limited slope of the pulse of light;
- Reflecting ratio of the materials;
- Pulse ToF measurement uses a threshold to detect arrival of a signal.

The error in the estimation of the ToF is caused by fact that given the variation of the received energy, the threshold is crossed at different times. Fig. 41 illustrates this event: the comparator input is the threshold voltage V_{th}. The steepest signal passes the threshold at t_1; the second signal, which barely reaches the clipping signal, passes the threshold at t_2 ($t_2 > t_1$); the third signal, very attenuated, passes the threshold at t_3 ($t_3 > t_2 > t_1$). Considering that the target is situated at the same distance, since the signals peak at the same instant, also the distance is the same, hence the walk error is a systematic error.

- **Repeatability**: is the capacity of the system to provide the same result when measuring the same quantity. The jitter, in each its forms, is the source of noise.

7.2.1.2 Signal-to-Noise Ratio

Defined as the ratio of the power of the signal to the power of the background noise.

$$\mathrm{SNR} = \frac{P_{Signal}}{P_{Noise}} \tag{16}$$

In a TDC based system, it is crucial to choose the correct threshold so that the probability of a false positive is acceptable for system performances.

Considering a white Gaussian noise, if the system can accommodate more than 16% of false results, the threshold at the input can be set to the background noise level average plus its RMS value. Furthermore, the time of triggering is equally distributed between 0 and the theoretical ToF.

7.2.2 Basic Optical Considerations

7.2.2.1 Optics

The design uses specific lenses to demonstrate collimation of the laser an focusing of the return signal on the receiving element. These components have been selected to satisfy several design requirements while retaining a simple implementation.

7.2.2.2 Focusing Lens

The main objective of the receiver lens is to collect as much light to the sensing element as possible. This is done employing a focusing lens. The size of the lens affects the amount of light focused on the photodetector.

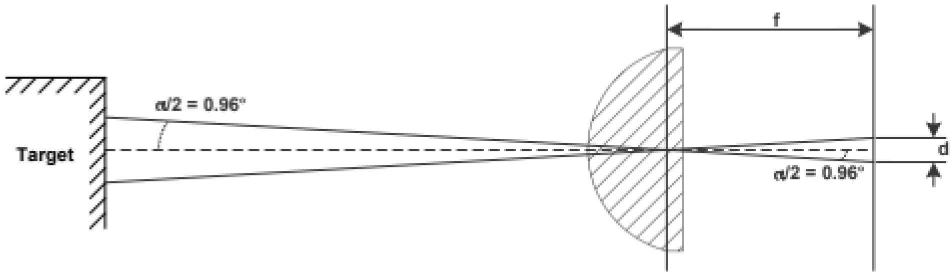


Fig. 42 Focusing Lens [26]

The received power is estimated as explained before, using *Equation(13).* In this project, the atmospheric attenuation is not considered since it is negligible for the distances considered and so the approximation that no power is lost on the way to the target is valid . The receiver device includes an optical bandpass filter to reduce noise from ambient light.

For optimum performance at long distance, the detector has to be placed in the focal point, *f*, of the lens. Measuring long distances is the worst scenario, since the received power decreases with the square of the distance, and thus, is the most important consideration when placing the focusing lens.

7.2.2.3 Collimating Lens

Regarding the emitting element, the laser output light needs to be collimated to produce a concentrated beam. Depending on the application, many solutions are available, but in this project a lens is used.

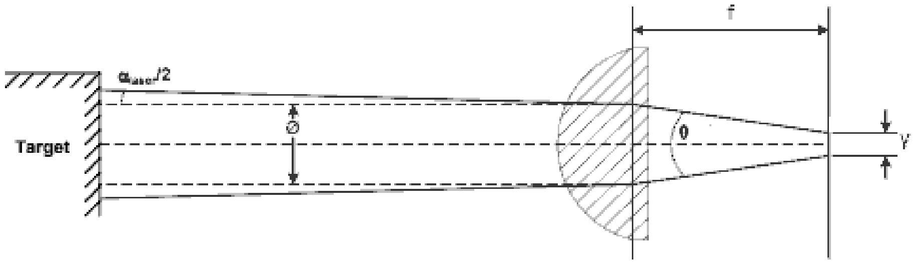


Fig. 43 Collimating Lens [26]

Fig. 43 shows the collimation process.

When dealing with lenses, a suitable threshold between beam width and divergence, must be reached; *Equation* (17) and *Equation* (18) show it:

$$\Phi_{Laser} = 2 \times f \times \tan\left(\frac{\theta}{2}\right) + y \tag{17}$$

$$\alpha_{laser} = 2 \times \operatorname{atan}\left(\frac{y}{2 \times f}\right) \tag{18}$$

8 Receiver Circuit: PCB design

Since the PCB design was an integral part of this project, this section highlights how the PCB is designed and laid out.

The circuit is directly printed without prototyping on a breadboard because this kind of prototype would not suffice with signal integrity, considering the small signal levels obtained during the simulation, as expounded in the next pages.

The schematic and PCB layout were carried out using Altium Designer.

8.1 Schematic

For the circuit design, the components described in the previous chapter were added and connected as shown in *Plate 1.*

All the consulted datasheets advice the use of specific decoupling capacitors and thus, they were implement following these guidelines.

8.2 Board Layout

In this project a two-layer PCB with 35 μm copper and FR-4 substrate with dielectric constant of 4 was used.

Three male SMA connector are used to connect the board to the external devices providing the START and STOP signals to the TDCs and the signal from the laser driver used to set the threshold to detect the emission of a laser pulse.

The components are placed in a logic order trying to minimize the trace length and maintaining the symmetry where possible.

Moreover, two connectors are implemented to program the TDCs and to exchange the data.

The decoupling capacitors are placed as close as possible to the components they are intended to decouple. To minimize reflections 50 Ω traces are use wherever possible and three 50 Ω resistor are implemented near the SMA connectors to mate the impedance of the transmission lines.

Ground plane with 0.25 mm trace insolation is added on the bottom layer only. The ground plane is maintained unbroken beneath the clock traces while some traces are placed on the bottom layer, where the integrity of the plane is not mandatory, to easily connect the components.

The top layer is used for components and routing. Care must be taken when routing the clock signal, since matching the length of each branch is required. The same consideration must be taken into account when routing the START and STOP signals.

Vias are used to connect the top and bottom layer especially around the components that are connected to ground.

The Top Layer Layout is shown in *Fig. 44* while the Bottom Layer Layout, in *Fig. 45*.

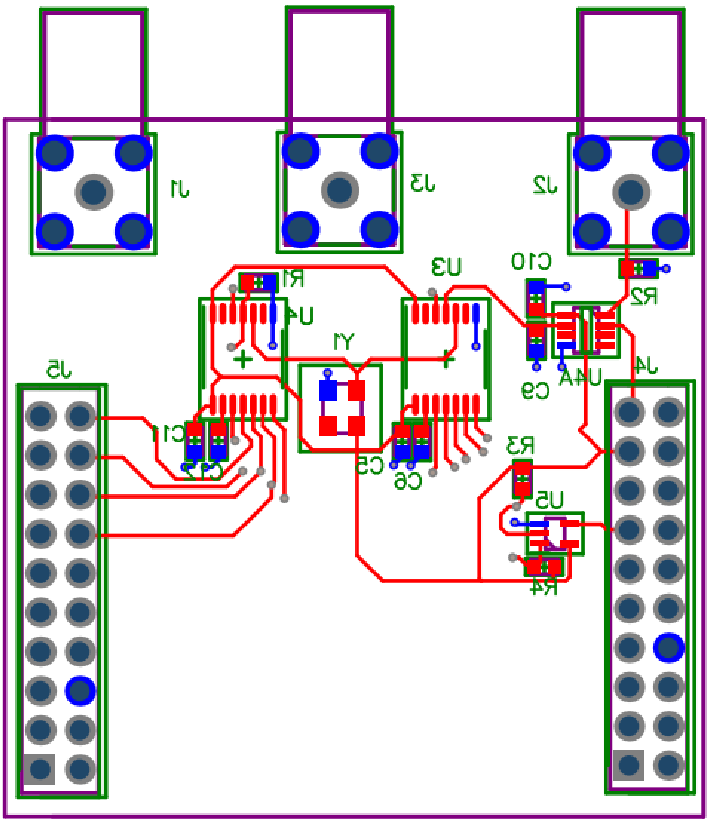


Fig. 44 Top Layer Layout

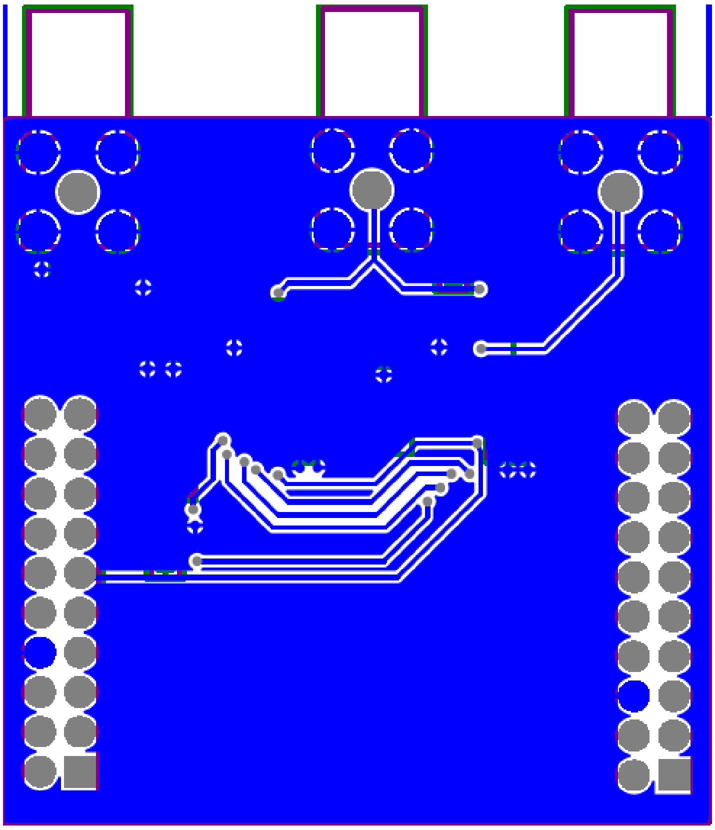


Fig. 45 Bottom Layer Layout

8.3 PCB Realization and Mounting

The PCB is printed by JLCPCB according to the gerber files generated by means of Altium.

The components are then hand soldered onto the PCB according to the schematic. The soldering process were carried out by the author in the LINKS workshops.

The final result is shown in Fig. 46 and in *Fig. 47*.

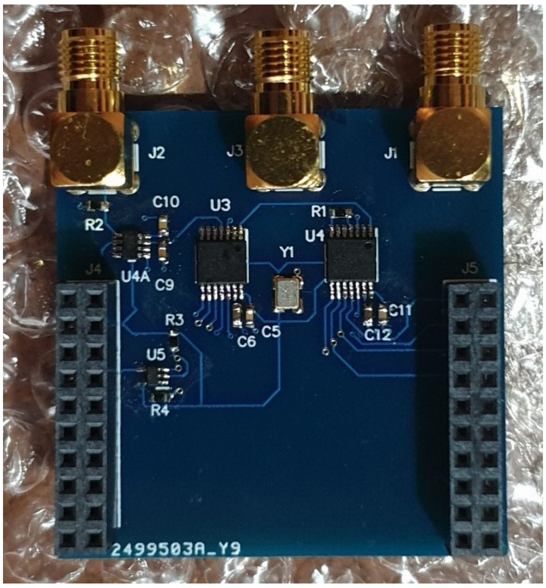


Fig. 46 Soldered PCB, Top Layer

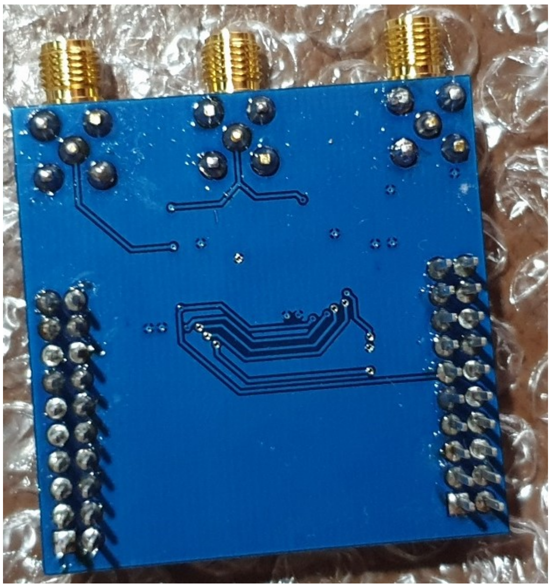


Fig. 47 Soldered PCB, Bottom Layer

9 Experimental Details

In this section the measurement setups used for performing the experiments on the receiver system is presented.

9.1 Signal Generation with TB-S5D5

Some basic functions of the realized circuit needed to be tested to verify the correct behavior and to ensure the correct communication with the microcontroller, before the measurement campaign.

To do that, a Renesas TB-S5D5 board was programmed to output digital signals which act as the start and stop signals for the LIDAR receiver circuit. The configuration is shown in *Fig. 48.*

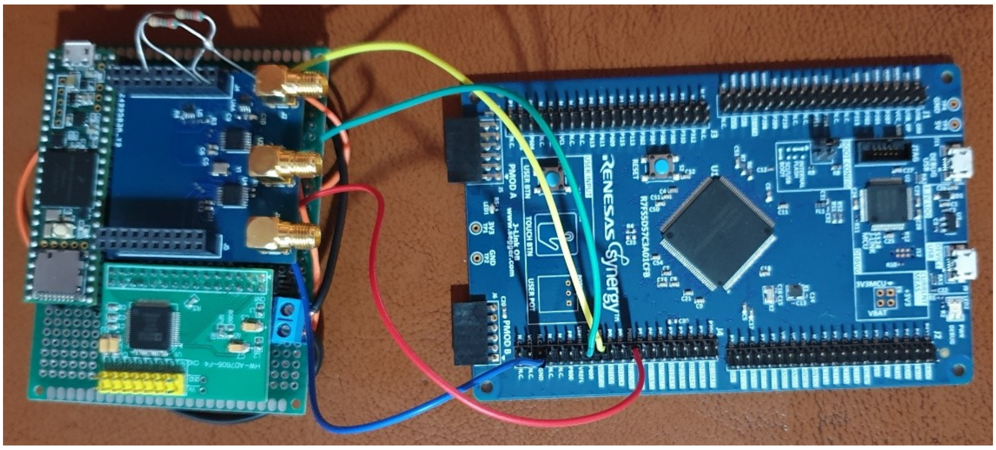


Fig. 48 Test Configuration with TB-S5D5

This allowed to verify the correct communication with the Teensy 3.6 and to ensure that all the configuration registers of the TDC7200s can be accessed correctly. Moreover, it was possible to obtain some information about the measuring system and the performance of the overall circuit.

All the possible configurations were tested by writing the dedicated registers and then, reading the same registers via the SPI protocol, it was possible to compare the values stored in the registers with the desired ones.

9.2 Measurement Setup

Two different measurement campaigns had been performed: one in the electronics laboratory in which the start and stop signals were obtained using a signal generator; the second one in the photonics laboratory, in this case the start and TX stop signal were generated by the laser while the RX stop by the photodetector.

9.2.1 Measuring Setup with Signal Generator

This laboratory activity was fundamental to establish the characteristics of the circuit and to assess which external factors could affect the measure.

Particular attentions have been aimed to determine the delay introduced by the comparator. To measure it, the signal generator was used to provide the star and stop signals to both the TDC7200s and then using a SMA cable, the stop signal was provided firstly to the branch in which there were not the comparator and then to the branch which had the comparator on it. The difference between the elapsed time between the star and stop signal measured on each branch was the delay introduced on the line that presents the comparator.

Once the delay between the two lines have been determined it was possible to perform some measures. To simulate the distance from the target, a coaxial cable was used as delay line. Since the delay introduced by the coaxial cable was well known, it was possible to evaluate the precision of the measure, its reliability and repeatability.

9.2.2 Measuring Setup with Laser and Photodetector

In this phase the test was foreseen with start and stop signals generated by a laser and by the photodetector to simulate the real behavior. The tests should have been carried out in March, but unfortunately, due to the current emergency health situation, this was not possible. Further tests of this type should be carried out in order to ascertain the correct behavior of the system when stressed by optical signals.

10 Results

This section resumes the results obtained during the measurement campaigns.

10.1 Signal Generator Setup

Different experiments, with the coaxial cable inserted, were performed to evaluate the circuit and the accuracy of the measurement.

In this section the different experiments are analyzed, and the results are shown.

10.1.1 The effect of the rising edge

This experiment was intended to evaluate the effect of the rising time of the start and stop signals on the accuracy of the measurement.

To do that the rising time was changed from 8ns to 20 ns, while the frequency and the amplitude remained constant.

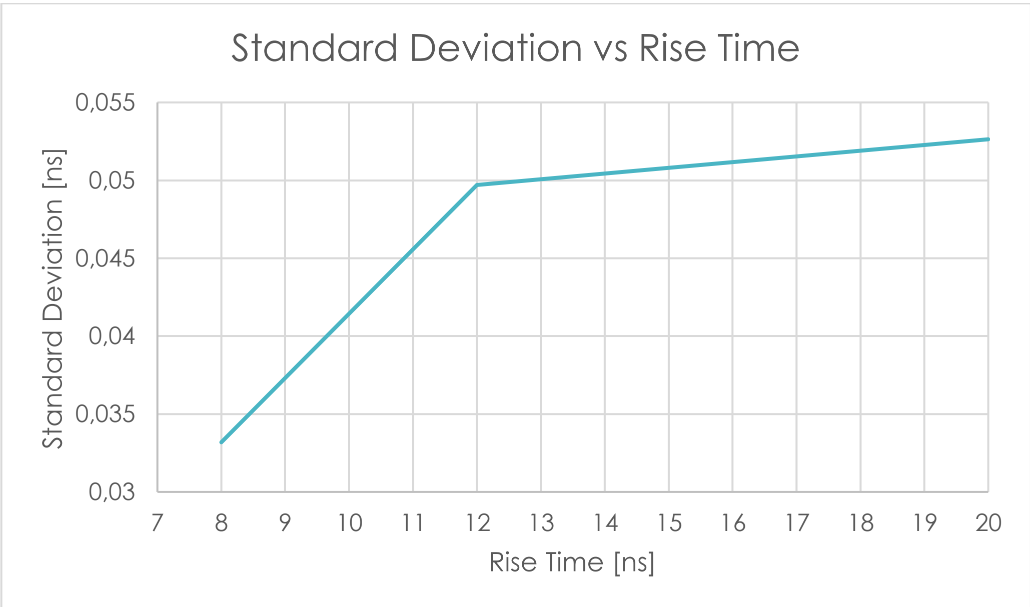


Fig. 49 Standard Deviation as Function of the Rise Time

10.1.2 Measurement Accuracy with T_{RISE} = 8 ns at F = 10 MHz

To evaluate the accuracy and the repeatability of the measurement, the same distance was measured many times and then the obtained data were analyzed using Excel.

In this experiment the rise time of the start and stop impulses was set to T_{RISE} = 8 ns and the frequency was F = 10 MHz.

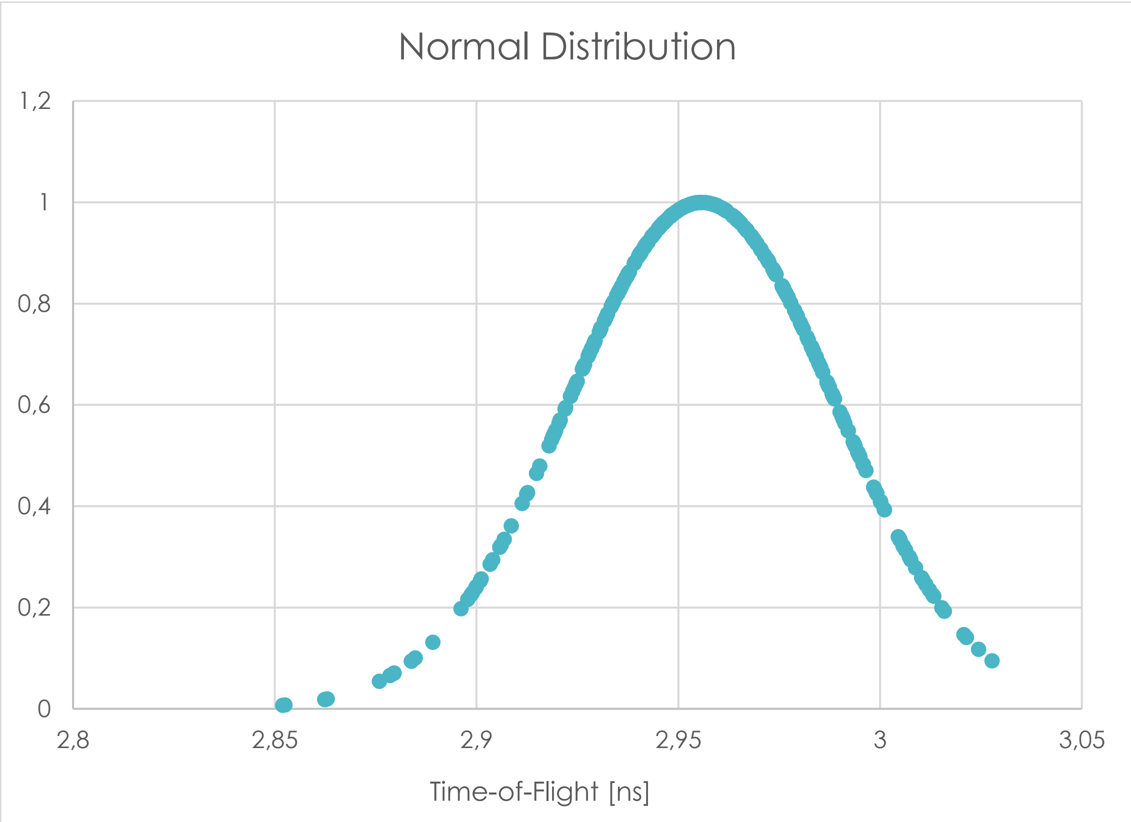


Fig. 50 Normal Distribution - T = 8 ns - F = 10 MHz

Table 12 Statistics T = 8 ns - F = 10 Mhz

Statistics	
Mean	2.955862
Standard Error	0.001867
Median	2.95585
Mode	2.9842
Standard Deviation	0.033185
Sample Variance	0.001101
Kurtosis	0.2057
Skewness	-0.44524
Range	0.1758
Minimum	2.852
Maximum	3.0278

The expected Time-of-Flight was:

$$ToF = 3\ ns$$

That means a distance of:

$$d = 300000000 * 3 * 10^{-9} = 0.9\ m = 90\ cm$$

With the obtained data it is possible to assert that a single measure will be in the range:

$$d_{measured} = \bar{d} \pm 3 \times \sigma = 0.89 \pm 0.03\ [m]$$

10.1.3 Measurement Accuracy with T_{RISE} = 12 ns at F = 10 Mhz

A huge number of measurements of the same distance was acquired and analyzed.

In this case the rise time of the start and stop signals was set to T_{RISE} = 12 ns and the frequency of the impulses was F = 10 MHz.

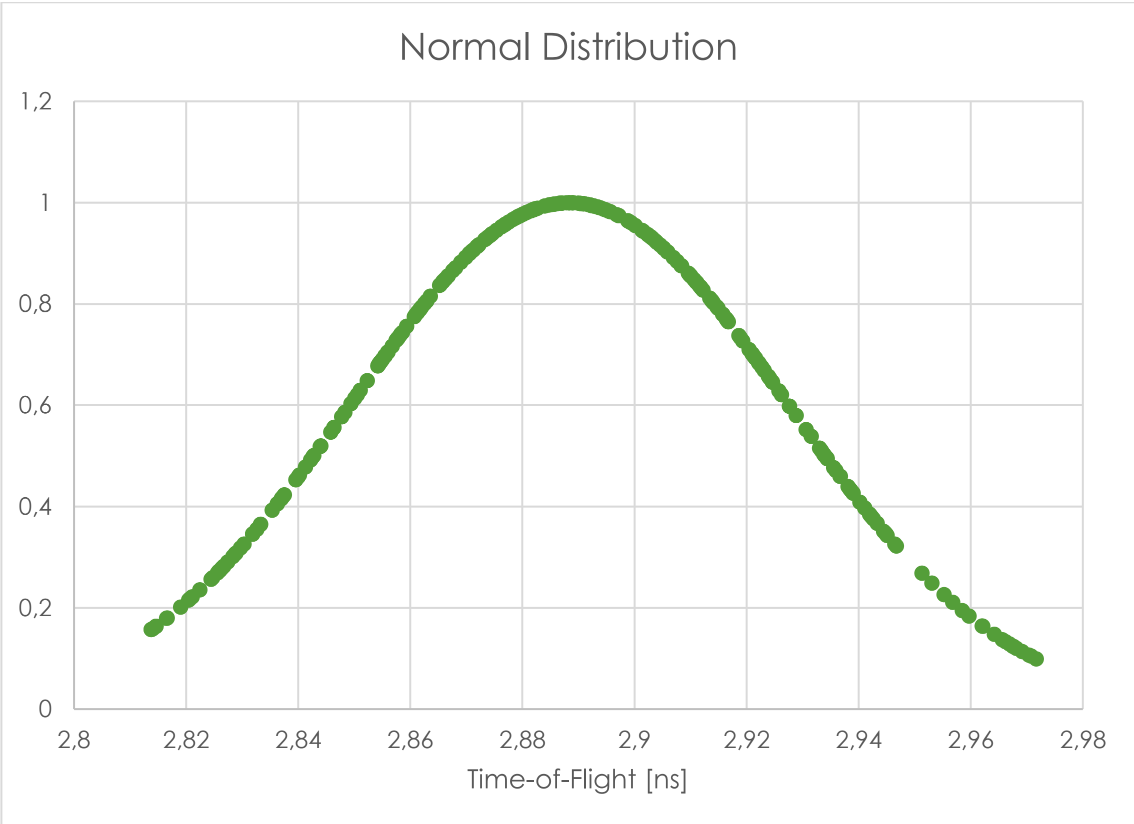


Fig. 51 Normal Distribution - T = 12 ns - F = 10 MHz

Table 13 Statistics T = 12 ns - F = 10 Mhz

Statistics	
Mean	2.888389
Standard Error	0.002182
Median	2.888362
Mode	#N/D
Standard Deviation	0.038783
Sample Variance	0.001504
Kurtosis	-0.66165
Skewness	0.110226
Range	0.157926
Minimum	2.813793
Maximum	2.971719

Since the measurement conditions was the same of the previous experiment, the expected ToF was 3 ns.

In these conditions, a single measurement will lay in the range:

$$d_{measured,12} = \bar{d} \pm 3 \times \sigma = 0.87 \pm 0.045\ [m]$$

10.1.4 Measurement Accuracy with T_{RISE} = 20 ns at F = 10 Mhz

The setup was exactly the same of the other two experiment expounded before. The data were obtained by the same procedure of the previous experiment and the data were analyzed using Excel.

The rise time of the start and stop signals were set to T_{RISE} = 20 ns and the frequency was F = 10 MHz

The acquired data produced the normal distribution shown in *Fig. 52.*

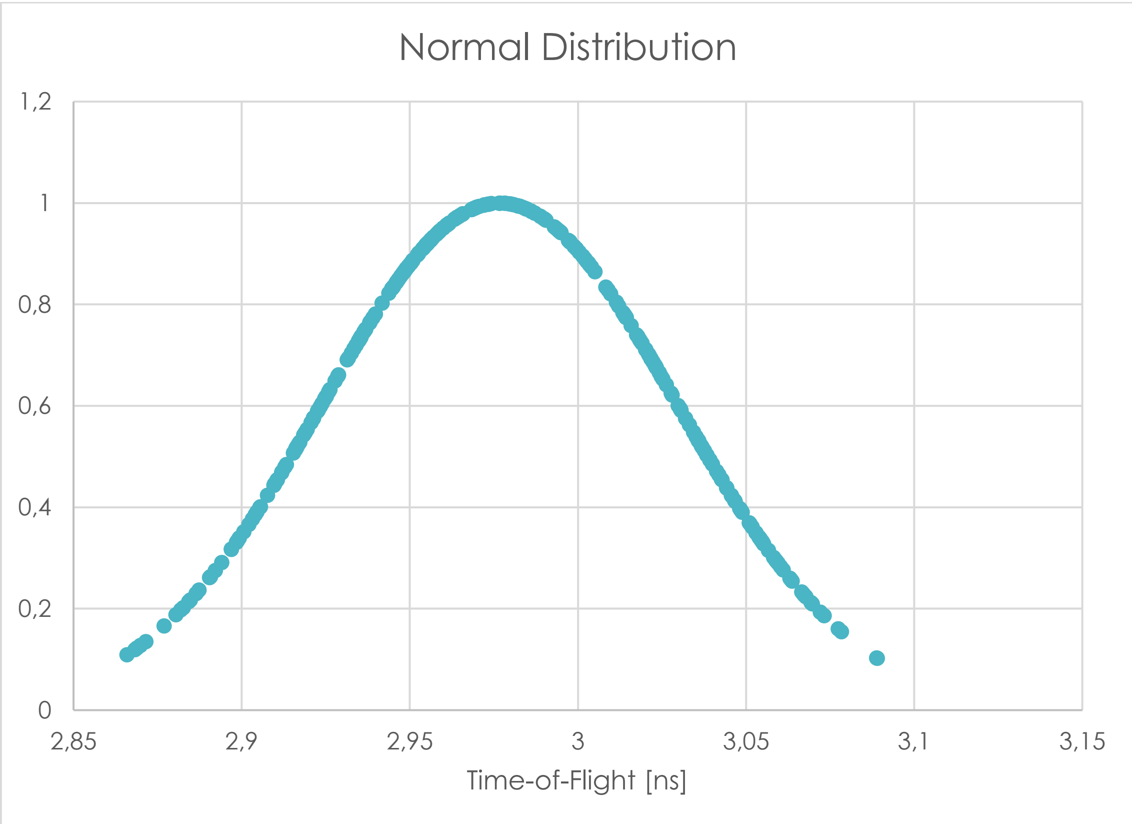


Fig. 52 Normal Distribution - T = 20 ns - F = 10 MHz

Table 14 Statistics T = 20 ns - F = 10 Mhz

Statistics	
Mean	2.976712
Standard Error	0.002961
Median	2.970328
Mode	#N/D
Standard Deviation	0.052635
Sample Variance	0.00277
Kurtosis	-0.89769
Skewness	0.063727
Range	0.223156
Minimum	2.865941
Maximum	3.089097

Once again, the expected ToF was 3 ns. In this configuration a single measurement will be in the range:

$$d_{measured,20} = \bar{d} \pm 3 \times \sigma = 0.89 \pm 0.047 \; [m]$$

10.1.5 The effect of the frequency: T_{RISE} = 8 ns, F=1 MHz

To investigate the effect of the frequency on the measurement accuracy, the rise time of the start and stop signals was set to T_{RISE} = 8 ns and the frequency was changed to F = 1 MHz.

The data have been obtained by the same procedure of the other experiments and the same analysis were performed using Excel.

The data produced the distribution represented in *Fig. 53*

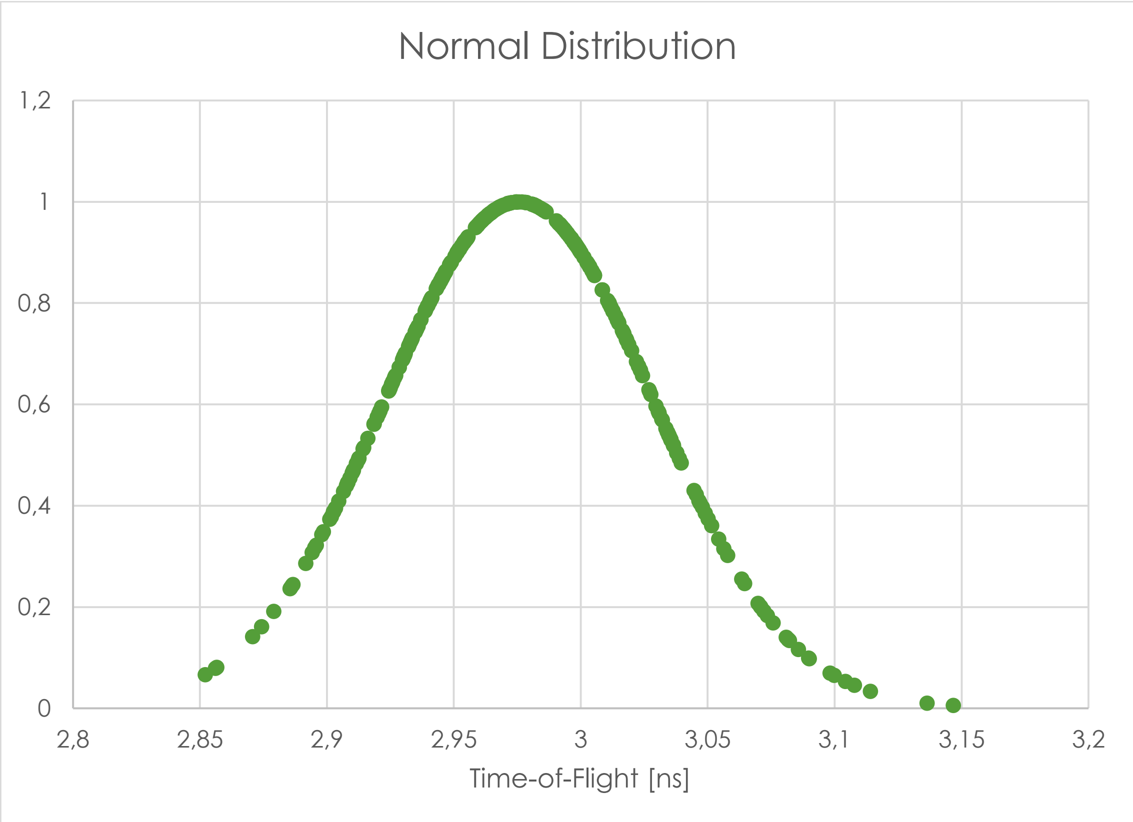


Fig. 53 Normal Distribution - T = 8 ns - F = 1 MHz

Table 15 Statistics T = 8 ns - F = 1 Mhz

Statistics	
Mean	2.975722
Standard Error	0.002987
Median	2.968164
Mode	#N/D
Standard Deviation	0.053098
Sample Variance	0.002819
Kurtosis	0.015591
Skewness	0.438259
Range	0.294658
Minimum	2.852165
Maximum	3.146823
Mean	940.3281

The expected ToF was 3 ns corresponding to a distance of:

$$d = 3 \times 10^{-9} * 3 \times 10^{8} = 0.9\ m$$

Performing a single measurement with this configuration will provide a result in range:

$$\begin{aligned} d_{measured,8ns,1MHz} &= \bar{d} \pm 3 \times \sigma \\ &= 2.9757 \times 10^{-9} * 3 \times 10^{8} \pm 3 * 0.0531 \times 10^{-9} * 3 \times 10^{8} \\ &= 0.8927 \pm 0.04779\ m \end{aligned}$$

10.1.6 The effect of the distance on the accuracy

To evaluate how much the accuracy of the measurement changed with the measured distance, different simulated delays have been generated. Then the acquired was processed with Excel to compute the standard deviation.

Fig. 54 shows how the standard deviation changes with the time-of-flight.

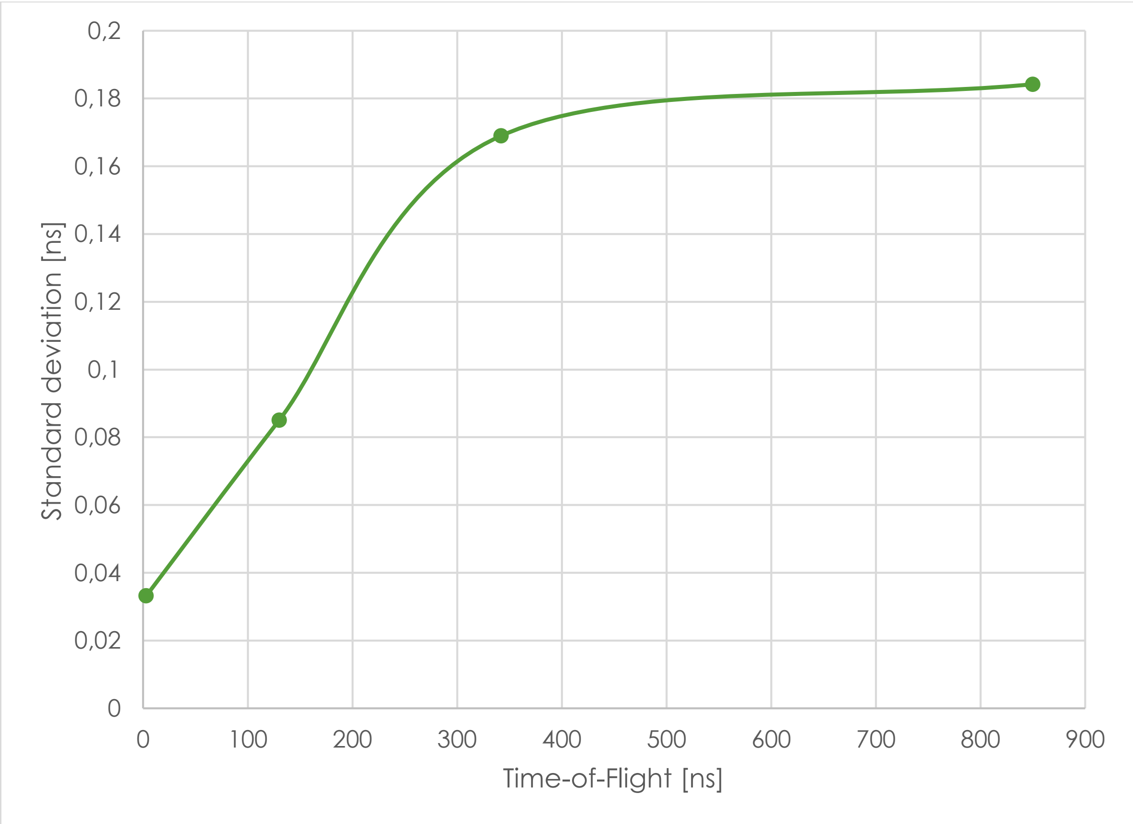


Fig. 54 Standard Deviation VS Time of Flight

In *Fig. 55* the measured distance is shown with the standard deviation expressed in meters.

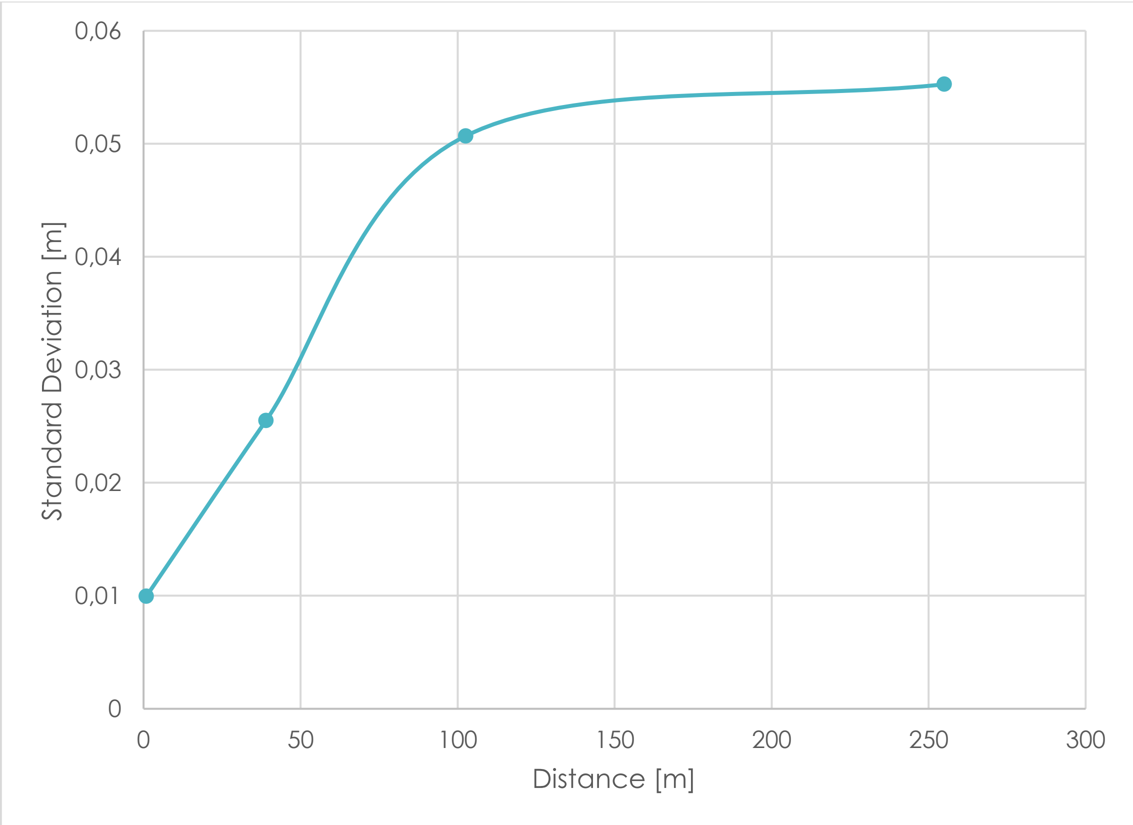


Fig. 55 Standard Deviation [m] VS Distance [m]

10.1.7 Averaging VS Single Measure

The TDC7200 implements a multi-stop mode in which the IC performs a series of measurements, up to 5, of the same ToF and will send the interrupt to the MCU only after the series has been completed. Then the average of that series of measurements is computed and the result represents the measured ToF. This method helps to increase the accuracy and repeatability of the measurements. All the experiment presented above were accomplished employing this mode, however for the sake of completeness one test was performed in order to compare the result obtained with the multi-stop mode with the one obtained with a single stop measurement.

The experimental setup was the same as for the previous experiment, the only change was in the configuration register of the two TDC7200s, that for this test was configured to waits for 5 stop signals before compute the average.

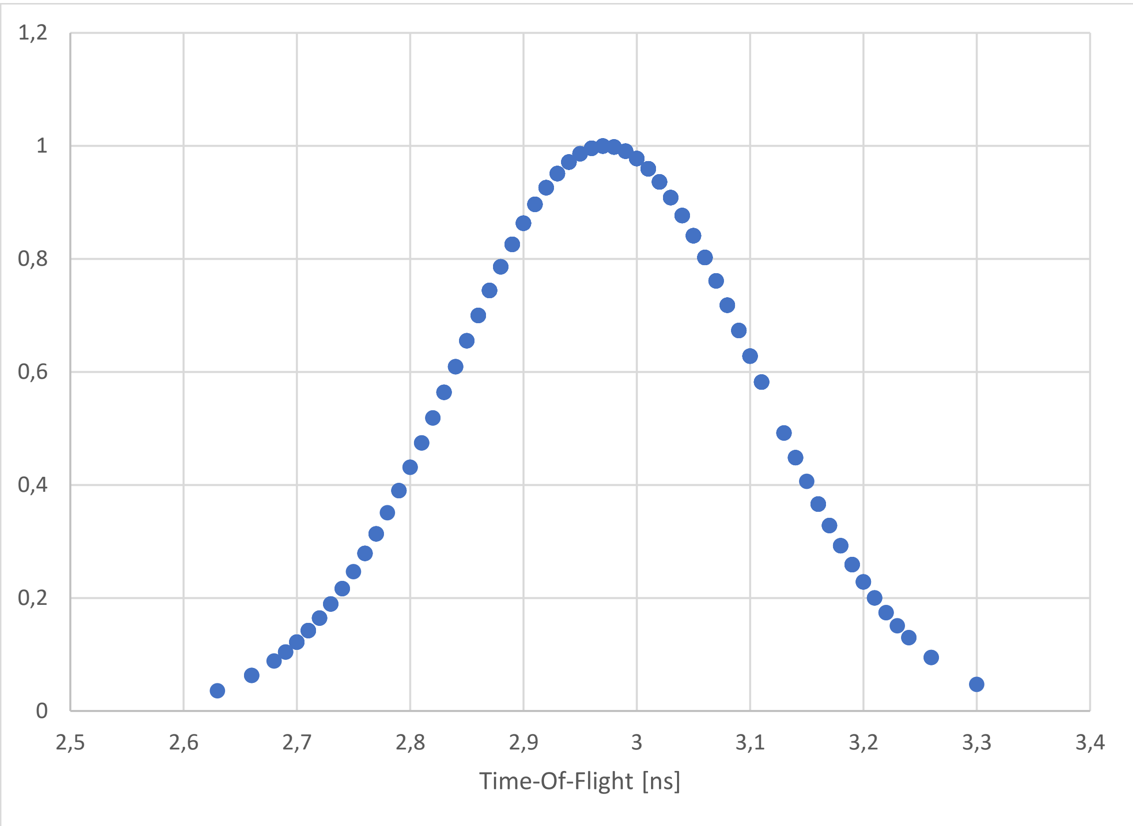


Fig. 56 Normal Distribution - No Average Mode

Fig. 56 shows the normal distribution obtained for that data, while the table below resumes the main statistic extracted from the measurements.

Table 16 Statistics T = 8 ns - F = 10 Mhz - No Average Mode

Statistics	
Mean	2.972
Standard Error	0.007249471
Median	2.97
Mode	3.01
Standard Deviation	0.132687107
Sample Variance	0.017605868
Kurtosis	-0.230234778
Skewness	-0.185670116
Range	0.67
Minimum	2.63
Maximum	3.3

The expected ToF was 3 ns corresponding to a distance of:

$$d = 3 \times 10^{-9} * 3 \times 10^{8} = 0.9\,m$$

Performing a single measurement with this configuration will provide a result in range:

$$\begin{aligned} d_{measured,8ns,1MHz} &= \bar{d} \pm 3 \times \sigma \\ &= 2.972 \times 10^{-9} * 3 \times 10^{8} \pm 3 * 0.13269 \times 10^{-9} * 3 \times 10^{8} \\ &= 0.8916 \pm 0.1194\,m \end{aligned}$$

11 Conclusions and Discussion

In this chapter the results from the experiments described in the previous section are reviewed and commented; moreover, conclusions are drawn from their analysis.

Future implementations and enhancements are then explained, both for the measurements on the used setup and the form of idea which could be applied to improve performance.

11.1 Measurements

The setup, described in the first sections of the previous chapter, was used in almost all the measurements except for the generation of long time-of-flight, used to evaluate the effect of the distance on the accuracy. Since a too long coaxial cable would be needed to obtain long delays, in these cases a longer time-of-flight was obtained by programming two pins of the Renesas board to output the star signal and the delayed stop signals. The delay was set using the internal timer of the board, measured with the oscilloscope to evaluate the precision of that delay and then measured with the designed circuit.

11.1.1 The effect of the rise time

The following table resumes the relevant characteristics of the data obtained with different rise times of the start and stop impulses.

Table 17 Resume Table for Different ToF

Rise Time [ns]	Mean [ns]	Standard Deviation [ns]
8	2.955862	0.033185
12	2.885944	0.049712
20	2.976712	0.052635

It is possible to notice that the rise time heavily affects the accuracy of the measurement, in particular the uncertainty increases as the rise time increases.

This behavior is show in *Fig. 57.*

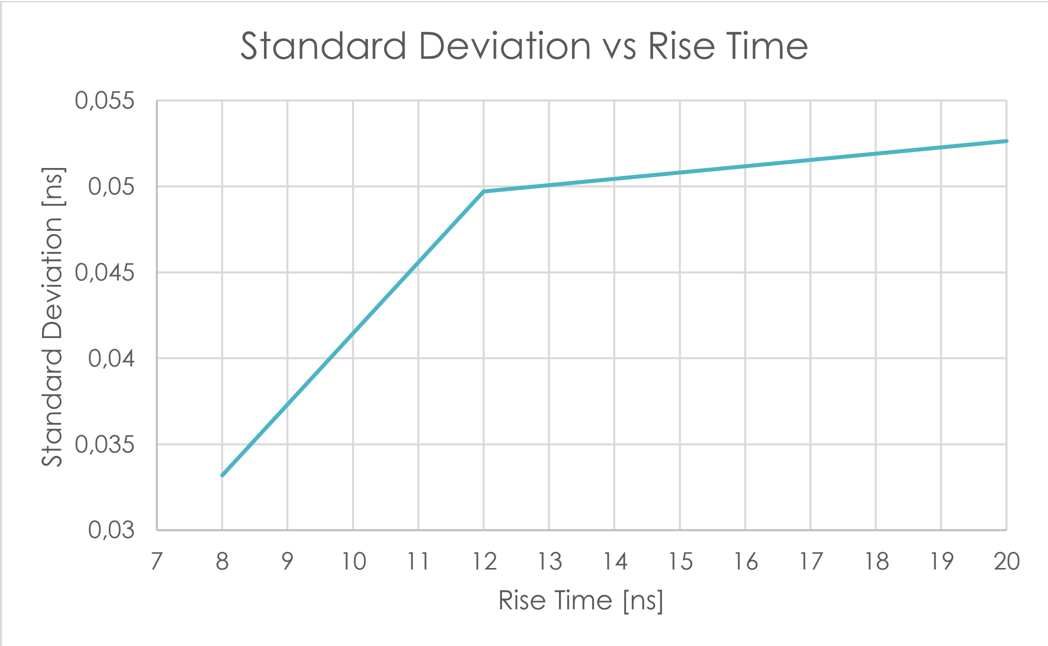


Fig. 57 Standard Deviation vs Rise Time

This observation leaded to the conclusion that a short rise time for both the signals is necessary to achieve high accuracy. Indeed, the best case in which the rise time was set to 8 ns, presents the lowest standard deviation that means the lowest dispersion of the data around the mean value. For this reason, the data acquired to evaluate the effect of the distance and the effect of the frequency of the impulses, were all taken with start and stop impulses with rise time of 8 ns.

11.1.2 The effect of the frequency

Another important feature of the impulses incoming the receiver chain is the frequency. The question was analyzed by taking the same distance and measuring it with impulses that presents the same rise time but different frequency. The interesting aspect are enhanced comparing the impulses at 10 MHz with that at 1 MHz; *Table 18* resumes the characteristics of the acquired data for both the frequencies.

Table 18 The Effect of the Frequency

Frequency [MHz]	Mean [ns]	Standard Deviation [ns]
1	2.9757	0.0531
10	2.9558	0.0332

The highest frequency produced data with the lowest standard deviation. Therefore, to obtain reliable and accurate data it is advisable to use high frequency impulses.

It would be interesting to test the system with impulses with frequency higher than 10 MHz, but unfortunately the available signal generators were only capable to generate impulses at frequency 10 MHz or lower.

The observed data are not enough to draw relevant conclusions, however it seems to be better to use high frequency impulses, for that reason all the other tests were performed using the maximum available impulses frequency, that is $F_{MAX} = 10$ MHz.

11.1.3 How the distance affects the accuracy

Since the system is designed to operate also over long distance, it was necessary to understand how the standard deviation changes as the distance from the target increases.

The optimum setup was used in this test:

- F = 10 MHz
- T_{RISE} = 8 ns
- Average Mode

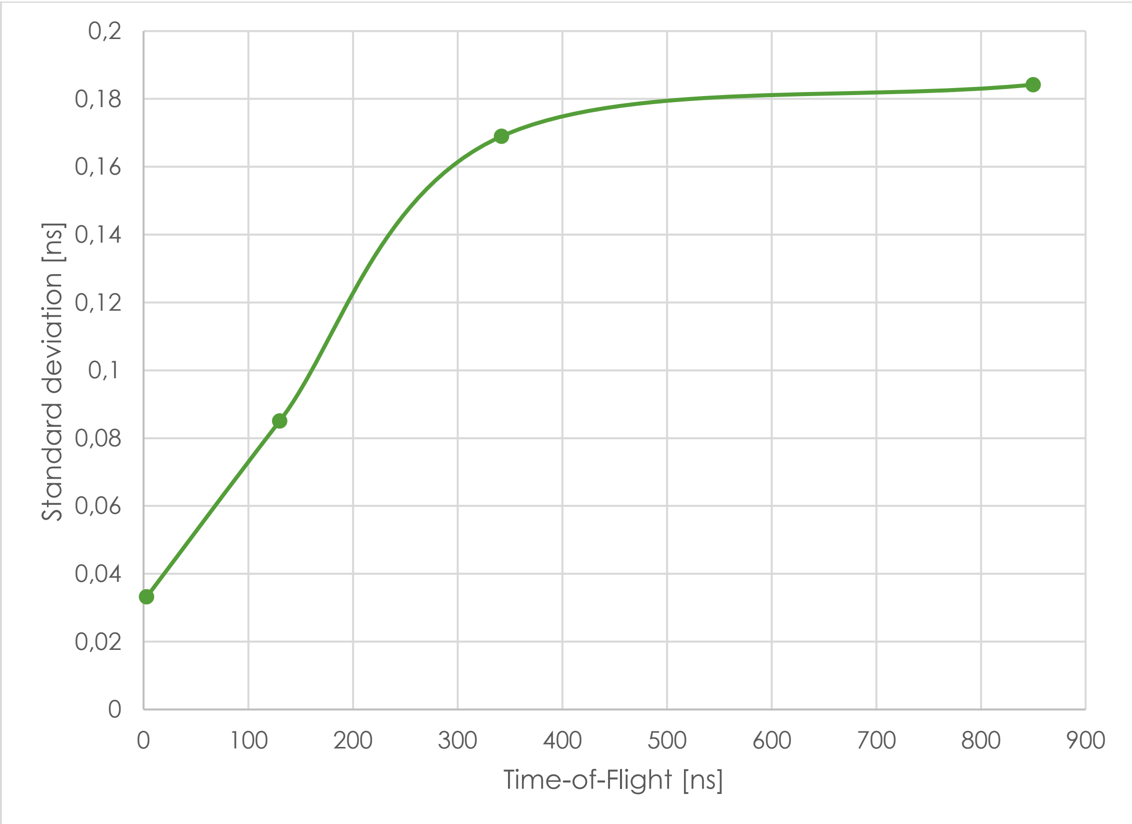


Fig. 58 Distance vs Deviation [ns]

As shown in *Fig. 58*, the measurements uncertainty increases with increasing the distance. Albeit this behavior may seem like a big problem, the computations emphasize a different aspect.

Table 19 Resume Table for Different Distances

Mean [ns]	Standard Deviation [ns]	Mean [m]	Standard Deviation [m]	2*σ (95%)	3 *σ (99%)
2.955	0.0331	0.8867	0.009	0.019	0.029
127.236	0.0850	38.170	0.025	0.051	0.076
342.078	0.1689	102.623	0.050	0.101	0.152
853.485	0.1842	256.045	0.055	0.110	0.165

Comparing the data acquired for the shortest measured ToF with the data of the longer one, it is possible to observe that the standard deviation of the longer ToF, expressed in nanoseconds, is more or less five times the one of the shorter one but the ToF is almost 300 times greater.

Moreover, it is interesting to notice that the uncertainty of the 255 m measurement is just 16 cm.

Whit respect to the objectives of this project, the obtained accuracy is considered more than enough for all the tested distances.

11.1.4 The best acquisition mode: Averaging Mode

As expected, the best acquisition mode relies on the averaging mode. By comparing the standard deviation of the data acquired for both the modes it is possible to observe that the averaging mode guarantees a 75% reduction in the standard deviation.

Table 20 Averaging Mode Improvements

Mode	Mean	Standard Deviation [ns]
Single Shot	2.972	0.132687
Averaging	2.955862	0.033185 (-75%)

11.2 Conclusions

The analysis of the acquired data suggested that to obtain accurate and reliable results the rise time should be the lowest possible while the frequency of the impulses should be as higher as possible but respecting the requirements of the TDC7200. Moreover, the results are reliable even when measuring long distances (~100 m). To obtain accurate data when measuring these distances, it is necessary to use the averaging mode. The main drawback of this approach is that the number acquired measurements is lower that the single shot mode and thus high frequency pulses are mandatory.

The implemented measurement setup resulted stable and reliable but high-performance instruments are suggested to better analyze the system. More tests are then necessary to access the performances of the prototype, but the obtained results are considered precise enough for this project.

11.3 Possible Improvements

Since the accuracy is influenced by the rise time of the start and stop signals it might be useful to insert a fast comparator or a schmitt trigger just before the start and stop pin of each TDC7200.

A more precise design of the pcb might help to reduce the noise affecting each measurement, for example matching the impedance on each trace allows to maintain the signal integrity and the clarity of the transmitted signals.

From the software point of view other enhancements are possible and a lighter integration can help to improve the overall performances. Moreover, in this project it was decided to use the Teensy board as the controller of the entire system but since each component is designed to communicate using the SPI protocol it is possible to use every microcontroller that provides this kind of communication protocol. It is then possible to better tune the timing between each operation of each component.

It should be possible to design a compact system that implements all the components in a relatively small pcb. This solution can be studied for future implementations.

Appendix 1: Codes

1 TDC7200 Code

```
#include <SPI.h>
#include"tdc7200.h"

#define serialbufferSize 50
#define commandDelimeters "|,.- "
#define TDC7200_CS_PIN 15


//real varibles
char inputBuffer[serialbufferSize] ;
int serialIndex = 0; // keep track of where we are in the buffer
int num_measure=0;
byte power = 16;
byte config1_rx, config2_rx, int_status, config1_tx, config2_tx;
byte int_mask, ccoh, ccol, clockcoh, clockcol, clockcstopmaskh,
clockcstopmaskl;
float tof_rx = 0;
float tof_tx = 0;
// End of real variables

// Setup Code
void setup()
{
   //Initialize SPI
 pinMode(TDC7200_CS_PIN, OUTPUT);
  pinMode(14 , OUTPUT);
  pinMode(MISO, INPUT);
  pinMode(MOSI, OUTPUT);
  pinMode(power , OUTPUT);


  SPI.begin();
   //Initialize Serial
 Serial.begin(9600);
   //Initialize Outputs
 digitalWrite(power, LOW);
  delay(5); //Sensor needs 300microseconds to startup, 1.5ms
recommended
 digitalWrite(power, HIGH);
  delay(5);
}
// Function used to start the measuring process
```

```
voidstart_TDC()
{

  while(num_measure < 1000){
    tdc7200 chip1;


    //WRITE 00000011 INTO CONFIG 1 TO GET READY FOR MEASUREMENT1
   chip1.writeReg(CONFIG1, B00000001);
   chip1.writeReg_TX(CONFIG1, B00000001);
   delayMicroseconds(100);
    //WRITE B01000011 INTO CONFIG2 for FOUR STOPS
              // 000: Single Stop
              // 001: Two Stops
              // 010: Three Stops
              // 011: Four Stops
              // 100: Five Stops
   chip1.writeReg(CONFIG2, B01000100);
   chip1.writeReg_TX(CONFIG2, B01000100);
  delayMicroseconds(100);


    //SEND 5 STOPS


  //READ REGISTERS AND COMPUTE VALUES
   chip1.time1_rx = chip1.readReg24(TIME1);
   chip1.time1_tx = chip1.readReg24_TX(TIME1);
   chip1.time2_rx = chip1.readReg24(TIME2);
   chip1.time2_tx = chip1.readReg24_TX(TIME2);
   chip1.time3_rx = chip1.readReg24(TIME3);
   chip1.time4_rx = chip1.readReg24(TIME4);
   chip1.time5_rx = chip1.readReg24(TIME5);
   chip1.time6_rx = chip1.readReg24(TIME6);
  chip1.clockcount1_rx = chip1.readReg24(CLOCK_COUNT1);
  chip1.clockcount1_tx = chip1.readReg24_TX(CLOCK_COUNT1);
  chip1.clockcount2_rx = chip1.readReg24(CLOCK_COUNT2);
  chip1.clockcount2_tx = chip1.readReg24_TX(CLOCK_COUNT2);
  chip1.clockcount3_rx = chip1.readReg24(CLOCK_COUNT3);
  chip1.clockcount4_rx = chip1.readReg24(CLOCK_COUNT4);
  chip1.clockcount5_rx = chip1.readReg24(CLOCK_COUNT5);
   chip1.cal1_rx = chip1.readReg24(CALIBRATION1);
```

```
chip1.cal1_tx = chip1.readReg24_TX(CALIBRATION1);
chip1.cal2_rx = chip1.readReg24(CALIBRATION2);
chip1.cal2_tx = chip1.readReg24_TX(CALIBRATION2);
//END OF READ REGISTERS



//READ REGISTERS AND COMPUTE VALUES



chip1.time3_tx = chip1.readReg24_TX(TIME3);
chip1.time4_tx = chip1.readReg24_TX(TIME4);
chip1.time5_tx = chip1.readReg24_TX(TIME5);
chip1.time6_tx = chip1.readReg24_TX(TIME6);
chip1.clockcount1_tx = chip1.readReg24_TX(CLOCK_COUNT1);
chip1.clockcount3_tx = chip1.readReg24_TX(CLOCK_COUNT3);
chip1.clockcount4_tx = chip1.readReg24_TX(CLOCK_COUNT4);
chip1.clockcount5_tx = chip1.readReg24_TX(CLOCK_COUNT5);


chip1.tof1 = chip1.getTOF_rx(chip1.time1_rx , chip1.time2_rx ,chip1.
clockcount1_rx, chip1.cal1_rx , chip1.cal2_rx);
chip1.tof2 = chip1.getTOF_rx(chip1.time1_rx , chip1.time3_rx ,chip1.
clockcount2_rx, chip1.cal1_rx , chip1.cal2_rx);
chip1.tof3 = chip1.getTOF_rx(chip1.time1_rx, chip1.time4_rx, chip1.
clockcount3_rx,chip1.cal1_rx,chip1.cal2_rx);
chip1.tof4 = chip1.getTOF_rx(chip1.time1_rx, chip1.time5_rx, chip1.
clockcount4_rx,chip1.cal1_rx,chip1.cal2_rx);
chip1.tof5 = chip1.getTOF_rx(chip1.time1_rx, chip1.time6_rx, chip1.
clockcount5_rx,chip1.cal1_rx,chip1.cal2_rx);


tof_rx=(chip1.tof1+chip1.tof2+chip1.tof3+chip1.tof4+chip1.tof5)/5;


chip1.tof1 = chip1.getTOF_tx(chip1.time1_tx , chip1.time2_tx ,chip1.
clockcount1_tx, chip1.cal1_tx , chip1.cal2_tx);
chip1.tof2 = chip1.getTOF_tx(chip1.time1_tx , chip1.time3_tx ,chip1.
clockcount2_tx, chip1.cal1_tx , chip1.cal2_tx);
chip1.tof3 = chip1.getTOF_tx(chip1.time1_tx, chip1.time4_tx, chip1.
```

```
clockcount3_tx,chip1.cal1_tx,chip1.cal2_tx);
 chip1.tof4 = chip1.getTOF_tx(chip1.time1_tx, chip1.time5_tx, chip1.
clockcount4_tx,chip1.cal1_tx,chip1.cal2_tx);
 chip1.tof5 = chip1.getTOF_tx(chip1.time1_tx, chip1.time6_tx, chip1.
clockcount5_tx,chip1.cal1_tx,chip1.cal2_tx);


 tof_tx=(chip1.tof1+chip1.tof2+chip1.tof3+chip1.tof4+chip1.tof5)/5;




  //END OF READ REGISTERS
  /*


 Serial.println();
Serial.print(chip1.tof1*1E6,8);
 Serial.print(",");
Serial.print(chip1.tof2*1E6,8);
 Serial.print(",");
Serial.print(chip1.tof3*1E6,8);
 Serial.print(",");
Serial.print(chip1.tof4*1E6,8);
 Serial.print(",");
Serial.print(chip1.tof5*1E6,8);

 Serial.println();
  */
    /*
   int_status = chip1.readReg8_TX(INT_STATUS);
   Serial.print("INT_STATUS = "); Serial.println(int_status,BIN);
   int_mask = chip1.readReg8(INT_MASK);
   ccoh = chip1.readReg8(COARSE_CNTR_OVF_H);
   ccol = chip1.readReg8(COARSE_CNTR_OVF_L);
   clockcoh = chip1.readReg8(CLOCK_CNTR_OVF_H);
   clockcol = chip1.readReg8(CLOCK_CNTR_OVF_L);
  clockcstopmaskh = chip1.readReg8(CLOCK_CNTR_STOP_MASK_H);
  clockcstopmaskl = chip1.readReg8(CLOCK_CNTR_STOP_MASK_L);

   Serial.print("INT_MASK = "); Serial.println(int_mask,BIN);
   Serial.print("COARSE_CNTR_OVF_H = "); Serial.println(ccoh,BIN);
```

```
  Serial.print("COARSE_CNTR_OVF_L = "); Serial.println(ccol,BIN);
  Serial.print("CLOCK_CNTR_OVF_H = "); Serial.println(clockcoh,BIN);
  Serial.print("COARSE_CNTR_OVF_L = "); Serial.println(clockcol,
BIN);
   Serial.print("COARSE_CNTR_OVF_L = "); Serial.
println(clockcstopmaskh,BIN);
   Serial.print("COARSE_CNTR_OVF_L = "); Serial.
println(clockcstopmaskl,BIN);


   Serial.println();

    */
   //TESTPRINT CONFIG1/2

  //Serial.print("CONFIG1_RX = "); Serial.println(config1_rx, BIN);
   //Serial.print(",");
  //Serial.print("CONFIG2_RX = "); Serial.println(config2_rx, BIN);
   //Serial.print(",");
  //Serial.print("CONFIG1_TX = "); Serial.println(config1_tx, BIN);
  //Serial.print("CONFIG2_TX = "); Serial.println(config2_tx, BIN);
  // Serial.print(",");
   //Serial.print("TIME1 = "); Serial.println(chip1.
readReg24_TX(TIME1), BIN);
  // Serial.print(",");
   //Serial.println();
 //  Serial.print("  ****************");
  // Serial.println();
  // Serial.print("* TOF1 = ");
  Serial.print(tof_rx*1E6,8); //Serial.println(" us*");
   Serial.print(",");

 //  Serial.print("* TOF1 TX = ");
  Serial.print(tof_tx*1E6,8);// Serial.println(" us*");
   //Serial.print(",");
    /*
    Serial.print("* TOF2 = ");
   Serial.print(chip1.tof2,8); Serial.println(" *");
    Serial.print("* TOF3 = ");
   Serial.print(chip1.tof3,8); Serial.println(" *");
    Serial.print("* TOF4 = ");
   Serial.print(chip1.tof4,8); Serial.println(" *");
```

```
    Serial.print("* TOF5 = ");
   Serial.print(chip1.tof5,8); Serial.println(" *");
    Serial.print(" ****************");
     */
    Serial.println();

    num_measure++;

    if (num_measure==1000)
     {
     num_measure=0;
       delay(1);
     }
  }




}


// Loop
void loop()
{

   // see if there is a command come in on the serial port and if so
call the command processor
   if (CheckSerial()) DoCommand(inputBuffer);
}

// Functions to interact with the board
booleanCheckSerial()
{
  boolean lineFound = false;
  // if there's any serial available, read it:
  while (Serial.available() > 0) {
    //Read a character as it comes in:

    char charBuffer = Serial.read();
    if (charBuffer == '\n') {
      inputBuffer[serialIndex] = 0; // terminate the string
       lineFound = (serialIndex > 0); //
       serialIndex = 0; // reset for next line of data
```

```
    }
    else if (charBuffer == '\r') {

    }
    else if (serialIndex < serialbufferSize && lineFound == false) {
      /*Place the character in the string buffer:*/
     inputBuffer[serialIndex++] = charBuffer; // auto increment index
    }
  }// End of While
  return lineFound;
}// End of CheckSerial()

// Function to receive commands through the serial interface
boolean DoCommand(char * commandBuffer)
{
 Serial.println("Command");
  char* Command; // Command Parameter
  char* Parameter; // Additional Parameter


  // Get the command from the input string
  Command = strtok(commandBuffer, commandDelimeters); // get the
command
  Parameter = strtok(NULL, commandDelimeters); // get the parameter
if any
  //if there are more than one parameter they will be ignored for now
 if (strcmp(Command, "start") == 0) { start_TDC(); num_measure=0;}

 else {  Serial.print("Error "); Serial.println(commandBuffer); }
 return true;




}
/*
 //READ/WRITE NEW_MEASUREMENT into Interrupt
 int_status = chip1.readReg8(INT_STATUS);
 int_mask = chip1.readReg8(INT_MASK);
 ccoh = chip1.readReg8(COARSE_CNTR_OVF_H);
 ccol = chip1.readReg8(COARSE_CNTR_OVF_L);
 clockcoh = chip1.readReg8(CLOCK_CNTR_OVF_H);
```

```
 clockcol = chip1.readReg8(CLOCK_CNTR_OVF_L);
 clockcstopmaskh = chip1.readReg8(CLOCK_CNTR_STOP_MASK_H);
 clockcstopmaskl = chip1.readReg8(CLOCK_CNTR_STOP_MASK_L);

 Serial.print("INT_STATUS = "); Serial.println(int_status,BIN);
 Serial.print("INT_MASK = "); Serial.println(int_mask,BIN);
 Serial.print("COARSE_CNTR_OVF_H = "); Serial.println(ccoh,BIN);
 Serial.print("COARSE_CNTR_OVF_L = "); Serial.println(ccol,BIN);
 Serial.print("CLOCK_CNTR_OVF_H = "); Serial.println(clockcoh,BIN);
 Serial.print("COARSE_CNTR_OVF_L = "); Serial.println(clockcol,BIN);
 Serial.print("COARSE_CNTR_OVF_L = "); Serial.
println(clockcstopmaskh,BIN);
 Serial.print("COARSE_CNTR_OVF_L = "); Serial.
println(clockcstopmaskl,BIN);


 Serial.println();


/*
  //PRINTS FOR DEBUG
 Serial.print("TIME1 = "); Serial.println(chip1.time1);
 Serial.print("CLOCK_COUNT1 = "); Serial.println(chip1.clockcount1);
 Serial.print("TIME2 = "); Serial.println(chip1.time2);
 Serial.print("CLOCK_COUNT2 = "); Serial.println(chip1.clockcount2);
 Serial.print("TIME3 = "); Serial.println(chip1.time3);
 Serial.print("CLOCK_COUNT3 = "); Serial.println(chip1.clockcount3);
 Serial.print("TIME4 = "); Serial.println(chip1.time4);
 Serial.print("CLOCK_COUNT4 = "); Serial.println(chip1.clockcount4);
 Serial.print("TIME5 = "); Serial.println(chip1.time5);
 //Serial.print("CLOCK_COUNT5 = "); Serial.println(chip1.
clockcount5);
 //Serial.print("TIME6 = "); Serial.println(chip1.time6);
 Serial.print("CALIBRATION1 = "); Serial.println(chip1.cal1);
 Serial.print("CALIBRATION2 = "); Serial.println(chip1.cal2);
*/
```

Code 1 TDC7200 Code

2 AD7606 Code

```
#include <SPI.h>

#defineSCALE_FACTOR0.000152587890625
#define SPIfrequency (uint32_t) 250000
#define BUSY 33 //3
#define RESET 5 //4
#define START_CONVERSION 5 //5
#define CHIP_SELECT 36 //10
#define TOTAL_RAW_BYTES 16
#define power 16


int bytesToRead = TOTAL_RAW_BYTES;
byteraw[TOTAL_RAW_BYTES];
signed long parsed[8];

void setup() {
 pinMode(BUSY, INPUT);
 pinMode(RESET, OUTPUT);
 pinMode(START_CONVERSION, OUTPUT);
 pinMode(power, OUTPUT);
 // 0pinMode(mode, OUTPUT);
 pinMode(MISO, INPUT);

 SPI.begin();
 SPI.setDataMode(SPI_MODE0);
 //Initialize Serial
 Serial.begin(9600);

 digitalWrite(power, HIGH);
 //digitalWrite(mode, LOW);
 digitalWrite(START_CONVERSION, HIGH);
 digitalWrite(CHIP_SELECT, HIGH);
 digitalWrite(RESET, HIGH);
 delay(1);
 digitalWrite(RESET, LOW);
}

void parseRawBytes() {
  int i;
```

```
     parsed[0] = (raw[0] << 8) + (raw[1] >> 0);
     parsed[1] = (raw[2] << 8) + (raw[3] >> 0);
     parsed[2] = (raw[4] << 8) + (raw[5] >> 0);
     parsed[3] = (raw[6] << 8) + (raw[7] >> 0);
     parsed[4] = (raw[8] << 8) + (raw[9] >> 0);
     parsed[5] = (raw[10] << 8) + (raw[11] >> 0);
     parsed[6] = (raw[12] << 8) + (raw[13] >> 0);
     parsed[7] = (raw[14] << 8) + (raw[15] >> 0);
for(i=0; i<8; i++) {
   parsed[i] = fixSignBit(parsed[i]);
  }



}

void loop() {
  int i;

 digitalWrite(START_CONVERSION, LOW);
 delayMicroseconds(10);
 digitalWrite(START_CONVERSION, HIGH);

  while (digitalRead(BUSY) == HIGH) {
    // wait for conversion to complete
  }

SPI.beginTransaction(SPISettings(SPIfrequency,MSBFIRST,
SPI_MODE0));
 digitalWrite(CHIP_SELECT, LOW);
  while (bytesToRead > 0) {
   raw[TOTAL_RAW_BYTES - bytesToRead] = SPI.transfer(0x00);
   bytesToRead--;
  }
 digitalWrite(CHIP_SELECT, HIGH);
 bytesToRead = TOTAL_RAW_BYTES;

 parseRawBytes();

 Serial.print((float)parsed[0] * SCALE_FACTOR, 5);
 Serial.println();
```

```
 Serial.print("\r\n");
  delay(500);
}



long fixSignBit(long reading) {

     if(reading & 0x8000) { // if reading is < 0 (stored as two's
complement)
       return reading | 0xFFFF0000; // set bits 31-16
      } else
        return reading;
   }
```

Code 2 AD7606 Code

3 DAC8734 Code

```
#include <SPI.h>
// define the buffer size...
#define serialbufferSize 50
#define commandDelimeters "|,.- "

// Now the real varibles
char inputBuffer[serialbufferSize]  ;
int serialIndex = 0; // keep track of where we are in the buffer
#define DAC_8734_CS_PIN  SS
#define VREF 5.000
// Default values below assume 0-10V Unipolar or -5 to +5 bipolar
output
#define DAC_VFSR VREF*4 // may be 0 - 10 or -5 to +5
#define DAC_8734_Command 0 // command register
#define DAC_8734_DataBase 0x04 // register offset to zero cal base
register
#define DAC_8734_CalZeroBase 0x08 // register offset to zero cal base
register
#define DAC_8734_CalGainBase 0x0C // register offset to gain cal base
register
#define DACMAX 0xFFFF
#define DACMIN 0x0000

// default setup of DAC8734, these can be adjusted if needed, some
functions also change them to change mode etc.
int DAC_Gain0 = 0; // 0 = *2, 1 = *4
int DAC_Gain1 = 0; // 0 = *2, 1 = *4
int DAC_Gain2 = 0; // 0 = *2, 1 = *4
int DAC_Gain3 = 0; // 0 = *2, 1 = *4
int DAC_GPIO0 = 0; // 1 = Group A in Unipolar 0=Bipolar (External
connection to control pin)
int DAC_GPIO1 = 0; // 1 = Group B in Unipolar 0=Bipolar (External
connection to control pin)
int DAC_PD_A = 0;  // 1 = group A power down
int DAC_PD_B = 0;  // 1 = group B power down
int DAC_DSDO = 0;  // 1 = Disable SDO bit.


SPISettingssettingsDAC8734(5000000,MSBFIRST,SPI_MODE1);
```

```
// Sine table, used to create a reasonable sinewave on a dac output,
256 16bit values, one complete cycle
byte val = 0;
unsigned int Sin_tab[256] = {  32768, 33572, 34376, 35178, 35980,
36779, 37576, 38370, 39161, 39947, 40730, 41507, 42280,
                             43046, 43807, 44561, 45307, 46047,
46778, 47500, 48214, 48919, 49614, 50298, 50972, 51636,
                             52287, 52927, 53555, 54171, 54773,
55362, 55938, 56499, 57047, 57579, 58097, 58600, 59087,
                             59558, 60013, 60451, 60873, 61278,
61666, 62036, 62389, 62724, 63041, 63339, 63620, 63881,
                             64124, 64348, 64553, 64739, 64905,
65053, 65180, 65289, 65377, 65446, 65496, 65525, 65535,
                             65525, 65496, 65446, 65377, 65289,
65180, 65053, 64905, 64739, 64553, 64348, 64124, 63881,
                             63620, 63339, 63041, 62724, 62389,
62036, 61666, 61278, 60873, 60451, 60013, 59558, 59087,
                             58600, 58097, 57579, 57047, 56499,
55938, 55362, 54773, 54171, 53555, 52927, 52287, 51636,
                             50972, 50298, 49614, 48919, 48214,
47500, 46778, 46047, 45307, 44561, 43807, 43046, 42280,
                             41507, 40730, 39947, 39161, 38370,
37576, 36779, 35980, 35178, 34376, 33572, 32768, 31964,
                             31160, 30358, 29556, 28757, 27960,
27166, 26375, 25589, 24806, 24029, 23256, 22490, 21729,
                             20975, 20229, 19489, 18758, 18036,
17322, 16617, 15922, 15238, 14564, 13900, 13249, 12609,
                             11981, 11365, 10763, 10174, 9598,
9037, 8489, 7957, 7439, 6936, 6449, 5978, 5523, 5085, 4663,
                             4258, 3870, 3500, 3147, 2812, 2495,
2197, 1916, 1655, 1412, 1188, 983, 797, 631, 483, 356, 247,
                             159, 90, 40, 11, 1, 11, 40, 90, 159,
247, 356, 483, 631, 797, 983, 1188, 1412, 1655, 1916, 2197,
                             2495, 2812, 3147, 3500, 3870, 4258,
4663, 5085, 5523, 5978, 6449, 6936, 7439, 7957, 8489, 9037,
                             9598, 10174, 10763, 11365, 11981,
12609, 13249, 13900, 14564, 15238, 15922, 16617, 17322,
                             18036, 18758, 19489, 20229, 20975,
21729, 22490, 23256, 24029, 24806, 25589, 26375, 27166,
                             27960, 28757, 29556, 30358, 31160,
31964
                            };
```

```
//Calibration table, you will need to adjust these values to suit
your build
int DAC_CAL_tab[8] = { 0x08, 0xfF, 0x0F, 0x4F,  0x3f, 0x80, 0x80,
0x80}; // zero's then gain's

// The DAC8734 uses Mode 1 SPI, has an 8 bit address byte followed by
16bit data
// high byte first, high bit first
// This is the register descriptions
// Reg  description
// 0   Control Register
// 1   Monitor
// 2   Not Used
// 3   Not Used
// 4   DAC 0 Data Register
// 5   DAC 1 Data Register
// 6   DAC 2 Data Register
// 7   DAC 3 Data Register
// 8   DAC 0 Zero Cal Register
// 9   DAC 1 Zero Cal Register
// a   DAC 2 Zero Cal Register
// b   DAC 3 Zero Cal Register
// c   DAC 0 Zero Gain Register
// d   DAC 1 Zero Gain Register
// e   DAC 2 Zero Gain Register
// f   DAC 3 Zero Gain Register


// the most basic function, write to register "reg", with a value
"val"
void WriteDACRegister(byte reg, unsigned int val)
{


SPI.beginTransaction(settingsDAC8734);

 digitalWrite(DAC_8734_CS_PIN, LOW); // Select the Chip
 SPI.transfer(reg); // Select the target register
 SPI.transfer(val >> 8); // Send the High Data Byte
 SPI.transfer(val & 0xFF); // Send the Low Data Byte
 digitalWrite(DAC_8734_CS_PIN, HIGH); // Release the Chip
```

```
 SPI.endTransaction();

}

// Currently set to Unipolar, 2* Gain and Powered up
void InitDAC()
{
  // for now, gain of 2, gpio hiZ,
  int DAC_INIT = 0x0000 | DAC_PD_A << 12 | DAC_PD_B << 11 | DAC_GPIO1
<< 9 | DAC_GPIO0 << 8 | DAC_DSDO << 7 | DAC_Gain3 << 5 | DAC_Gain2 <<
4 | DAC_Gain1 << 3 | DAC_Gain0 << 2 ;
 //Serial.print("Command Reg = "); Serial.print(DAC_INIT, HEX);
Serial.print(" ");Serial.println(DAC_INIT, BIN);
 WriteDACRegister(DAC_8734_Command, DAC_INIT);
}

// Helper Function to output to a specified DAC, a Desired value
between 0 and 65535
// this could represent 0V - 10V or -5 to +5V depending on mode
void SetDAC( byte channel, unsigned int value)
{
  if (channel > 3)
  {
   Serial.println("DAC  must be 0 - 3");
  }
  else
  {
   Serial.print("DAC "); Serial.print(channel); Serial.print(" Value
");Serial.println(value, HEX);
   WriteDACRegister(DAC_8734_DataBase + channel, value);
  }
}

//Output the Calibration table to the DAC for all channels
voidCalibrateDAC()
{
  for (int x = 0 ; x < 4 ; x++)
  {
   WriteDACRegister(DAC_8734_CalZeroBase + x, DAC_CAL_tab[x]) ;//
Zero cal
   WriteDACRegister(DAC_8734_CalGainBase + x, DAC_CAL_tab[x + 4]);
// Gain cal
```

```
  }
}

void sine( byte channel,  int loopcount)
{
  if (loopcount > 65535) loopcount = 65535;
  if (channel > 3) channel = 0;
 Serial.print("Sine on DAC ");Serial.print(channel ); Serial.print("
Cycles=");Serial.println(loopcount);
  for (int y = 0 ; y < loopcount ; y++)
  {
    for (int x = 0 ; x < 256 ; x++)
    {
     SetDAC(channel, Sin_tab[x]);
    }
  }
 Serial.println("Complete");
}

void help()
{
 Serial.println("DAC8734 Test Program:");
 Serial.println("Commands:");
  Serial.println("help\" for this menu" );
  Serial.println("dacx  x is from 0 to 3, value 0 to 65535" );
  Serial.println("bpA or bpB mode  Bipolar");
  Serial.println("upA or upB mode Unipolar" );
  Serial.println("offA or offB Power Off group A or B" );
  Serial.println("onA or onB   Power On for Group A or B" );
  Serial.println("sinex cycles, x is DAC from 0 to 3, cycles 0 to
65535" );
 Serial.println();
  Serial.println("make sure \"NewLine\" is on in the console setup" );

}

void setup()
{
 Serial.begin(115200);
  // print the help menu for the DAC8734 tester
  help();
  // setup the chip select pin for the DAC
```

```
 pinMode(DAC_8734_CS_PIN, OUTPUT);
  pinMode(MISO, INPUT);
  pinMode(MOSI, OUTPUT);
  SPI.begin();
 SPI.setDataMode(SPI_MODE1);
  // Initialisze the DAC8734
  InitDAC();
  // output the DAC_CAL_tab to the DAC
 CalibrateDAC();
  // set the DAC outputs to Zero
  SetDAC(0, 0x0000);
  SetDAC(1, 0x0000);
  SetDAC(2, 0x0000);
  SetDAC(3, 0x0000);
}

void loop()
{
  // see if there is a command come in on the serial port and if so
call the command processor
   if (CheckSerial()) DoCommand(inputBuffer);
}



booleanCheckSerial()
{
  boolean lineFound = false;
  // if there's any serial available, read it:
  while (Serial.available() > 0) {
    //Read a character as it comes in:
    //currently this will throw away anything after the buffer is
full or the \n is detected
    char charBuffer = Serial.read();
    if (charBuffer == '\n') {
      inputBuffer[serialIndex] = 0; // terminate the string
      lineFound = (serialIndex > 0); // only good if we sent more
than an empty line
      serialIndex = 0; // reset for next line of data
     }
    else if (charBuffer == '\r') {
      // Just ignore the Carrage return, were only interested in new
```

```
line
    }
   else if (serialIndex < serialbufferSize && lineFound == false) {
     /*Place the character in the string buffer:*/
     inputBuffer[serialIndex++] = charBuffer; // auto increment index
    }
  }// End of While
  return lineFound;
}// End of CheckSerial()


// Enhanced Command Processor using strtok to strip out command from
multi parameter string
boolean DoCommand(char * commandBuffer)
{
 Serial.println("Command");
  char* Command; // Command Parameter
  char* Parameter; // Additional Parameter
  unsigned int analogVal = 0; // additional parameter converted to
analog if possible

  // Get the command from the input string
  Command = strtok(commandBuffer, commandDelimeters); // get the
command
  Parameter = strtok(NULL, commandDelimeters); // get the parameter
if any
  //if there are more than one parameter they will be ignored for now

  // Make sure we have an analog value if we are to allow PWM output
  unsigned int outparameter = isNumeric (Parameter);

  //if it is a number then convert it
  if (outparameter)
   {
    analogVal = atoi(Parameter);
    // check the analog value is in the correct range
    if (analogVal < DACMIN || analogVal > DACMAX) outparameter =
false;
   }
  // Standard way to handle commands
  if (strcmp(Command, "help") == 0) { help(); }
  //DAC Outputs if we have a valid analog parameter
```

```
  else if (strcmp(Command, "dac0") == 0  && outparameter ) {
SetDAC(0, analogVal);}   // Set the DAC 0 output
  else if (strcmp(Command, "dac1") == 0  && outparameter ) {
SetDAC(1, analogVal);}  // Set the DAC 1 output
  else if (strcmp(Command, "dac2") == 0  && outparameter ) {
SetDAC(2, analogVal);}   // Set the DAC 2 output
  else if (strcmp(Command, "dac3") == 0  && outparameter ) {
SetDAC(3, analogVal);}   // Set the DAC 3 output
  //Sine Wave Outputs
  else if (strcmp(Command, "sine0") == 0  && outparameter ) { sine(0,
analogVal);}    // Set the DAC 3 output
  else if (strcmp(Command, "sine1") == 0  && outparameter ) { sine(1,
analogVal);}   // Set the DAC 3 output
  else if (strcmp(Command, "sine2") == 0  && outparameter ) { sine(2,
analogVal);}   // Set the DAC 3 output
  else if (strcmp(Command, "sine3") == 0  && outparameter ) { sine(3,
analogVal);}   // Set the DAC 3 output
  // UNIPOLAR
  else if (strcmp(Command, "upA") == 0 ) { Serial.println("Setting
UniPolar  Group A");  DAC_GPIO0 = 1; InitDAC(); }
  else if (strcmp(Command, "upB") == 0 ) { Serial.println("Setting
UniPolar  Group B");  DAC_GPIO1 = 1; InitDAC(); }
  // BIPOLAR
  else if (strcmp(Command, "bpA") == 0 ) { Serial.println("Setting
BiPolar  Group A");   DAC_GPIO0 = 0; InitDAC();  }
  else if (strcmp(Command, "bpB") == 0 ) { Serial.println("Setting
BiPolar  Group B");   DAC_GPIO1 = 0; InitDAC();  }
  // Power Down
  else if (strcmp(Command, "offA") == 0 ) {Serial.println("GrpA Off
"); DAC_PD_A = 1; InitDAC(); }
  else if (strcmp(Command, "offB") == 0 ) {Serial.println("GrpB Off
"); DAC_PD_B = 1; InitDAC(); }
  // Power Up
  else if (strcmp(Command, "onA") == 0 ) { Serial.println("GrpA On");
DAC_PD_A = 0; InitDAC(); }
  else if (strcmp(Command, "onB") == 0 ) { Serial.println("GrpB On");
DAC_PD_B = 0; InitDAC(); }
  // Catch All
  else {  Serial.print("Error "); Serial.println(commandBuffer); }
  return true;
}
```

```
// Utility function to make sure the string is a numneric one
int isNumeric (const char * s)
{
  while (*s)
   {
    if (!isdigit(*s)) return 0;
     s++;
   }
  return 1;
}
```

Code 3 DAC8734 Code [27]

Appendix 2: Schematics

1 Schematic

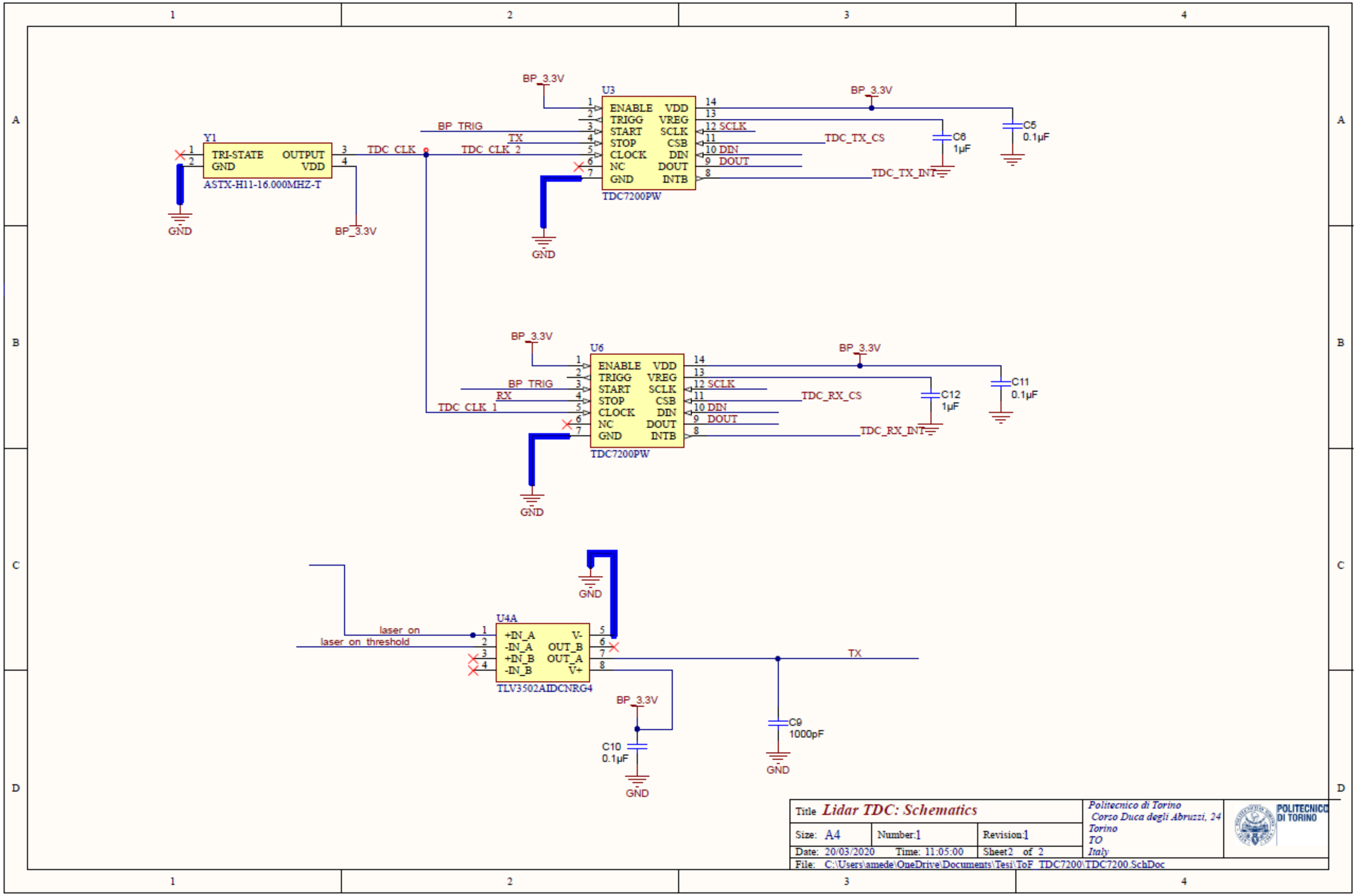


Plate 1: Schematic

2 Connectors

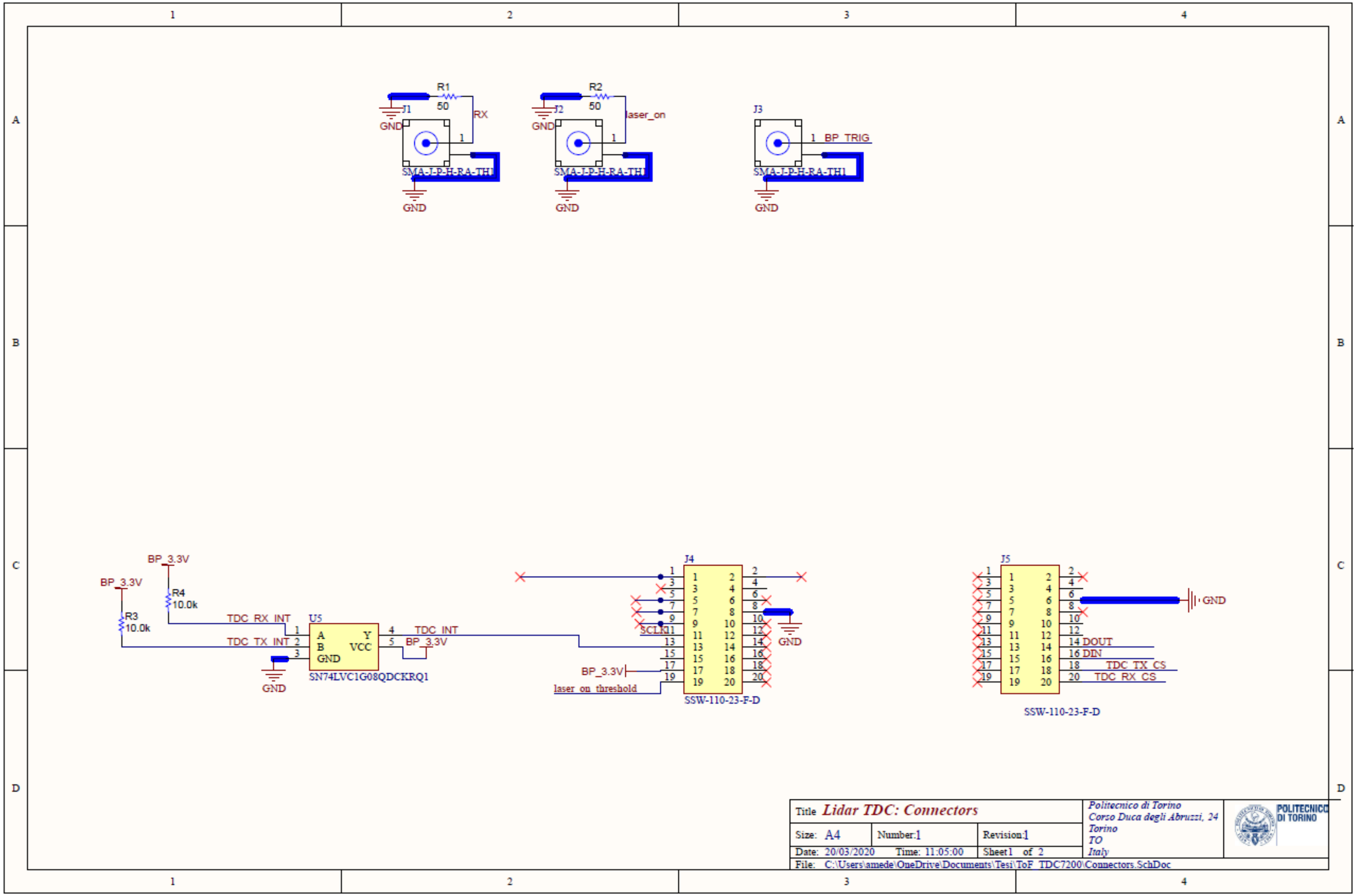


Plate 2: Connectors

Bibliography

[1] D. Guidry, *LIDAR-Pulsed Time-of-Flight Reference Design Using High-Speed Data Converters*, 2016.

[2] L. Bardelli, G. Poggi, M. Bini, G. Pasquali and N. Taccetti, "Time measurements by means of digital sampling techniques: a study case of 100ps FWHM time resolution with a 100MSample/s, 12bit digitizer," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 521, pp. 480-492, 2004.

[3] R. Hui and M. O'Sullivan, "Chapter 2 - Basic Instrumentation for Optical Measurement," in *Fiber Optic Measurement Techniques*, R. Hui and M. O'Sullivan, Eds., Boston, Academic Press, 2009, pp. 129-258.

[4] D. Paschotta, *Laser Safety*, 2020.

[5] K. Barat, Laser Safety in the Lab, Bellingham: Society of Photo-Optical Instrumentation Engineers (SPIE), 2012.

[6] "Laser Safety," [Online]. Available: https://en.wikipedia.org/wiki/Laser_safety.

[7] "Maximum permissible exposure (mpe) and nominal hazard zone (nhz)," 2020. [Online]. Available: https://www.smu.edu/BusinessFinance/RiskManagement/Health-Safety/ResearchSafety/LaserSafety/MPE-and-NHZ.

[8] P. C. D. Hobbs, Building Electro-Optical Systems: Making It All Work., 2nd ed.. ed., 2009.

[9] D. Amanor, W. Edmonson and A. Anyanhun, "Visible Light Communication System for Inter-satellite Communication of Small Satellites," 2016.

[10] "Photodiodes," [Online]. Available: https://learnabout-electronics.org/Semiconductors/diodes_27.php.

[11] S. B. Alexander, Optical communication receiver design / Stephen B. Alexander, Bellingham London: SPIE Optical Engineering Press IEE, 1997.

[12] "Photodetectors for LiDAR," [Online]. Available: https://www.hamamatsu.com/resources/pdf/ssd/Photodetector_lidar_kapd0005e.pdf.

[13] "PDA10CS(-EC) InGaAs Switchable Gain Detector," [Online]. Available: https://www.thorlabs.com/drawings/a92ba9ad57b5de51-0CA30886-054E-722B-F8F6B92AEC8A6841/PDA10CS-Manual.pdf.

[14] "GVS001 and GVS002 GVS101 and GVS102 GVS201 and GVS202 GVS301 and GVS302 Scanning Galvo Systems User Guide," [Online]. Available: https://www.thorlabs.com/drawings/da7b4ab3883e569d-2F4D4D89-F0F8-C4E5-9C1282BC0CFE37C4/GVSM002-EC_M-GalvanometerManual.pdf.

[15] "Quad, 16-Bit, High-Accuracy, ±16V Output, Serial Input DIGITAL-TO-ANALOG CONVERTER," 2009. [Online]. Available: http://www.ti.com/lit/ds/sbas465a/sbas465a.pdf.

[16] "DAC8734EVM User's Guide," 9 2009. [Online]. Available: http://www.ti.com/lit/ug/sbau156b/sbau156b.pdf.

[17] "8-/6-/4-Channel DAS with 16-Bit, Bipolar Input, Simultaneous Sampling ADC Data Sheet AD7606/AD7606-6/AD7606-4," 2012-2018. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ad7606_7606-6_7606-4.pdf.

[18] "TDC7200 Time-to-Digital Converter for Time-of-Flight Applications in LIDAR, Magnetostrictive and Flow Meters," 2015-2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/tdc7200.pdf.

[19] SiTime, " Clock JItter and Measurement," 2 6 February 2009.

[20] *Phase-locked loops in an IC-based clock distribution system: Part 2 – Phase noise.*

[21] D. W. Allan, "Conversion of Frequency Stability Measures from the Time-domain to the Frequency domain, vice-versa and Power-law Spectral Densities," 2012.

[22] "TLV3502-Q1, 4.5-ns Rail-to-Rail High-Speed Comparator," 2010-2014. [Online]. Available: http://www.ti.com/lit/ds/sbos507a/sbos507a.pdf.

[23] "SN74LVC1G08-Q1 Single 2-input positive-AND gate," 2004-2019. [Online]. Available: http://www.ti.com/lit/ds/symlink/sn74lvc1g08-q1.pdf.

[24] B. F. Logan, " Information in the Zero Crossings of Bandpass Signals," *The Bell System Technical Journal,* vol. 56, 4 April 1977.

[25] G. C. Matthieu Chevrier, " How to build a LIDAR system with a time-to-digital converter," *Analog Applications Journal*, 2017.

[26] M. Chevrier, *LIDAR Pulsed Time of Flight Reference Design,* Texas Instruments Incorporated, 2016.

[27] P. Oakes, "Arduino and how to drive the DAC8734 from TI (AKA the DAC8734EVM Road test Board)," [Online]. Available: https://www.element14.com/community/community/arduino/blog/2015/07/28/arduino-and-how-to-drive-the-dac8734-from-ti-aka-the-dac8734evm-road-test-board.

[28] P. Toliver, I. Ozdur, A. Agarwal and T. K. Woodward, "Comparison of LIDAR system performance for alternative single-mode receiver architectures: modeling and experimental validation," in *Laser Radar Technology and Applications XVIII*, 2013.

[29] K. Schröder, Handbook on Industrial Laser Safety, Techincal University of Vienna, 2000.

[30] T. Mirtchev and S. Minko, "Optimizing the Feedback Control of Galvo Scanners for Laser Manufacturing Systems," 6 2010.

[31] M. Lei, H. Li, Z. Lei and X. Liu, " Research on Detection Performance in Laser Ranging System," 2011.

[32] A. Krueger and S. White, *Diode-Pumped Lasers: Performance, Reliability Enhance Applications.*

[33] Z. Jiao, B. Liu, E. Liu and Y. Yue, "Low-Pass Parabolic FFT Filter for Airborne and Satellite Lidar Signal Processing," *Sensors (Basel, Switzerland),* vol. 15, pp. 26085-26095, 10 2015.

[34] E. Hergert, "A Guide to Choosing the Right Detector," 2015.

[35] D. Guidry, *LIDAR-Pulsed Time-of-Flight Reference Design Using High-Speed Data Converters,* Texas Instruments Incorporated, 2016.

[36] V. Giarola, "Advanced LiDAR Systems Design of a LiDAR platform," 2017-2018.

[37] M. Di Paolo Emilio, "Design PCB," 2016, pp. 55-65.

[38] L. Bardelli, G. Poggi, M. Bini, G. Pasquali and N. Taccetti, "Time measurements by means of digital sampling techniques: a study case of 100 ps FWHM time resolution with a 100 MSample/s, 12 bit digitizer," *Nuclear Instruments and Methods in Physics Research A,* vol. 521307080, 4 2004.

[39] K. Barat, Laser Safety in the Lab, Bellingham: Society of Photo-Optical Instrumentation Engineers (SPIE), 2012.

[40] J. S. S. T. Association, " Definition of Skew Specifications for Standard Logic Devices - JESD65B," 9 2013.

[41] B. Behroozpour, P. Sandborn, M. Wu and B. Boser, "Lidar System Architectures and Circuits," *IEEE Communications Magazine,* vol. 55, pp. 135-142, 10 2017.

[42] "Noise Equivalent Power (NEP)," [Online]. Available: https://www.eotech.com/121/noise-equivalent-power-(nep).

[43] "Diode Pumped Lasers Overview," [Online]. Available: http://www.aml.engineering.columbia.edu/ntm/level3/ch02/html/l3c02s01.html.