Mohamed HAGE HASSAN

Integrated Electronic Systems
2017/2018

FEMTO-ST Institute
15B, Avenue des Montboucons, 25030 Besançon

Sub-surface physical and chemical sensing: passive transducers acting as cooperative targets to Ground Penetrating RADAR

from 05/02/2018 to 06/07/2018

Under the supervision of :

- **FEMTO-ST supervisor :** Jean-Michel, FRIEDT, jean-michel.friedt@femto-st.fr
- **Phelma Tutor :** Béatrice, CABON, cabon@minatec.grenoble-inp.fr

Confidentiality : yes no

École nationale
supérieure de physique,
électronique, matériaux

Phelma
Bât. Grenoble INP - Minatec
3 Parvis Louis Néel - CS 50257
F-38016 Grenoble Cedex 01
Tél +33 (0)4 56 52 91 00
Fax +33 (0)4 56 52 91 03
http://phelma.grenoble-inp.fr

Acknowledgments

I would like first to thank my internship supervisor Jean-Michel FRIEDT for his hospitality, remarks and engagement through this learning experience, whose scientific curiosity and discussions are a source of inspiration for my intellect.

I would also like to thank David RABUS for his passionate participation and appreciation throughout this scientific project, and Gwenhaël GOAVEC-MÉROU for his excellent support and input, without which this endeavor might not have flourished.

I am also very thankful to the CoSyMA team for their welcome, as well anyone interested in the project.

Contents

List of Figures

Glossary

transducer device capable of transforming a physical quantity type into another.

RADAR RAdio **D**etection **A**nd **R**anging is a device using electromagnetic wave to probe, detect and examine the velocity and range of objects.

passive sensor a type of sensors specifically designed with no active elements.

real-time executive system a real-time executive/operating system executes operations with a real-time basis, without delays.

delay line a device introducing delays in the received electromagnetic wave.

piezoelectric Piezoelectricity is a physical phenomenon relating the transformation of electromagnetic energy into a mechanical one and vice-versa.

VNA **V**ector **N**etwork **A**nalyser is a device used in the characterization of radiofrequency components in terms of their S-parameters.

lithium niobate $LiNbO_3$ is a type of compounds combining niobium, lithium and oxygen, used extensively in integrated optics and piezoelectric sensors for its properties[28].

embedded software is a program designed to run in embedded architectures, such as the ARM cortex microprocessor.

stroboscopy is a phenomenon caused by aliasing, which occurs when the sampling frequency is much less than the sampled high frequency signals.

STM32F334 is a microcontroller of the STM32 family, designed by STMicroelectronics and based on the ARM Cortex 32-bit microprocessor architecture.

Direct Digital Synthesizer a device capable of generating frequency and phase-tunable arbitrary wave forms using digital data processing blocks, referenced to a fixed frequency internal high precision clock source[29].

avalanche breakdown a physical phenomenon occurring in semiconductor devices, allowing the effect of current multiplication, and is exploited in avalanche transistors[30].

Serial Peripheral Interface a synchronous bus capable of sending and receiving data simultaneously, to a single or multiple devices, using the Master-Slave scheme.

SRAM Static **R**andom **A**ccess **M**emory is a type of volatile memory using latches (bistatic flip-flops) to store information.

FT232R a integrated circuit designed by FTDI to allow a conversion between USB and serial protocols[31].

ARM cortex-m4 a microprocessor based on the RISC (Reduced Instruction Set Computer) architecture, designed by ARM for embedded applications.

instruction set a set of instructions that are recognized and executable by a specific processor.

baremetal a program running directly on a microcontroller without the need for abstraction layers (operating system).

monolithic composed of a single indivisible part, and having all the necessary elements to function.

POSIX Portable **O**perating **S**ystem **I**nterface is a standard designed to allow portability of code and compatibility of computer systems, designed by IEEE Computer Society.

ANSI the C standard ratified by the American National Standards Institute.

Application Programming Interface a set of functions, protocols and instructions to build up software.

pthreads POSIX threads, an execution model within the POSIX framework to allow parallel computing .

daemons background applications running on an operating system.

Pulse Width Modulation a method to modulate the duty cycle of a signal.

General Purpose Input Output controllable pin with a general purpose during embedded software runtime, allowing for an output or input functionality on a microcontroller.

USART Universal Synchronous and Asynchronous Receiver-Transmitter is a serial interface that can be programmed to function in a synchronous or asynchronous manner.

FEMTO-ST Institute : Time-Frequency department Presentation

The FEMTO-ST Institute (Franche-Comté Electronique Mécanique Thermique et Optique Sciences et Technologie) established on 1st January 2004 with the merger of five Franche-Comté laboratories, is a mixed research unit associated with French National Center for the Scientific Research (CNRS), and attached to the University of Franche-Comté (UFC), École Nationale Supérieure de Mécanique et des Microtechniques (ENSMM) and University of Technology of Belfort-Montbéliard (UTBM).

The main institute mission is to find novel ways to associate information science and technology with engineering sciences. Its areas of research are diversified, and associate to acoustics, micro-nanosystems, optics, time-frequency, mechatronics. The laboratory currently houses 7 departments : AS2M (Automatic Control), DISC (Computer Science), Energy, Applied Mechanics, Micro Nano Sciences and Systems, Optics and Time-Frequency.

FEMTO-ST is a significantly large laboratory, with more than 700 permanent staff in the scientific departments and common services.

Time-Frequency[1] is a department of the FEMTO-ST institute, with a main objective focused on creating ultra-stable resonators and oscillators, sensing applications and metrology of time-frequency sources.

Time-Frequency is also a highly diversified department, combining a multitude of scientific approaches, like electronics, photonics, acoustics, Micro-Electro-Mechanical Systems (MEMS), signal processing and atomic physics.

The department is composed of 3 teams : ACEPI (Acousto-électronique et Piézoélectricité), focused mainly on oscillators, piezoelectric resonators, and characterization of time-frequency devices, OHMS (Ondes, Horloges, Métrologie et Systèmes), with activities focused on microwave sources and photonics. Finally, the CoSyMA team (Composants et Systèmes Micro-Acoustiques) with research based on MEMS-piezoelectric resonators and resonating sensors in general.

The present study is held within the CoSyMA team and funded by the ANR UNDERGROUND[3] grant.

Introduction

Global sensor markets are rapidly expanding, with current estimation of 7.20% growth in revenue over the period of 2018-2023[2]. This is largely due to increasing requirement for monitoring hazardous risks, and increasing demands to eliminate potential problems residing in industrial zones, especially throughout chemical sensing.

Wireless sensor networks represent the core of such expansion: the need for sustainable capability to probe environments remotely has lead to the adoption of passive wireless sensors, especially Surface Acoustic Waves (SAW) **transducers**, developed by SENSeOR SAS, which has demonstrated their ability to act as passive sensors for short-range based **RADAR** systems.

The UNDERGROUND ANR[3] project has therefore been deployed to meet technological challenges required for the successful deployment of **wireless passive sensors** in underground conditions, and the ability to probe such sensors using available technology, such as Ground Penetrating RADAR (GPR).

Extensive experimentation with currently available commercial GPRs[4][5][6], such as the Malå Geoscience's ProEx GPR, confirmed their ability to probe wireless SAW transducers. However, the GPR's local oscillator instability has lead to timing drift, which renders the interrogation of SAW passive sensors practically impossible.

It is deemed necessary to develop specific GPR architecture, which could eliminate such drift problem: the CoSyMA team of the FEMTO-ST Institute, in a joint laboratory approach with SENSeOR SAS, PhASES, has developed a dual-chip pulsed stroboscopic RADAR[7] capable of probing SAW transducers acting as cooperative targets, running on the basis of the STM32F334 micro-controller, selected for its high resolution timer.

The main objectives of this internship are: to develop the embedded software of the RADAR, validate the electronic architecture of receiver and the pulsed emitter, establish a solid radio-frequency link for the underground probing, assemble a RADAR prototype for measurements, and develop signal processing algorithms to validate the ability to measure physical quantities, such as temperature or stress.

The first part introduces the general context of the internship, a systemic approach to the RADAR architecture, probing methodologies and therefore establish the necessary specifications to accomplish. The second part examines potential solutions for the embedded software development, and an introduction to the NuttX **real-time executive system**, and the development made under such platform. The electronic implementation is then introduced, analyzed, and validated. An early prototype of the RADAR is discussed later with signal processing algorithms in mind, and the execution of a series of measurements of temperature variations.

1 General Context

1.1 Wireless passive SAW transducers

General description of surface acoustic wave transducers, their theory of operation, and capability as a passive wireless sensors of physical properties:

The ability to address sensing elements over long durations has led to an increasing interest in passive elements. Acoustic wave transducers fit into the area of completely passive sensors: their ability to reflect incoming electromagnetic energy however minute the received power, and survivability in extreme circumstances, and relatively simple architecture contribute to their major advantages[6].

Basic theory of operation:
Surface acoustic wave transducers acting as wireless passive sensors are essentially used as **delay lines**: an electromagnetic impulse is generated with a probing mechanism - interrogator, or RADAR - and usually, delay lines are used to delay the signal from the surrounding clutter, which represents unwanted reflected echoes of the interrogator with an interface. While dielectric delay lines can delay such signal beyond clutter, their extended geometric properties, which depends on the electromagnetic wave propagation path, prohibits their usage in an effective manner [6]. A more suitable approach, which exists in acoustic wave transducers, is the usage of **piezoelectric** substrate: an incoming electromagnetic wave is converted to a surface-propagating mechanical wave, at the air-substrate interface which reduce the wave's speed by a factor of 10^5[4], thus reducing the propagation path of the wave, hence the compact sensor dimensions.

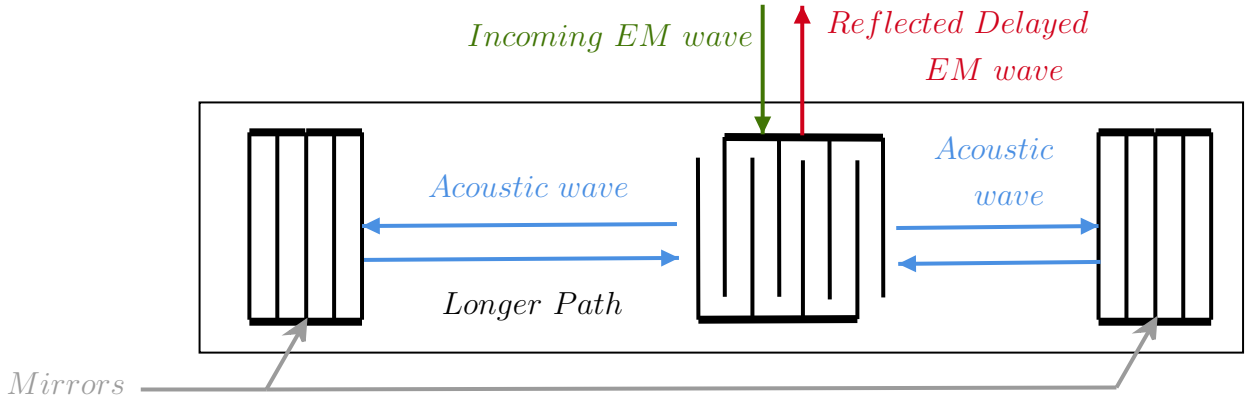

Figure 1: A schematic simplifying the architecture of the surface acoustic wave transducer

Fig. 1 shows the basic architecture of the SAW sensor: A thin layer of piezoelectric substrate is deployed, which contains interdigitated interface, which converts an incoming electromagnetic wave into a surface propagating mechanical wave, which travels on **2 directions**, one of which is longer than the other. The two mechanical pulses are then reflected at the mirrors, then travels back to be converted to electromagnetic waves at the interface. Typical time delays in the microsecond range require a propagation path in the mm range, considering the 3000 m/s acoustic wave velocity.
An important aspect to consider is the surface-propagating wave's velocity dependence on physical properties, such as stress or temperature[4]. **Such dependence** represents the sensing capability of the sensor. A differential approach, such as the cross-correlation algorithm,

discussed in section. 4.2, can be used to get an accurate estimate of the wave velocity from the 2 reflected echoes[4].

A characterization of a SAW transducer, designed as a GPR target for underground measurements is seen in figure 2. while being connected directly to Port 1 of the **VNA**. The sensor should be energetically coupled correctly, with a bandwidth of 20 MHz set around 95 MHz. Here we see a sweep from 85 MHz to 104 MHz, and the attenuation effects of the sensors can be clearly seen in this bandwidth, for the first trace.

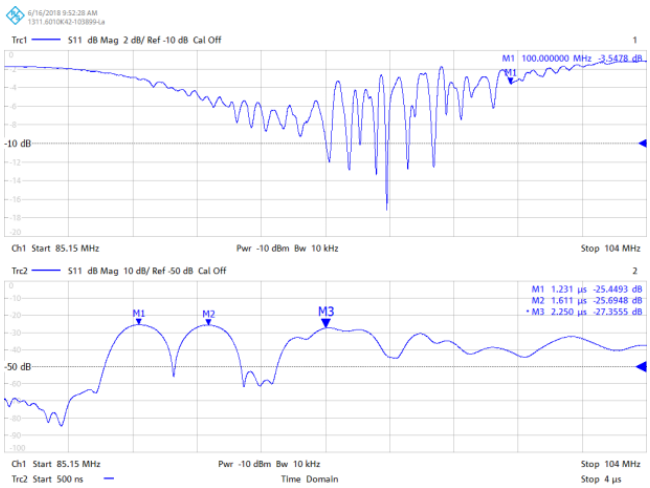

Figure 2: S_{11} characterization of a SAW sensor with Rohde & Schwarz ZNB8 Vector Network Analyzer, and the time domain analysis of S_{11}

The second trace represent the inverse Fourier transform of the first one, which is the time domain analysis of the S_{11}. We can clearly see the formation of 3 lobes, **which correspond** to the delay introduced by the sensor when reflecting the incoming electromagnetic wave. Here the delays are set at $1.2\mu s$, $1.6\mu s$ and $2.2\mu s$ respectively.

The characterization of the sensor is crucial for the successful deployment of probing strategies, which will be discussed in the following section.

1.2 Probing methodologies and the stroboscopic method

The time base drift of analog timing mechanisms in commercial GPRs, such as the Malå ProEX, even if they are capable of exciting SAW sensors, makes them unsuitable for recovering temperature measurements[4]:

Taking temperature as the physical quantity under measurement, change is induced to the acoustic wave velocity propagating through the piezoelectric substrate in the sensor: this defines the targeted time delay accuracy. The piezoelectric substrate used in the design of the SAW sensor, is **lithium niobate** with a YXI/128° cut, which generally has a temperature sensitivity of 70 ppm/K (part per million per Kelvin)[6]. For the this sensitivity, 1 kelvin temperature resolution, and a 400 ns difference between the 2 echoes, this accuracy is 28 ps, details are discussed in annex A.1. The Malå GPR's time base drift is of the orders of $4ns$, which renders the recovery of the physical quantity under investigation useless[6].

Our strategy envisioned has been to design a new timebase generator for the commercial instrument. While successful, this strategy cannot be widely deployed and does not meet additional needs such as low power consumption and **embedded software** dedicated to sensor probing.

Alternative approaches have been investigated by the CoSyma team, which requires the implementation of GPRs specifically designed for interrogating the SAW sensors, providing accuracy within the $28ps$ range. This latter approach is the topic of this work.

1.2.1 Stroboscopy

In both methods implemented by the team, **stroboscopy** resides at the core of the sampling mechanism. Instead of using expensive radiofrequency analog-to-digital converters (ADC) to sample the acquired signal at high resolution, a fast high resolution track and hold with a combination of a slower ADC can achieve similar performance using the stroboscopic method[7], represented in figure 3.

To achieve a stroboscopic sampling, we need a device capable of generating a high resolution signal, that can be used as a track and hold. While some solutions exists on the market[7], an excellent candidate for this task is the usage of the **STM32F334 microcontroller** chip: this microcontroller can provide a fine timer resolution up to $217ps$ with the High Resolution TIMer (HRTIM).

For the SAW sensors, it is mentioned that we need maximum energetic coupling with electromagnetic pulses with a bandwidth of 20 MHz and a center frequency of 95 MHz.

The stroboscopic method is applied through acquiring the reflected 95 MHz signal from the sensor, acquiring samples using the slow ADC, and incrementing the track and hold instant of the ADC by $217ps$ on each period. A high bandwidth track and hold ADC needs to be chosen for this task.

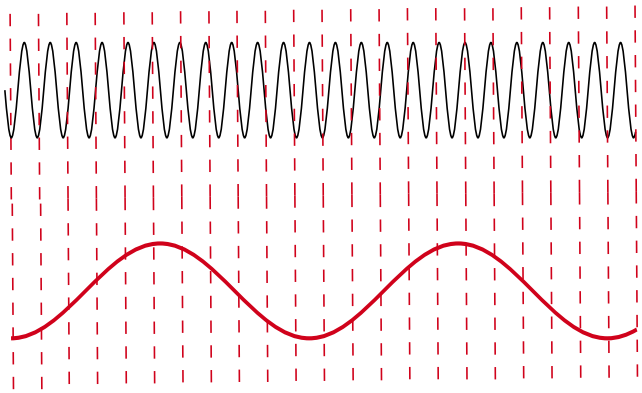

Figure 3: A fast track and hold sampling mechanism, illustrating stroboscopy: if we increment the track and hold instant by a small amount, and recovering a point from the black periodic signal on each measure, we can reconstruct a high frequency signal (Red signal).

By using the STM32F334 HRTIM capability, we reach $\frac{1}{217\times10^{-12}} = 4.6$ GSample/s equivalent sampling rate.
A strong hypothesis, however, needs to be verified when admitting this approach: **Signal stationarity**. The signal should be repeated and independent of the physical quantity variation (such as temperature) for the reconstruction of the high frequency signal, within **the sampling time frame**.

1.2.2 DDS based approach

A first approach[7] (figure 4) is made using a **Direct Digital Synthesizer** to generate a radiofrequency signal at the SAW sensor operating frequency.

The STM32F334 microcontroller is programmed to provide the DDS with a time basis, synchronize it with a radiofrequency switch, which commutes between sending the generated carrier and acquiring the subsequent echoes from the sensor, and stores the necessary information using the stroboscopic approach on the acquired signal. This method is proved to be effective in recovering the SAW sensor echoes, yet lacks the necessary power to probe buried sensors, since the amplitude generated by the DDS is relatively weak for propagation. The issue with a radiofrequency generator amplified by a linear amplifier is the high mean power consumption

of the amplifier to reach peak power in the tens to hundreds of watts as will be tackled in the next section.

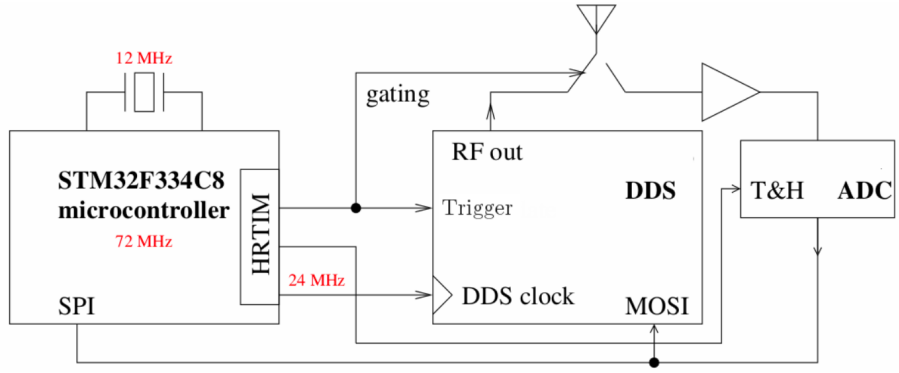

Figure 4: First approach ground penetrating RADAR architecture. Image is taken from the author[7] (The 24 MHz signal is used as a reference signal).

1.2.3 Pulsed RADAR

Another approach[8] is the usage of a pulsed emitter system, with a combination of the HRTIM track and hold timer, a high bandwidth track and hold slow ADC, and a radiofrequency switch, all of which are controlled by the STM32F334 microcontroller.

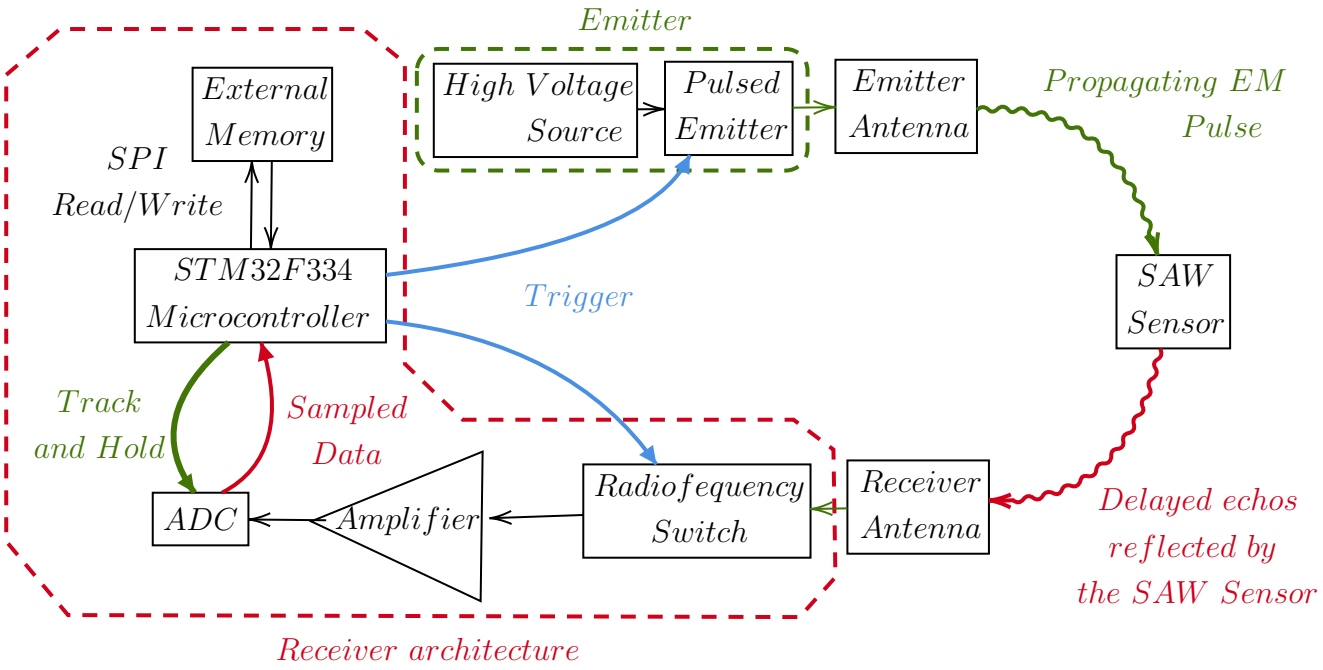

Figure 5: Architecture of the implemented RADAR

The pulsed emitter: A generator needs to be conceived, capable of charging up for a few microseconds yet discharging rapidly : the duty cycle of such emitter is low. The previous DDS method suffered from low output amplitude, the pulsed system needs to emit a highly amplified signal. The best candidate suited for this operation is the avalanche bipolar transistor : When a high voltage is applied on its collector, the transistor is capable of drawing up current in a few nanoseconds generating a high level pulse on its emitter in a phenomenon called "the

avalanche breakdown". Based on previous work, a period of $14\mu s$ is ideally suited for GPR measurements [7].

Theory of operation: an embedded C software runs on the STM32F334 microcontroller, executes commands to probe a sensor, and generating two high precision timer signals (217 ps) with a period of $14\mu s$, which represents the delay between each pulsed emission. The first HRTIM signal, with a duty cycle of $\frac{0.5\mu s}{14\mu s} = 3.5\%$, triggers the avalanche-transistor based RADAR emitter, which requires a high voltage source, and the radiofrequency switch at **the same time**, thus sending an electromagnetic signal to the sensor. The switch isolate the receiver circuit from the RADAR clutter, which corresponds to $0.5\mu s$ after the rising edge of the trigger HRTIM, then allows signals to pass, when the trigger signal is off.

Since the SAW sensor acts as a delay line, it sends 2 echoes of interest well beyond clutter after $1.2\mu s$ and $1.6\mu s$, thus this returned signal gets amplified, and sampled by the stroboscopic method with the second HRTIM signal acting as a high resolution track and hold, with a unitary step of 217 ps, thus allowing equivalent sampling rate of 4.6 GSample/s.

The microcontroller time reference is generated by an external quartz crystal, which establish a relatively stable timebase for this operation. The two HRTIM signals are synchronized with the emitted pulse: since our echoes resides between $0.9\mu s$ and $2\mu s$, the track and hold signal starts sweeping at $0.9\mu s$ with a step of $217ps$, which is incremented **on each HRTIM period**. For an interval of $2\mu s - 0.9\mu s = 1.1\mu s$, we need $1.1\mu s/217ps = 5070$ HRTIM periods, or in other terms, stroboscopic points.

It is established that the ideal HRTIM period is at $14\mu s$ [7], therefore, a total measurement time of the reflected echoes is of $14\mu s \times 5070 = 70.9ms$. For such short time, the variation of physical quantities for buried sensors is weak to be detected. The signal remains stationary enough within this time frame to reconstruct the signal with the stroboscopic method.

An in depth look into the electronic implementation, analysis and functionality of this architecture is discussed in section 3.

1.3 Required specifications

The general context has been established, dealing with the basic inner workings of the SAW transducers, acting as cooperative wireless sensors, the stroboscopic method, and a systemic approach to viable probing methodologies and their emitter/receiver architecture.
We can therefore deploy a list of specifications to be met for this internship:

- Develop a versatile embedded software running on the STM32F334 microcontroller, which is capable to achieve the required timer resolution (217 ps), and maintain synchronization between the 2 different HRTIM clocks.
- Analyze the receiver and emitter architecture, and optimize their configuration for optimal radiofrequency link.
- Design the emitter and receiver antennas, to achieve maximum electromagnetic coupling with the transducer around its operating frequency of 95 MHz.
- Introduce a viable implementation of the cross-correlation algorithm, which can extract the useful information out of the 2 returned echo signals, first offline and ideally online.
- Assemble a battery powered prototype of the ground penetrating RADAR for experimental measurements

- Execute a long term measurement for achieving the required precision within the 28 ps range in time delay accuracy.

The next section will discuss viable approaches in dealing with the embedded software development, the electronic implementation required for signal sampling with the stroboscopic method, and the experimental setup needed for the software development.

2 Embedded software development

2.1 Sampling electronics

For the embedded software development, a methodology has been implemented to setup a correct sampling method, in which only the sampling architecture is needed: we inject a known periodic signal, and the objective is to correctly sample it with an equivalent sampling rate of 4.6 GSample/s.

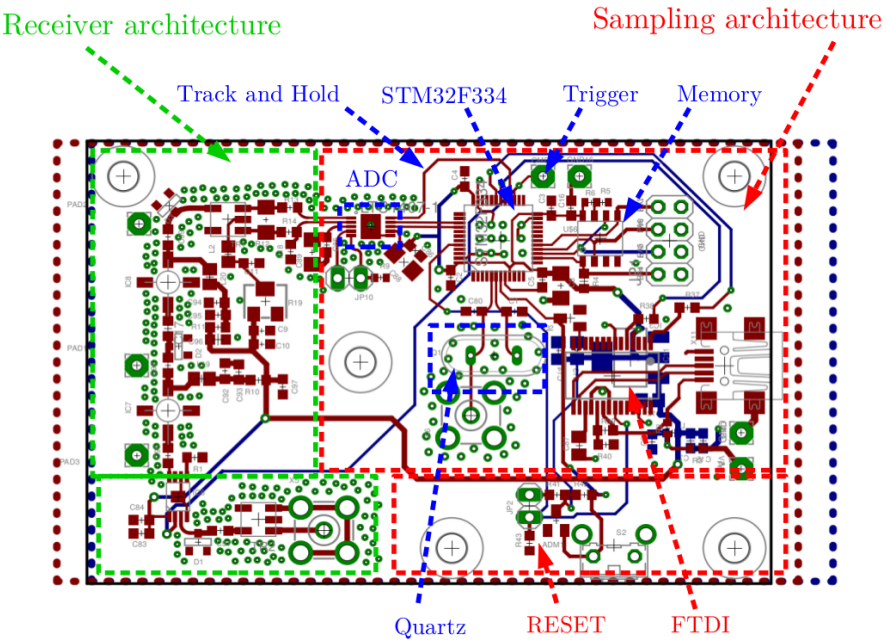

Figure 6: Electronic implementation of the stroboscopic RADAR architecture, developed by the CoSyma team: we are interested in setting up the sampling part for the embedded software development.

Figure 6 examines the sampling method in details:

1. A slow ADC yet with a high bandwidth track and hold capability needs to be used. Very few solutions exists on the market, but the most attractive choice is the LTC1407A[10] 14-bit sampling ADC. While the input frequency is limited at 1.5 MHz, the "start conversion signal" needed by the ADC is attached internally to a high performance track and hold functionality, capable of supporting up to 8 GHz in incoming start conversion signal frequency.
2. The signal is amplified by the receiver, which is fed into the LTC1407A[10] 14-bit sampling ADC: Only 1 of the 2 available differential channels are used, the other one is linked to ground. The input amplitude of the signal is regulated in the earlier stage for the

maximum input range of the ADC, which varies from -1.25 to $1.25V$ in differential mode.

The ADC, even if designed to operate ideally in the 100 kHz to 1 MHz range, can be used with the stroboscopic method in mind if the input signal frequency exceeds 10 MHz. As discussed before the stroboscopic method has a time step of $217ps$, with a period of $14\mu s$, which corresponds to a sampling frequency of $71.4KHz$, which resides well in the best performance characteristics, regarding the Effective number of bits (ENOB), Signal to Noise Ration (SNR), and total harmonic distortion (THD) (LTC1407A Datasheet[10]).

3. The track and hold signal, sent from the STM32F334 microcontroller, increments the sampling step by 217 ps each HRTIM period. The ADC then samples the incoming signal at $t = 0.9\mu s + n \times 217ps$, where $\mathbb{N}$ represent step number.
4. The ADC sends back the (n-1) sample to the STM32F334 microcontroller by the **Serial Peripheral Interface** (SPI) which stores it directly into the 23LC1024 **SRAM** memory chip[11], using the SPI bus.

The whole circuit is time referenced with a single 8 MHz quartz, which keeps a steady time base for the STM32F334. An **FTDI FT232R**[31] chip is present to provide the ability to connect to a laptop computer, with a mini-usb connection, and is used mainly to program the STM32F334 through a serial link. An experimental setup has been used to proceed with the development:

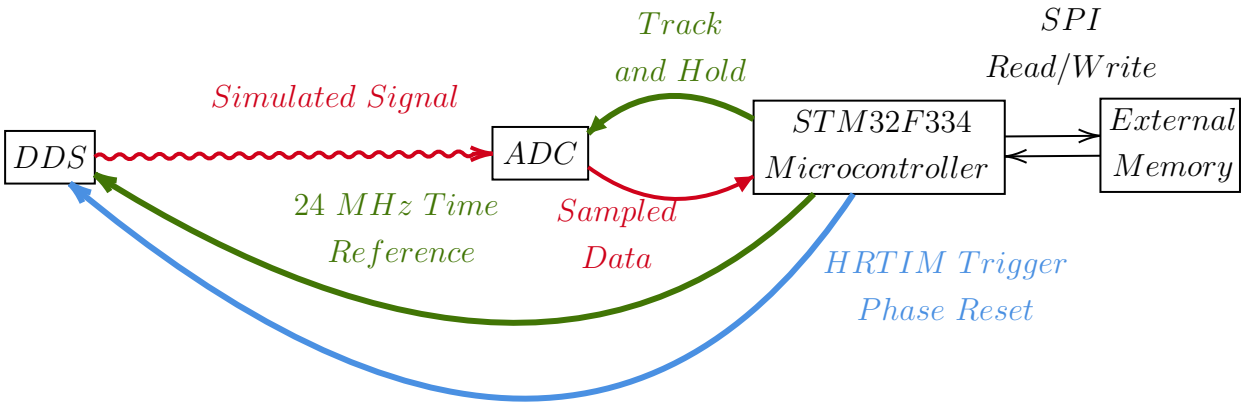

Figure 7: Experimental setup used to develop the sampling method within the STM32F334. A DDS is used to guarantee a synchronization between the 2 HRTIM signals and the radiofrequency signal attacking the ADC input

Figure 7 represent a DDS feeding a sinusoïdal radiofrequency signal to the ADC. For the pulsed RADAR, signal synchronization between the emitted pulse, the 2 HRTIM signals, and the received signal is guaranteed, since the STM32F334 is generating the timebase, and triggers the pulsed emitter exactly at the rising front of the HRTIM trigger signal. However, when using a DDS, a time reference should be generated with a 3rd 24 MHz HRTIM clock, and a phase reset on each HRTIM trigger: this ensures that the signal is synchronized with the high resolution timers.

2.2 Potential solutions

The STM32F334 microcontrollers are based on the **ARM cortex-m4** central processing unit, which can be programmed directly with an embedded application. The application is compiled with a specific **instruction set**, and can be executed without a main **operating system** running on the microcontroller. Such an application is called **baremetal**. For the DDS based approach, developed in section 1.2.2, a baremetal application has been developed controlling the STM32F334, which is capable of probing the sensors [7].

This approach suffers from low sustainability of the developed code: being a single **monolithic** application interfacing directly with the STM32F334 peripherals, the code is complex enough that only the author can successfully develop and maintain.

Another interesting choice is the deployment of a **real-time executive system**, which grants flexibility in developing the embedded software and versatility to deploy additional functionality provided by the system itself. The following section discusses the potential implementation of the NuttX executive system.

2.3 NuttX executive system

NuttX is a real-time executive system, aimed at compliance with the **POSIX** and **ANSI** standards[12]. It is used in real-time scenarios such as the controlling firmware of the PX4 drone system (pixhawk.org).
The system's compliance with the POSIX standard allows a very high flexibility in developing software, since the **Application Programming Interface (API)** is similar to the **Linux operating system**. It is essentially composed of user applications, a **middleware** layer, and a **low level driver** layer controlling the target microcontroller peripherals directly.

NuttX contains a significant amount of configuration files, each defining parameters for a specific type of hardware, such as STM32F1, STM32F4, STM32F334.

2.4 Development under NuttX

The NuttX executive system handles drivers and applications as follows:

1. **High level applications:** these are defined as user applications, can be integrated quite easily in NuttX, and can exploit full features of the OS: multitasking (**pthreads**), **daemons**, and can interface directly with the middleware of NuttX (Upper level drivers). A high level application can be programmed at this level to handle the RADAR configuration.

2. **Upper level drivers:** these drivers are defined to be used as middleware, can interface with the lower level drivers, which are board/architecture specific, and the upper level applications: contains multiple methods to control directly the lower level drivers and interface with the hardware. They are also divided into main sections: device drivers and bus drivers (like the SPI one).

3. **Lower level drivers:** these are the lowest form of drivers in NuttX, and can interface directly with the hardware. They are also architecture/board specific. Our driver implementation is handled here, and **interfaced with the high level application** through the usage of system wide **open, close, read, write, (poll), ioctl** system calls.

The advantage of using such system resides in the abstraction layers: once the low level driver layer has been developed, a specific API can be provided, and a sustainable development approach can be carried on with the high level applications.

Since our objective is to provide a manageable firmware implementation for the RADAR system, the embedded C programming should only be restricted to the High level application and low level drivers layers, without modifying the middleware of NuttX.

Plan: First the best practice development strategy aimed at separating userspace, driver and kernel layers is used. Following the conclusion that excessive latencies are introduced by the multiple abstraction layers, a full low-level implementation at the driver (interrupt) level is proposed to solve the targeted acquisition rate, and shown to be successfully implemented despite breaking the abstraction layer definitions.

2.4.1 High resolution timers

The STM32F334 microcontroller contains a high resolution timer, composed of 6 16-bit timing units, each with its independent counter and 4 compare units, and works on 2 channels[13]. For our purpose, we need to generate 3 independent digital signals for trigger, track and hold and the DDS 24 MHz reference timing. Under NuttX, this is implemented in a low level driver, which controls the hardware peripherals directly.

The trigger and track and hold signals are both linked to the HRTIM D timer: The trigger runs on the first channel, the ADC track and hold runs on the second channel. The periodicity of the channels are common yet the duty cycle of each can be configured at will.

PWM Generation:
The HRTIM signals are essentially **Pulse Width Modulation** signals, which are generated by the STM32F334 using a counter and compare functions: a local counter increments till it reaches a specified 16-bit integer period value P, then resets[14].

$$P = \frac{T_{counting}}{T_{highresolution}} \tag{1}$$

For our case, $P = 14\mu s/217ps = 64516$, this period is common for both HRTIM D signals. For

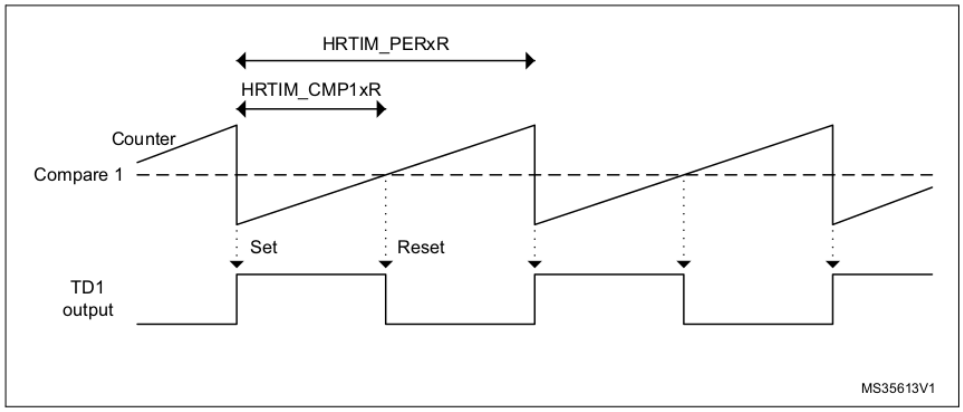

Figure 8: Basic functionality of the PWM generation with high resolution timers. Image is taken from[14].

the 24 MHz signal, we need a period of $41.6ns$, which corresponds to the integer value of 192.

The duty cycle of the timer can be set, by comparing the counter value to a specified duty cycle value, if reached, the comparator indicates that the threshold has been reached, and sets the output to 0, which is illustrated in figure 8.

For a duty cycle of $0.5\mu s$, which is indicated to be sufficient to eliminate the clutter as discussed in section 1.2.3, this corresponds to the integer value of 2304. The duty cycle of the DDS clocking signal needs to be as half the period, passing a value of $192/2 = 96$ to for the counter.

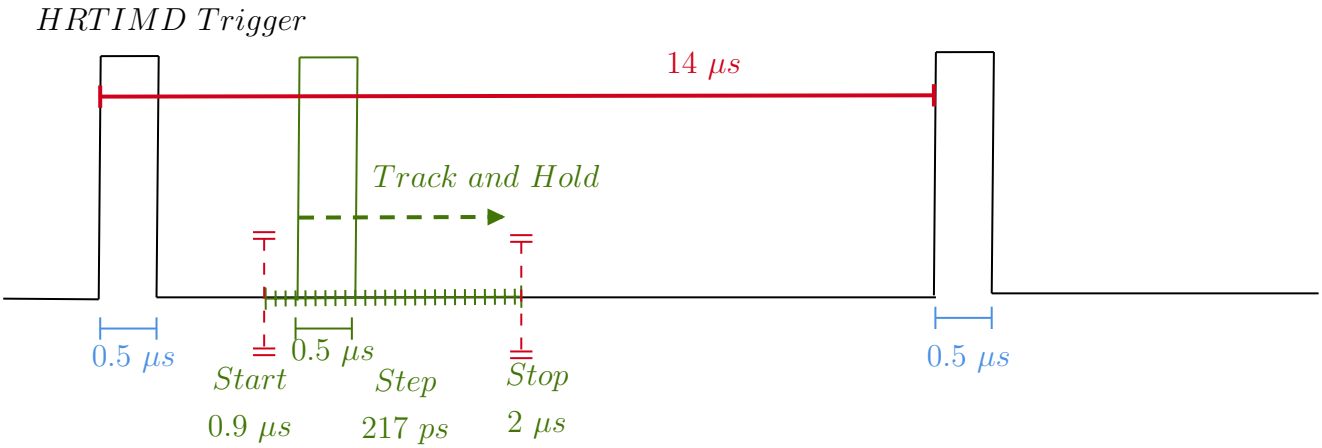

Figure 9: Illustration of the 2 High resolution signals, with the sampling track and hold signal sweeping from $0.9\mu s$ to $2\mu s$.

For the track and hold signal, we use 2 counters instead of a single one: the first counter sets the track and hold rising edge, and the second counter sets its falling edge. The 2 counters set positions, increment by the same value (217 ps) at the same time, which will create a dynamic (sweeping) track and hold signal, as presented in figure 9.

An example is included in the annex A.2 for the code in NuttX.

HRTIM Interrupts:
The HRTIM configuration is set with the usage of the **ioctl** functions, or input/output control functions. These are called from the application layer, which are then passed through the NuttX middleware to the lowest NuttX layer: the lower level drivers. In general, embedded microcontrollers like the STM32F334 have a set of **interrupts**, which are hardware flags, set at the occurrence of certain events, like the **rising edge** of the trigger and hold signal: they are sent to the processor to indicate that such an event requires immediate attention. Once an interrupt has been received by the processor, it needs to be **acknowledged** by an acknowledgment (ACK) signal, to indicate that the interruption has been successfully managed.

The usage of HRTIM interrupts is vital for our objective, and will be first discussed in section 2.4.3.

2.4.2 Sample acquisition and storage

A vital part of the sampling method is storing the sampled data in an external RAM, due to the low memory footprint of the STM32F334 (8 KBytes of **RAM** for the system).
Since the ADC implemented has a resolution of 14-bits, each sample received from the ADC

is stored in a 16-bit value, or 2 Bytes. It has been demonstrated that sampling the interval $[0.9\ ;\ 2\mu s]$ needs 5070 points, in section 1.2.3.

This means for a single coherent reconstruction of the radiofrequency signal, or one complete measurement, we need 10KBytes. The 23LC1024 memory has a 1 Mbit storage capacity[11], or 131 KBytes. In other terms, even with an external memory we are limited to $131/10 = 13$ measurements.

The issue will be resolved in the following sections, with the implementation of a commuting algorithm, which switch between measuring (or writing to memory), and reading, without exceeding the space limits.

Implementation:

NuttX needs to interface with the memory and the ADC using the SPI bus. Even if the STM32F334 has a single SPI bus, it has 2 sets of **General Purpose Input Output (GPIO)** pins. At NuttX initialization, the SPI bus configures the GPIOs for the external SRAM, which also enables the **chip select pin (CS)** for activating the SRAM when needed, by admitting a low impedance state[11]. This allows for selecting either the SRAM or the ADC, to write/read from.

The 23LC1024 SRAM works by[11] (illustrated by figure 10):

1. Select the appropriate operation mode before accessing the SRAM in either read or write operations: the sequential mode has been forced, which allows the memory to store incoming 8-bit samples as long as the CS is in low state.
2. Write an 8-bit word to the memory, by forcing the CS in low state, sending the "WRITE" command, sending the 24-bit address, and finally the 8-bit word.
3. The write ends when putting the CS in a high state

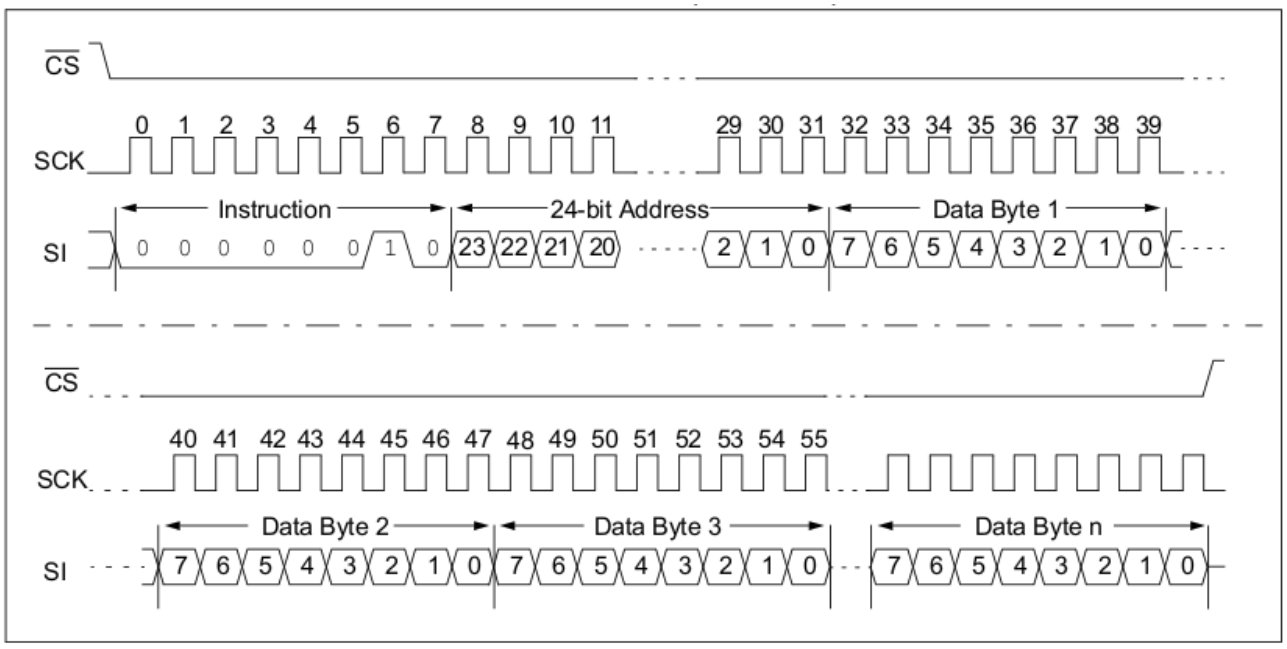

Figure 10: Sequential write sequence for the 23LC1024 memory. Image is taken from[11].

The read operation is similar, by sending the "READ" command instead of "WRITE" command.

For the LTC1407A 14-bit ADC, this device only sends 2 sets of 14-bit data if 32 SPI clock cycles have been seen by its SCK pin, after the rising edge of the track and hold signal[10]. It is important to notice that the acquisition of one sample needs to be limited to the time frame of $min([14 - 0.9\ ;\ 14 - 2\mu s])$ or $12\mu s$, for an acquisition each HRTIM period. The SPI bus frequency has been pushed to the maximum speed allowed by the memory, 20 MHz, which corresponds to a single SPI clock period of $50ns$.

The following algorithm has been implemented to achieve a sample acquisition and storage throughout the serial peripheral bus:

Algorithm 1: Sample acquisition and storage

Result: Store sample in memory
if *(Rising edge of track and hold)* **then**
    Acknowledge the HRTIM interruption;
    Send 16 SPI clock cycles;
    Store value from SPI into sample variable;
    Set memory CS low;
    Send WRITE instruction and the address **Most Significant Bits** (16 bits, 16 SPI clock cycles);
    Send memory address **Least Significant Bits**(16 bits, 16 SPI clock cycles);
    Send 2 8-bits words (16 bits, 16 SPI clock cycles);
    Set memory CS High;

With an SPI word fixed at 16-bits, Algorithm 1 shows that at least $16 \times 4 = 64$ SPI clock cycles are needed to complete the transfer. Since the SPI bus speed is fixed at 20 MHz, we have a total transfer time of $64 \times 50 \times 10^{-9} = 3.2\mu s$, which is well below the $12\mu s$ limit.

The usage of the NuttX API has proven unsuccessful: NuttX introduces large amount of delays due to numerous function calls executed while exchanging on the SPI bus. This will be discussed in details in the 2 following sections, with the introduction to the "polling method" and the subsequent low level driver methodology.

2.4.3 Polling method and timing limits

Our current model of the internal architecture, at least the one directly in relation with our implementation, is viewed as:

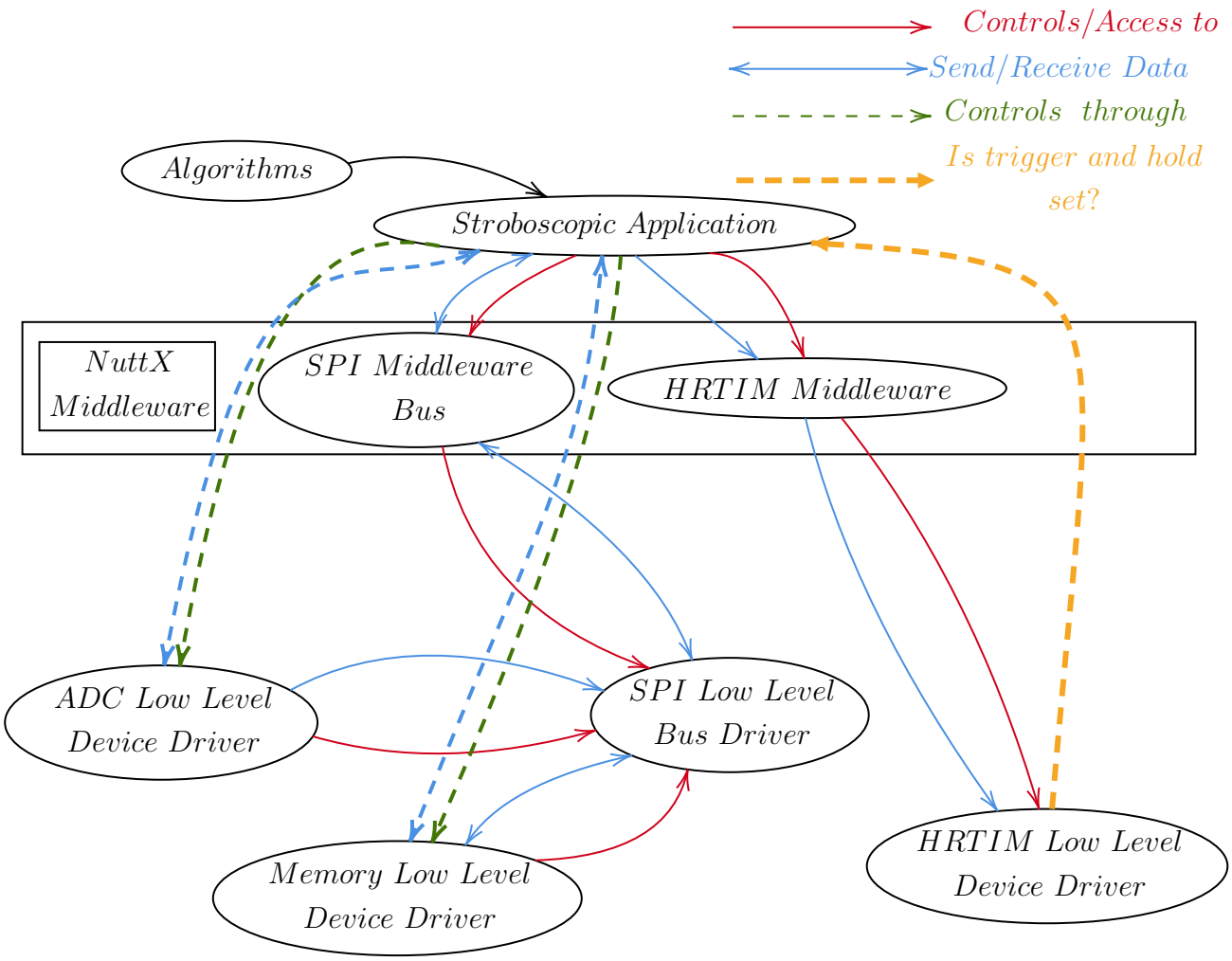

Figure 11: Hierarchical description of the NuttX implementation

Figure 11 constitutes a natural way for the implementation of the firmware: A high level "stroboscopic" application controls the low level (close to hardware interface), through the NuttX middleware layer.

Each system functionality, like the ADC, memory and HRTIM are divided into separate low level drivers, the ADC and memory drivers enables the communication with the physical world through the SPI low level driver, which is initialized by the middleware.

Each of the drivers implements **open, close, read, write and ioctl** system calls, which are called from the stroboscopic application, this allows the application to recover/write information from/to the drivers through the NuttX middleware.

To check if the rising edge of track and hold signal has occurred, the HRTIM low level driver continuously checks for **interrupt flag** corresponding to the rising edge, and sends the information back to the high level application: this methodology is called the **polling mode**.

Algorithm 1 can therefore be implemented in the application, to communicate with the drivers. A reformulation of this algorithm is examined in algorithm 2.

Algorithm 2: Sample acquisition and storage implemented in the application

Result: Store sample in memory
while *(ioctl(hrtim, CHECK_INTERRUPT) != 1)* **do**
  wait;
ioctl(hrtim, ACK);
read(adc, sample);

ioctl(memory, SET_CS, low);
write(memory, instruction_address_msb);
write(memory, address_lsb);
write(memory, sample);
ioctl(memory, SET_CS, high);

This implementation, however, suffers from latencies introduced by NuttX middleware: since the STM32F334 processor runs at a maximal frequency of 72 MHz, it can only execute 72 instructions in 72 clock cycles in $1\mu s$, assuming 1 instruction/cycle for such processor. In other terms, it has been observed that the system calls, such as **read, write, ioctl** take **significant time** for the critical time frame of $14\mu s$, greatly extending the SPI communication phase beyond $3.2\mu s$ calculated in the previous section.

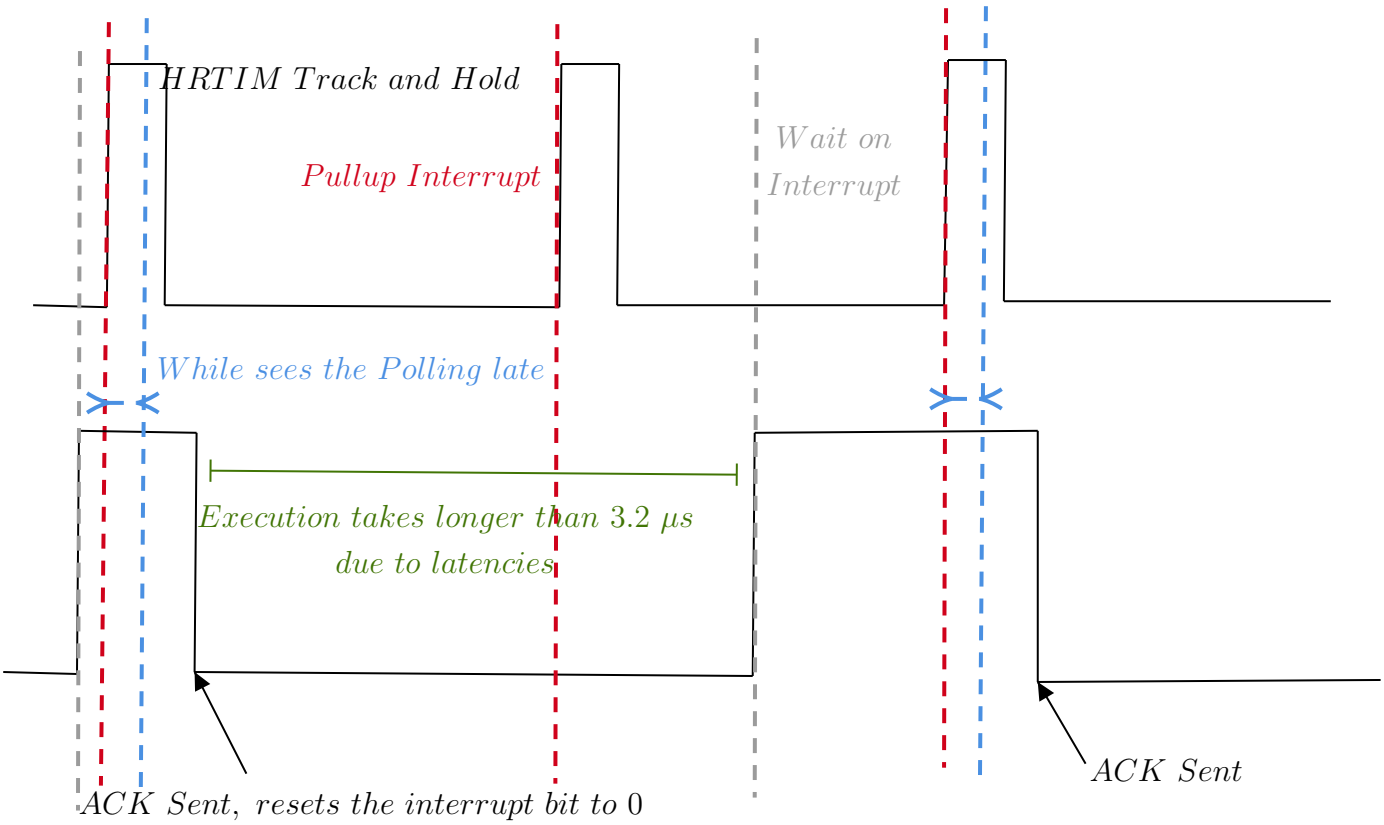

Figure 12: Latencies introduced by the NuttX middleware.

Figure 12 clarifies the effects of latencies on the sampling algorithm: delays in the detection of the rising edge interrupt, delays in the execution of signal acknowledgment, the read/write/ioctl

execution takes significant time larger than 14μs, thus causing the algorithm to miss points.

The solution of this problem is discussed in another look at the implementation, discussed in the following section.

2.4.4 Monolithic low level driver

As discussed in the previous section, latencies introduced by the NuttX middleware forbids the implementation of the algorithm at the high level layer. The solution is the implementation of algorithm 1 in a monolithic lower level, which has the fastest access to the hardware, as illustrated in figure 13.

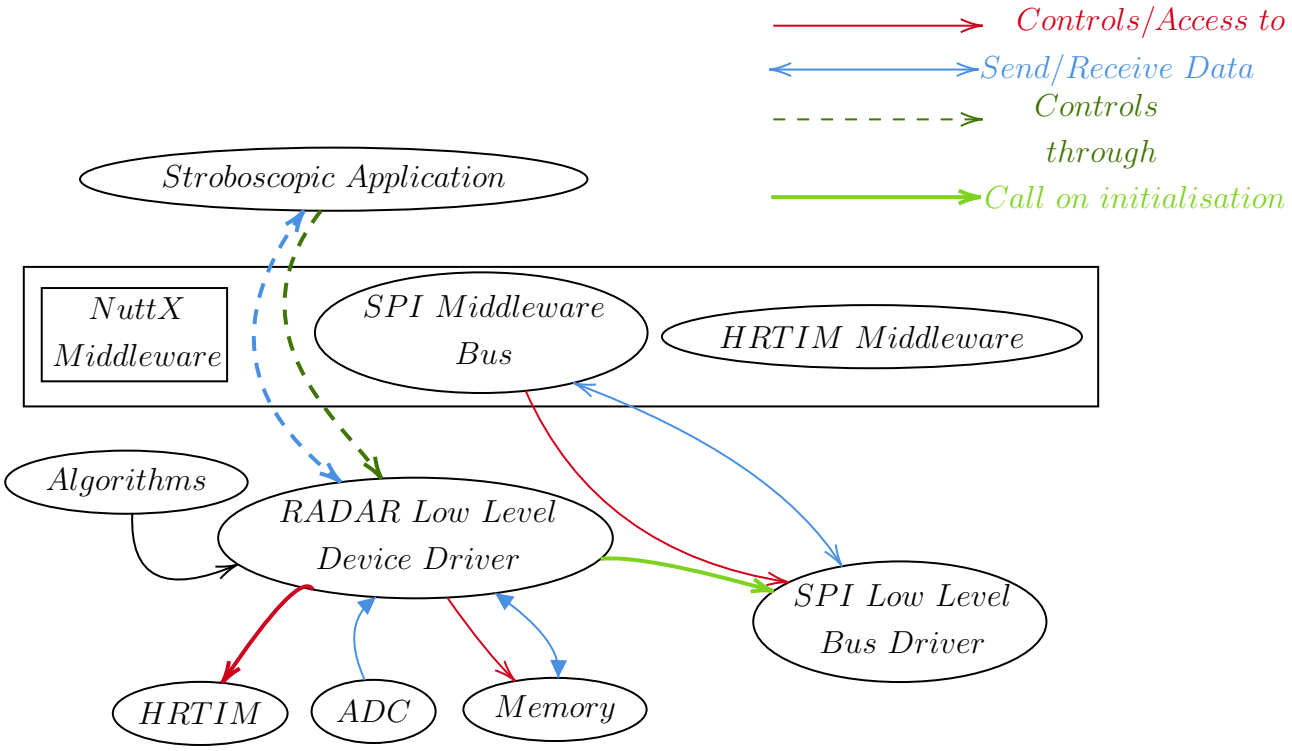

Figure 13: Hierarchical description of the adopted NuttX implementation. The common "RADAR" driver controls the communication with the hardware devices with direct read/write into the STM32F334 SPI registers.

The implementation of algorithm 1 with direct access to STM32F334 SPI registers provides the fastest possible way to communicate information between the ADC and the external SRAM.

Interrupt Service Routine:
Figure 12 shows that latencies introduced by NuttX while checking the interrupt bit through **ioctl**, delays our knowledge of the instant of the interrupt occurrence. Is it possible to start execution of algorithm 1 at the same instant of the rising edge of track and hold?

The solution is the implementation of the **Interrupt Service Routine (ISR)**, which handles the hardware interrupt made by the rising edge of track and hold. This represents a special function, that the processor will directly execute each time the rising edge interrupt is set to 1, saving its current state and restoring it later after the routine finishes.

Under NuttX, the ISR is implemented for all the timers and advanced timers, except the HRTIM ones.

For NuttX, all the hardware interrupts, such as rising and falling edges of comparators, timers, are handled in an architecture-specific vector table. We have introduced our own version of the ISR into the NuttX middleware, by hijacking the vector table associated with each interrupt.

The following example indicates that the HRTIM D timer **Interrupt Request (IRQ)** vector, is associated with the physical interrupt `stm32_hrtim_td`, which occurs on specific events, such as the rising edge of track and hold signal.

```
1 VECTOR(stm32_hrtim_td, STM32_IRQ_HRTIMTD)        /* 71: HRTIM timer D interrupt */
```

We can therefore define the ISR as a special function, in the low level driver, such as **hrtim_isr_handler**, that gets **attached** to this vector, thus allowing the NuttX platform to execute it each time an interrupt happens in a **periodic** manner.

The following implemented algorithm represent the acquisition of multiple samples and saving them to the external SRAM, using the ISR function, programmed in the "RADAR" Low level driver, with a sweeping track and hold signal, starting from $0.9\mu s$, corresponding to the integer value $0.9\mu s/217ps = 4148$, and arriving at $2\mu s/217ps = 9218$.
For executing multiple coherent measurements, the sliding track and hold signal needs to stop at 9218 value and resets to 4148.

Algorithm 3: Stroboscopic acquisition algorithm

Result: Store sample[i - 1] in memory
Initialize
    Set $i \leftarrow 4148$;
    Set $nmeasure \leftarrow 0$;
    Set $address \leftarrow 0$;
    Set $sample \leftarrow 0$;
HRTIM Interrupt Service Routine
    if $i = 9218$ **then**
        nmeasures ← nmeasure + 1;
        $i \leftarrow 4148$;
    Interrupt Acknowledge (ACK);
    Set Track and hold signal at i;
    $i \leftarrow i + 1$;
    if $nmeasure < 14$ **then**
        Send 16 SPI clock cycles;
        Sample ← Acquired SPI value[i - 1] from ADC;
        Set memory CS low;
        Send WRITE instruction and the address **MSB** (16 bits, 16 SPI clock cycles);
        Send memory address **LSB**(16 bits, 16 SPI clock cycles);
        Send sample (16 bits, 16 SPI clock cycles);
        Set memory CS High;
    else
        break;
    $address \leftarrow address + 2$;

It is also important to notice that the memory address is incremented by 2 each cycle, since we are sending a 16-bit sample to the memory, which should fill 2 addressed memory spaces in the external SRAM, instead of an 8-bit sample which will fill 1 memory space.

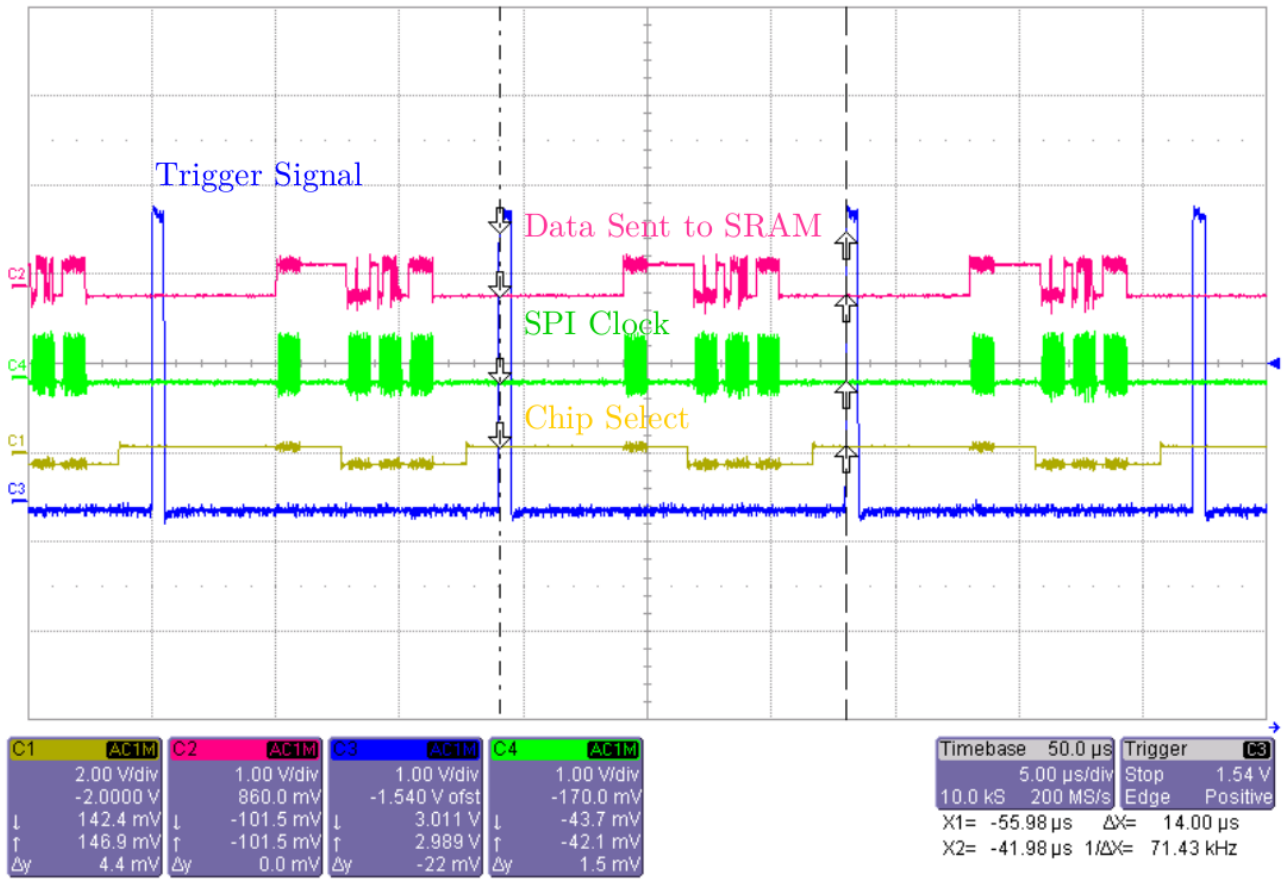

Figure 14: Measurement of the Chip select, SPI clock (SCK), data sent to RAM, triggered on the HRTIM trigger signal.

Figure 14 illustrates the deployment of algorithm 3, with a varying HRTIM track and hold signal (not visible here). The variation on 3 periods is in the order of $3 \times 217ps = 651ps$, not visible on the graph. Here we show that implementing the ISR is a viable option, safeguarding the transmission of the 64 SPI bits, which are seen as 4 packets of 16 bits clearly in the SPI clock signal.

Reading samples from the SRAM:
A viable method to store information sampled with the ADC using the stroboscopic method is done with algorithm 3. Reading from the SRAM uses a similar method, yet we are not restrained with HRTIM signal synchronization on reading.

Algorithm 4: Reading algorithm: **read**(nmeasures)

Result: Print measurements from memory
Initialize
  Set $address \leftarrow 0$; Set $sample \leftarrow 0$;
for $(address < 2 \times 5070 \times nmeasures)$ **do**
  Set memory CS low;
  Send READ instruction and the address **MSB** (16 bits, 16 SPI clock cycles);
  Send memory address **LSB**(16 bits, 16 SPI clock cycles);
  read sample (16 bits, 16 SPI clock cycles);
  Set memory CS High;
  $address \leftarrow address + 2$;
print(sample);

The high level application can therefore establish a call to a read method implemented in the lower level monolithic driver, using **read** system call. Algorithm 5 illustrate such principle: the memory space for the 23LC1024 SRAM is limited to 13 measurements (nmeasures), while each measurement contains 5070 points. The address space is twice this space, since each sample is stored in 2 addresses. The output is printed to the NuttX shell screen, which is available at the STM32F334 **USART 1**, as this serial bus is connected to the FTDI chip available on the receiver part of the RADAR prototype, which then transfers the data to a PC connected over USB. The means to connect to the chip is discussed in annex A.3.

Since we have configured a DDS to generate a synchronous signal with our HRTIM signals and using the developed algorithms, we can see the results of the stroboscopic sampling of a $10MHz$ sinusoïdal wave generated by the DDS:

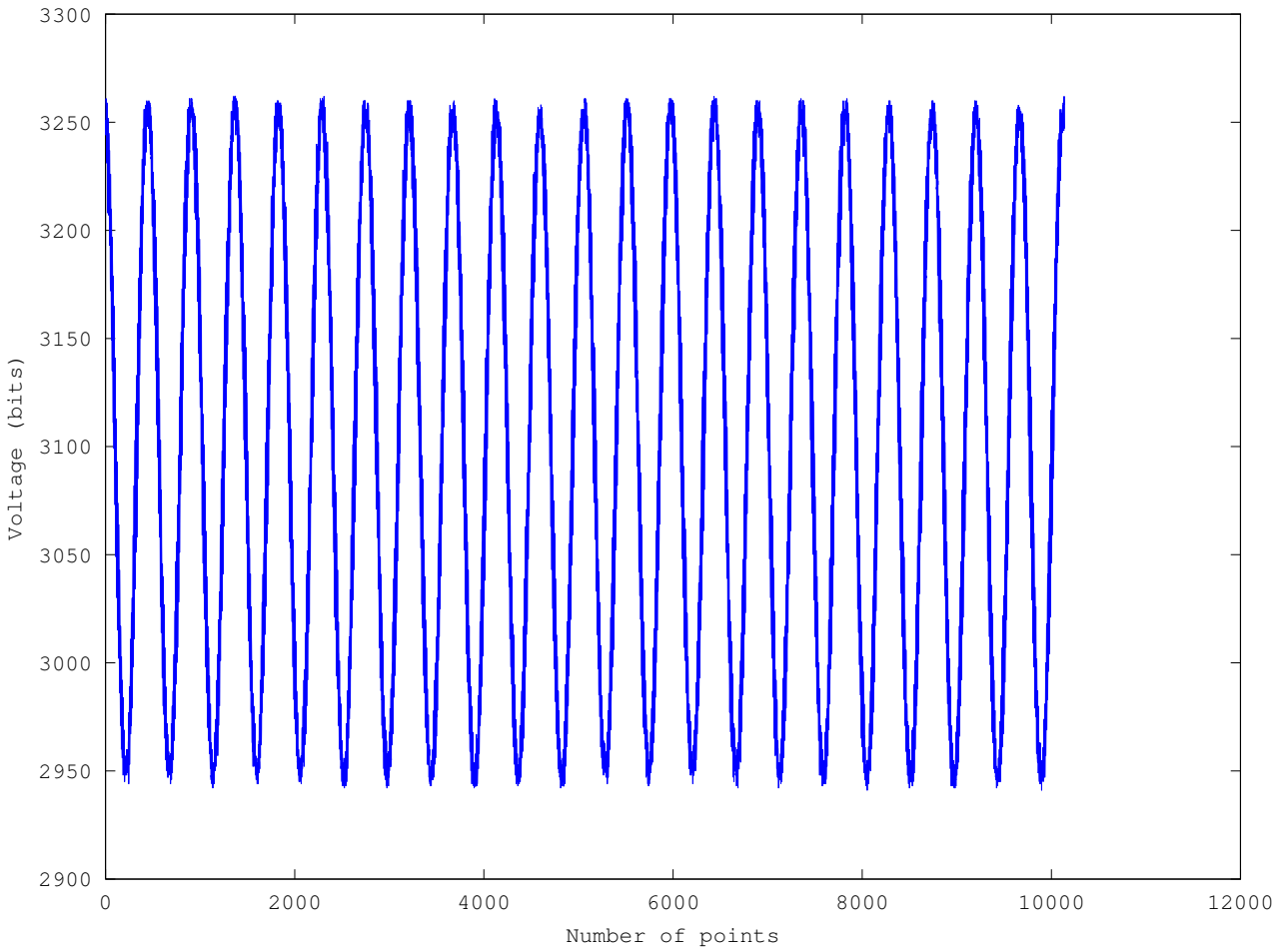

Figure 15: Sampled 10 MHz signal fed to the LTC1407 ADC by a synchronous DDS

Two measurements have been stored and printed out of the memory, corresponding to 10140 points. We can validate the correctness of the measurement, since each period of the wave corresponds to nearly 460 points, sampled with 217 ps precision, in other terms, the frequency of the wave is $1/(460 \times 217ps) = 10.018MHz$. The offset is due to imprecise estimation of the number of points.

Executing measurements beyond the memory limits:

It has been established that a single measurement, or 5070 sample points, takes $70.9ms$ in section 1.2.3. The SRAM memory limits the measurement count to 13, which is evidently insufficient for long term measurements lasting potentially for hours.

Since the data is sent back to a connected computer with the **read** system call, it is possible to devise a method to collect samples indefinitely. The following algorithm is implemented at the **application layer**, but with keeping the **atomicity** of the ISR routine.

Algorithm 5: Commuting algorithm between ADC sampling and reading from memory

Result: Print n measurements to associated computer
Parameters:
  nmeasures;
Initialize:
  Set $measures \leftarrow 0$;
for *(measures < nmeaures)* **do**
  execute_ISR(radar, 1); Executes the ISR for 1 measurement ;
  sleep(time); Sleep for the specified time ;
  read(radar, 1); Read from the memory for 1 measurement ;

"`execute_ISR`" is not a specific function but rather an added functionality to algorithm 3, allowing it to sample and store data when specified, or just acknowledge the interruption when idling. Moreover, the specified time is the controlling factor, as it should be longer than a single measurement ($70.9ms$).

2.5 Conclusion

The correct functionality of the embedded C software for the STM32F334 is an important point for the development of a RADAR prototype capable of probing SAW sensors. The design was made with versatility in mind, with the adoption of abstraction layers of the NuttX executive system to ease the task of further embedded software enhancement. After meeting excessive delays in propagating information through abstraction layers, the core part of the measurement algorithm was slowly concentrated to the low level interrupt service routine associated with the timer.

This development brings 2500 lines of code in the C programming language, with many enhanced features, not discussed in details in this document, such as:

- Sampling for limitless number of measurements, with a possibility of defining the time between each measurement.
- Sampling for sections in the received signal.
- Advanced sampling which is capable of reaching $4\mu s$ instead of previously discussed $2\mu s$, while maintaining the equivalent sampling rate of 4.6 GSample/s, thus recovering the 3rd sensor echo, which can be seen in figure 2.

The code has been extensively documented[15], dealing with the aforementioned problem of polling, the technical programming aspects of NuttX and the low level driver, the entire flowchart of the RADAR embedded software, as well as setting up all the necessary tools and environment to continue software development.

The performance of the embedded software is now verified, the development of the stroboscopic RADAR is handled in the next section.

3 Stroboscopic RADAR

The sampling part of the stroboscopic RADAR is introduced in section 2.1, yet we need to analyze the prototyped receiver architecture of figure 6.

3.1 Analysis of the receiver

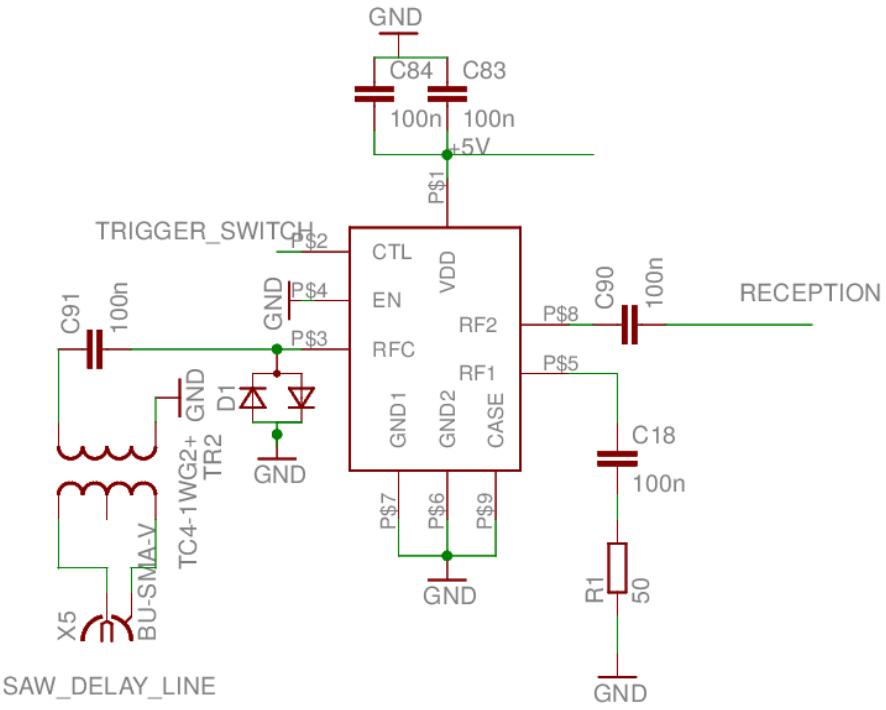

Figure 16: Schematic of the receiver section of the RADAR implemented by the CoSyma team: a transformer and radiofrequency switch as an entry point.

The first stage of the receiver section of the stroboscopic RADAR is the minicircuits TC4-1WG2+ radiofrequency transformer, which operates up to 800 MHz. It exhibits acceptable insertion loss ($0.88dB$) and input return loss ($15.77dB$)[16], yet this transformer has been found to be completely mismatched with the receiver dipole antennas, it was thus replaced with the MCL T1-1H [17], with similar performances around a narrowband of 8 to 300 MHz, with an insertion loss of $1dB$ around 100 MHz.

Throughout the circuit, decoupling capacitances are inserted between different stages of the receiver architecture, and to filter out high frequency noise mainly between the 5 volts power supply and the common ground. Protective diodes on the signal path are found to be superfluous and removed in an updated schematic. The received signal is switched on or off using a HMC349MS8G[18] radiofrequency switch, controlled by the HRTIM Trigger, and having a main objective of eliminating the **direct signal interference** of the emitted pulse by the pulsed RADAR stage, and the clutter. It operates from DC to 4 GHz, and exhibits high isolation of $70dB$ around 100 MHz from RFC (incoming signal) to RF2, the amplification stage. Judging from the IIP3 and the 1 dB compression point, the switch is sufficiently linear, which is a necessary condition for the measurement of SAW sensors, while not leaking excessive harmonics on the radiofrequency spectrum. The switch is controlled with CTL and EN inputs. Forcing EN to ground, if HRTIM is in LOW state, the signal passes, if not the signal goes to ground.

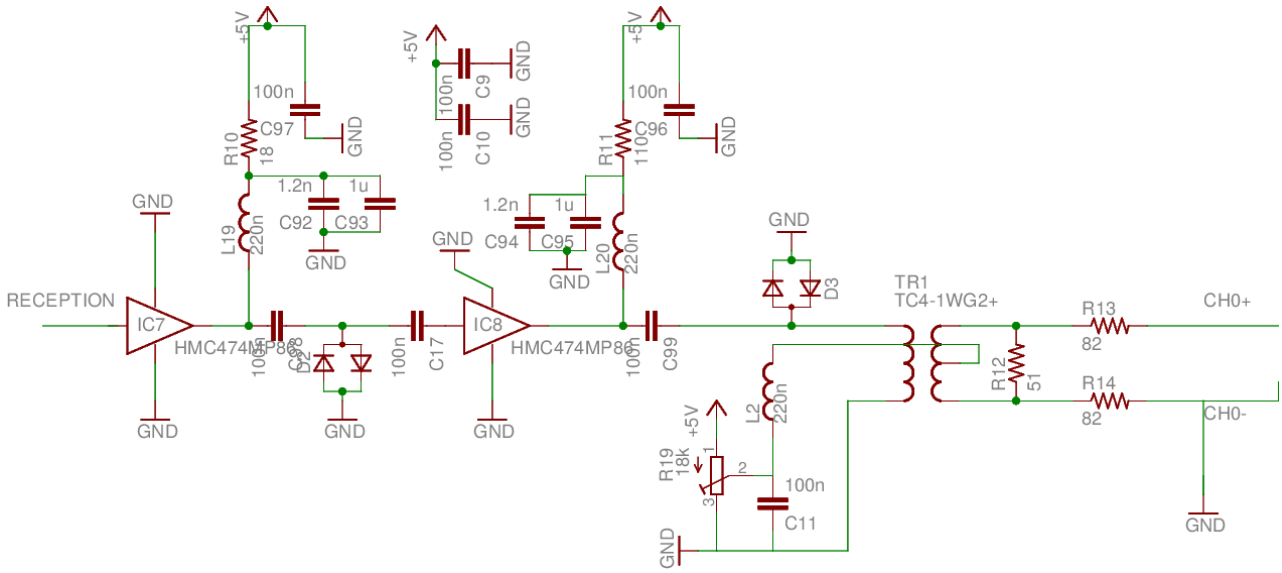

Figure 17: Amplification stage following the switch, with a variable potentiometer for a modifiable offset

Figure 17 illustrates the usage of 2 HMC474MP86 MMIC amplifiers[19] with their respective bias circuits. For a 100 MHz input signal, the individual gain of the HMC474MP86 is typically at $13dB$ with a 5V power supply, with a bias current of $25mA$ each. Cumulative gain of the 2 amplifiers saturates at $19dB$. We therefore replace both amplifiers with the similar yet more powerful HMC478MP86 MMIC amplifier[20] with an identical bias circuit and a bias current of $62mA$ for a $5V$ power supply. The typical gain of this amplifier reaches $22dB$.
The secondary RF transformer, with a potentiometer, acts as an offset circuit for the incoming RF signal, which is then sampled by the ADC (CH0+), in an unbalanced setting (CH0- is linked to ground). The updated schematic is presented in annex B.1.

3.2 Antenna Design

The receiver and emitter antennas need to be matched to their respective circuits, and wide band enough to enable a maximum electromagnetic energy coupling with the SAW sensor within its bandwidth of operation. An attractive antenna design is the usage of "Bow-tie" architecture, which are ultra-wideband antennas. Yet, this architecture suffers from poor transmission efficiency, due to the high impedance of the antenna at 95 MHz.
The simplest antenna design which is sufficiently wideband for our ground penetrating RADAR application is the thick dipole antenna. The dipole exhibits large bandwidth (larger than 20 MHz) at 95 MHz. First, we design the antenna dimensions in accordance to the probing environment. The RADAR prototype is developed on sand environment, with a measured relative permittivity $\epsilon_{sand} \simeq 2$[6]. We can therefore theoretically estimate the antenna dimensions, with the effective permittivity being defined as:

$$\epsilon_{eff} = \frac{\epsilon_{sand} + 1}{2} \simeq 1.5 \tag{2}$$

The wavelength λ_g of a wave propagating in the medium is defined as:

$$\lambda_g = \frac{c_0}{f_g \sqrt{\epsilon_{eff}}} \tag{3}$$

with c_0 the velocity of the electromagnetic in vacuum and f_g the operating frequency at 95 MHz, we find that $\lambda_g = 2.44m$. The length of a single antenna strand is equal to $\lambda_g/4 = 64cm$. In reality, the antenna lengths will be adjusted to obtain a maximum energy transfer through the observation of the returned signal's signal to noise ratio.

3.3 Pulsed Emitter hardware

The RADAR needs to emit a **wideband** electromagnetic wave: the SAW sensor needs to couple with an incoming electromagnetic with an operating frequency centered around 95 MHz, and a wide bandwidth of 20 MHz. The implemented solution is based on high speed amplifier techniques, using an avalanche transistor[21], and has been studied extensively in previous work[8][9]. Figure 18 shows the schematic of the pulsed circuit.

Operational Theory:

Store energy in a capacitor with a high voltage source. The transistor used in this configuration is an avalanche-based bipolar junction transistor, the voltage on the collector of this transistor will build up, until it reaches a tripping point which corresponds to the avalanche breakdown voltage. At this point, the transistor will start conducting current from the collector to the emitter, on which we connect a radiofrequency transformer. Furthermore, the transistor will react as a negative resistance, thus building up more current, which allows for the establishment of a high voltage signal on the emitter. This is done by absorbing the stored energy in the capacitor and sending it to the transformer, creating a pulse which lasts few nanoseconds, so spectrally a wide band signal.

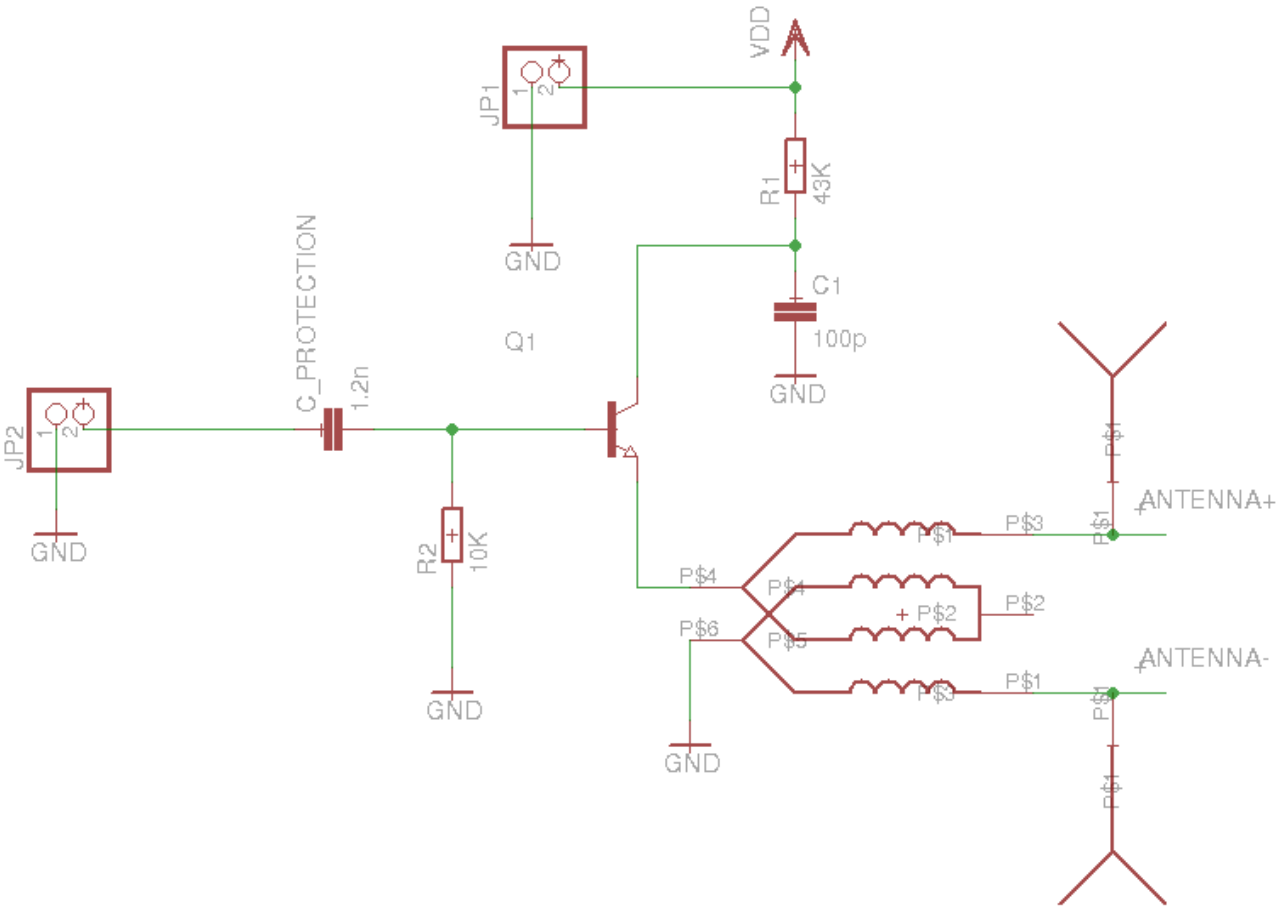

Figure 18: Pulsed emitter based on a quick current unloading scheme

The avalanche regime is triggered using a controlling signal, in our case the digital HRTIM trigger, biasing the transistor base, while its collector voltage is close but not larger then the avalanche breakdown voltage[8]. Based on the previous work[9], the values of $C = 100pF$, and $R = 43k\Omega$ in the RC collector network has been chosen to achieve maximum pulse voltage, which is $\propto \sqrt{C}$, and a resistance value small enough to allow the RC constant τ, describing our pulse repetition rate, to be within the range of our application (Admitting that the permanent voltage on the collector establishes around 3τ, the chosen value is close to $14\mu s$).

The FMMT413[22] transistor has been used, which has a particular breakdown voltage around 150 V. We use a high voltage supply, and the a high amplitude pulse (40 V) is seen when the threshold voltage exceeds 166 V for this particular configuration.

Ideal RADAR is meant to generate a single high voltage pulse on breakdown, yet in practical applications, the emitted pulse has a ringing waveform, clearly visible in figures 19 and 20. This is mainly due to the characteristics of the antenna transformer, which acts along with the antenna as the transistor load.

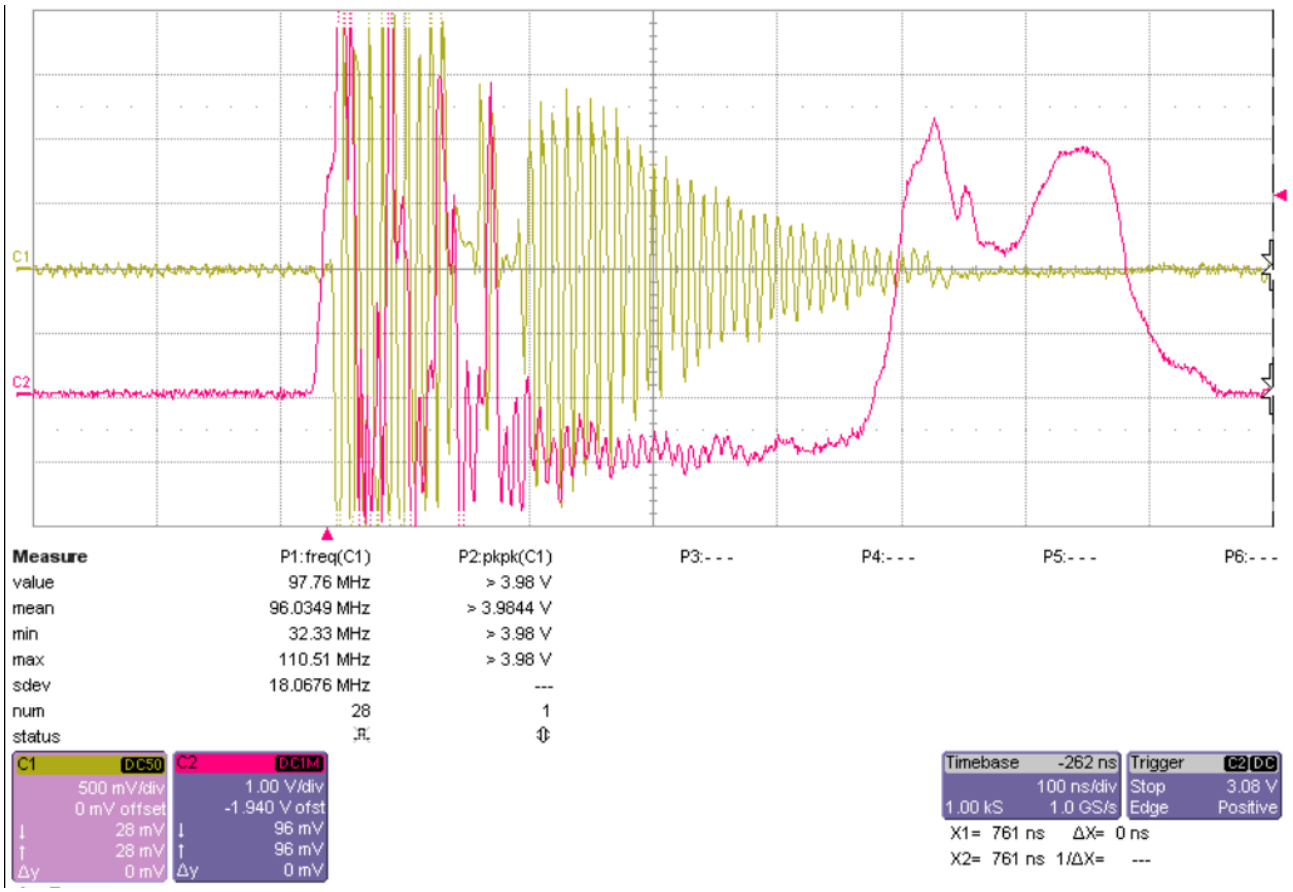

Figure 19: Pulsed emission (C1), trigger with the HRTIM trigger signal in red (C2). The excessive ringing is due to the transformer's high quality coefficient.

In figure 19, we can see the distorting effects of the emitted pulse on the HRTIM trigger, yet it still manages to trigger the avalanche mode. The excessive ringing seen is due to the high quality factor of the first implemented transformer[23], which causes the pulse to attenuate very slowly. The ringing phenomena of the pulse creates a more narrow band signal, yet excessive ringing doesn't couple well with the SAW sensor: the number of oscillations should be close the one of the interdigitated electrodes in the fabricated SAW sensor, to maximize the electro-mechanical wave conversion efficiency.

To reduce the number of oscillations, we have implemented a low quality factor transformer[24], while adapting the dipole antenna to the circuit correctly. The result is visible in figure 20.

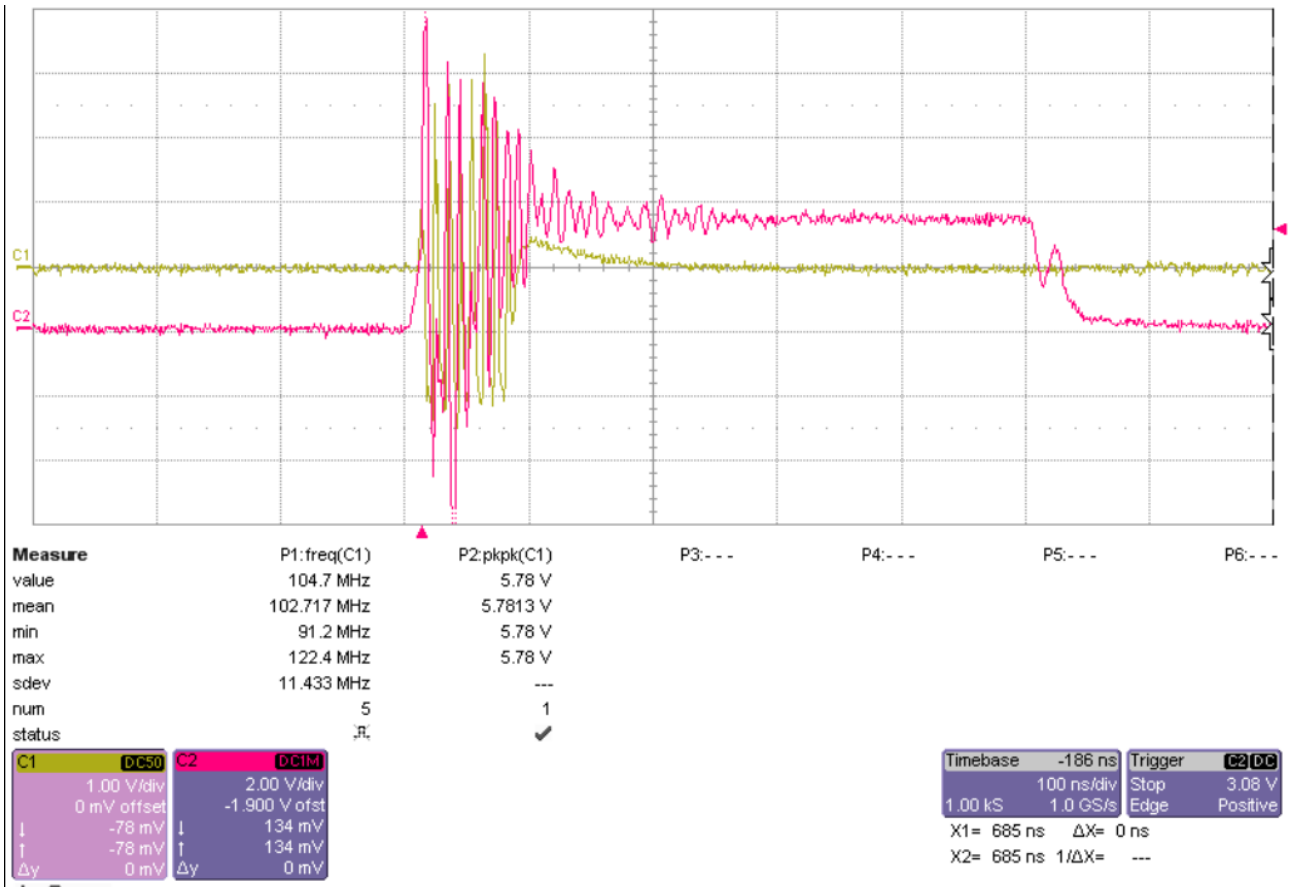

Figure 20: Pulsed Emitter based on a quick current unloading scheme

Figure 20 represents a highly attenuated pulse, with its visible effects on the HRTIM signal, yet this configuration is more stable. We can clearly notice that the number of oscillations has been reduced, which will allow a spectrally broader signal, and an efficient energetic coupling with the SAW sensor.

Sampling the received echoes:
The pulsed emission stage is now completed, with the ability to send high power wideband signals centered around 95 MHz, with a pulse repetition rate of $14\mu s$. We can therefore assemble the receiver dipole antenna with the dimensions calculated in section 3.2.

Even if the 2 receiver and emission antennas are designed to function with a large bandwidth for a given environment, establishing an experimental apparatus where the SAW sensor and its connected antenna, of similar dimensions, is set to be very close to the receiver/emission RADAR, we can manage to recover a good amplitude signal with the corresponding 2 echoes, delayed beyond clutter, shown in figure 21. The window corresponds to measurements between $1\mu s$ and $2\mu s$, with a visible $400ns$ time delay between the 2 echoes, using the embedded software developed in section 2.

Once multiple measurements have been conducted to verify the reproducibility of the results, the focus is shifted to assemble a battery-powered prototype for the ground penetrating RADAR, in the following section.

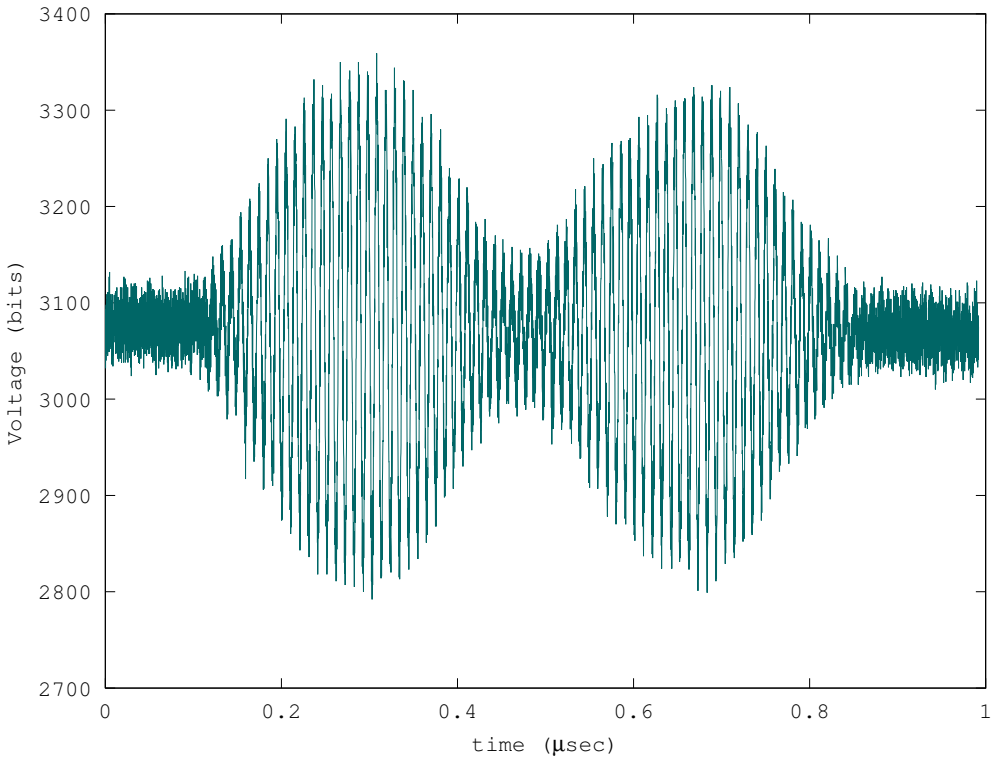

Figure 21: Sampled echoes of the returned SAW sensor signal with the stroboscopic method, and an equivalent sampling rate of 4.6 GSample/s.

3.4 Protective measures and RADAR prototype

The assembly of the ground penetrating RADAR needs to take into account the power supply voltage of the used high voltage source, or 12 V. Therefore, a 12 V battery has been chosen for the RADAR prototype.

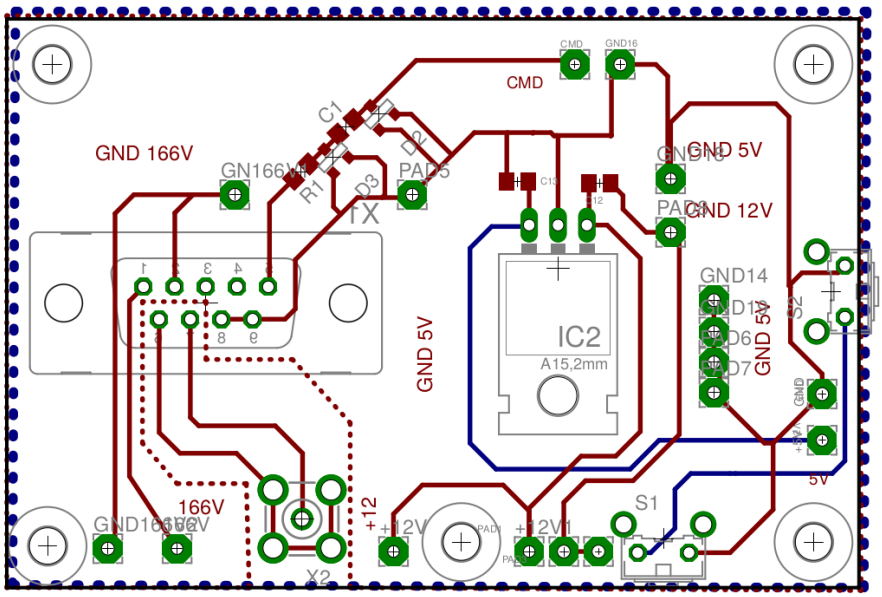

Figure 22: Regulator chip with protective Schottky clamping diodes for the HRTIM Trigger signal (CMD).

The RADAR assembly was designed to be modular: both receiver antenna and the pulsed emitter are embedded in a casing, while the receiving part of the RADAR, the high voltage regulator and the battery are in a separate casing. The two casings are connected through a D-subminiature E connector.
Figure 22 represents a design of an additional card used in the RADAR to connect the 2 casings, and as a power supply regulator for the RADAR receiver. It transfers the high voltage signal (166 Volts), its ground, the command or trigger signal (CMD) and the 5V-12V ground to the emitter chip through the DE connector.

All voltages are referenced to the same ground, with a 5V reference connection to the 12 V reference on this chip. The 12 V ground and the 166 V ground are connected on the high voltage regulator, but are kept separate on this chip, to eliminate a potential **ground loop**.

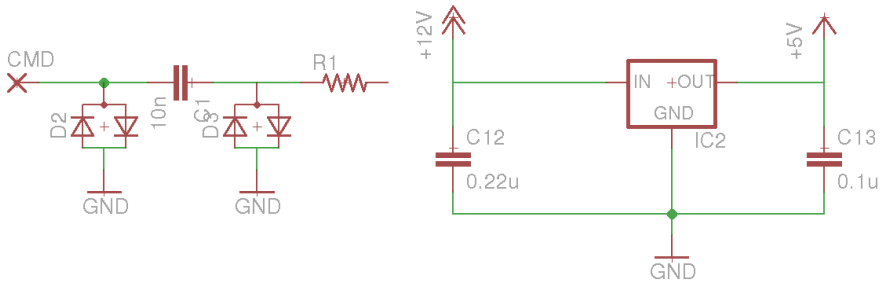

Figure 23: Double Schottky diodes with a protective capacitor and the 12 V to 5V stabilized regulator

Figure 23 shows the usage of clamping Schottky diodes (ZHCS1000[25]) to limit potential reverse current caused by the high voltage surges in the pulsed emitter, which can cause serious damage to the STM32F334 microcontroller HRTIM trigger GPIO. This chip therefore takes 12 V input and regulate it using the LM7805 regulator[26] with a fixed voltage output, to feed the RADAR receiver with 5V power supply. We can therefore establish the total RADAR circuit consumption:

Component	**Input Voltage**	**Consumption**
RADAR Receiver	5 V	150 mA
12 V to 166 V Voltage Regulator (idle)	12 V	40 mA
12 V to 166 V Voltage Regulator(Emitter ON)	12 V	120 mA
12 V to 5V Voltage Regulator	12 V	10 mA
Total circuit (idle)	12 V DC	200 mA
Total circuit (active)	12 V DC	280 mA

A single stroboscopic measurement lasts $70.9ms$, or in other terms the active state of the RADAR. Supposing that we take a measurement each second, the mean current consumption is $220mA$. The battery was chosen with a capacity of 1.2 A.h, therefore theoretically lasting for 5h and 30 mins.

Annex C.1 shows the main RADAR casing, with the high voltage regulator, the RADAR receiver and the 12 V battery.

4 Signal processing

4.1 Description of the underlying processes

The prototype can measure SAW sensors and return a specified n number of measurements. Recapping the measurement procedure:

- The RADAR sends a wideband electromagnetic pulse to the buried SAW sensor, while switching its radiofrequency switch to off state, rejecting the RADAR clutter.
- The SAW sensors send the delayed pulses, with 2 echoes, indicating 2 acoustic travel paths, d_1, d_2, with $d_1 > d_2$.
- The RADAR receiver amplifies the received signal, and the ADC samples a single point from it, at the specified time value of HRTIM track and hold.
- HRTIM track and hold increments by 217 ps.
- The process loops until all 5070 points have been collected, indicating the end of a single measurement.

The 2 lobes or echoes, have been implemented to allow a differential approach to timing measurement: this differential method will allow a suppression of the RADAR flight time between transceiver and sensor, and allow us to have a precise measure of the time delay by using a matched filter processing. The main impact of the environment on the SAW characteristics is through the acoustic velocity c propagating on the air-piezoelectric band, measuring the time difference between the 2 echoes refers to:

$$\tau = \tau_1 - \tau_2 = \frac{d_1}{c} - \frac{d_2}{c} = \frac{D}{c} \tag{4}$$

with D the distance difference between the 2 SAW sensor paths, τ_1 and τ_2 the respective flight time for the 2 paths.
Measuring time variance directly is quite challenging: another approach is to measure the **phase variation** due to change in wave velocity, since this variation can be expressed with c, based on the following derivation.

The signal phase is expressed as: (with f_c the pulse carrier operating frequency)

$$\varphi \ = 2\pi f_c \ \tau \tag{5}$$

Taking for example a temperature variation for the sensor, it induces a variation in c, therefore a variation in τ.

$$\frac{d\varphi}{dT_{emp}} = 2\pi f_c \frac{d\tau}{dT_{emp}} \tag{6}$$

Considering 4, we can replace τ by its terms,

$$\frac{d\varphi}{dT_{emp}} = 2\pi f_c \frac{d}{dT_{emp}} \left(\frac{D}{c}\right) \tag{7}$$

$$\frac{d\varphi}{dT_{emp}} = 2\pi f_c \ D \ \frac{d(c)}{dT_{emp}} \ \frac{1}{c^2} \tag{8}$$

For a unitary temperature variation, we have the relation for the phase variation:

$$d\varphi = 2\pi f_c \ D \ \frac{d(c)}{c} \ \frac{1}{c} \tag{9}$$

with D a fixed distance set during the design and fabrication processes, $d(c)/c$ the temperature sensitivity of the piezo-electric substrate, defined as 70 ppm/k for the lithium niobate with a YXl/128° cut[6], we can see that the **phase variation** is directly due to c variation.

A viable method for recovering the time delay through the phase is the cross-correlation algorithm, discussed in the following section.

4.2 The cross-correlation algorithm

Since the two echoes are generated by reflections of the same incoming pulse, a matched filter for detecting the echo pattern in a noisy measurement is the matched filter implemented as a cross-correlation aimed at detecting one echo shape as a copy of the other echo. Additionally, computing the cross correlation allows for a fine estimate of the time delay between the tow echoes.

The cross-correlation algorithm 6 has been implemented in GNU Octave[27], a versatile programming language for performing scientific computations.

Algorithm 6: Signal processing of the collected data

Result: Cross-correlation maxima position
Data: **n** measures
Initialize:
$size \leftarrow sizeof(measures)$;
for $\boldsymbol{i} \rightarrow \boldsymbol{n}$ **do**
$measures[all, \boldsymbol{i}] \leftarrow measures[all, \boldsymbol{i}] - \textbf{mean}(measures[all, \boldsymbol{i}])$;
$flobe[\boldsymbol{i}] \leftarrow measures[begin \rightarrow size/2, \boldsymbol{i}]$;
$slobe[\boldsymbol{i}] \leftarrow measures[size/2 \rightarrow end, \boldsymbol{i}]$;
$signal[\boldsymbol{i}] \leftarrow \textbf{xcorr}(flobe[\boldsymbol{i}], slobe[\boldsymbol{i}])$;
$\textbf{maxcorr}[\boldsymbol{i}] \leftarrow \textbf{max}(signal(all, \boldsymbol{i}))$;
return:
position(**polyfit**(**maxcorr**[$\boldsymbol{i}$], 2)) - **position**(**polyfit**(**maxcorr**[***1***], 2)) ;

First, we acquire n samples of the same size (5070) points for example, we remove the mean amplitude of the each of the signals, centering them around 0. Considering that the 2 lobes are centered in the sampled data, we get the size of a single measurement, and dissect the 2 lobes into 2 sections, a first section containing the first echo, the second section containing the secondary echo.

Then we cross-correlate the 2 sections using the built-in **xcorr** function by Octave, this cross-correlation will find the time delay between the 2 echoes, and the maximum difference as the index of the maximal value of the cross-correlation. We repeat this process until all measures have been evaluated.

Emphasis is focused on **the relative position** of the cross-correlation maximum, in respect to the first cross-correlation maximum. This indicates the variation of the phase in respect to the first position. To increase the precision of the phase variation measurement, we implement a parabolic fitting technique, which fits a quadratic expression around the local maxima.
The polynomial fitting is illustrated in figure 24. The complete cross-correlation algorithm written in Octave is in annex D.1.

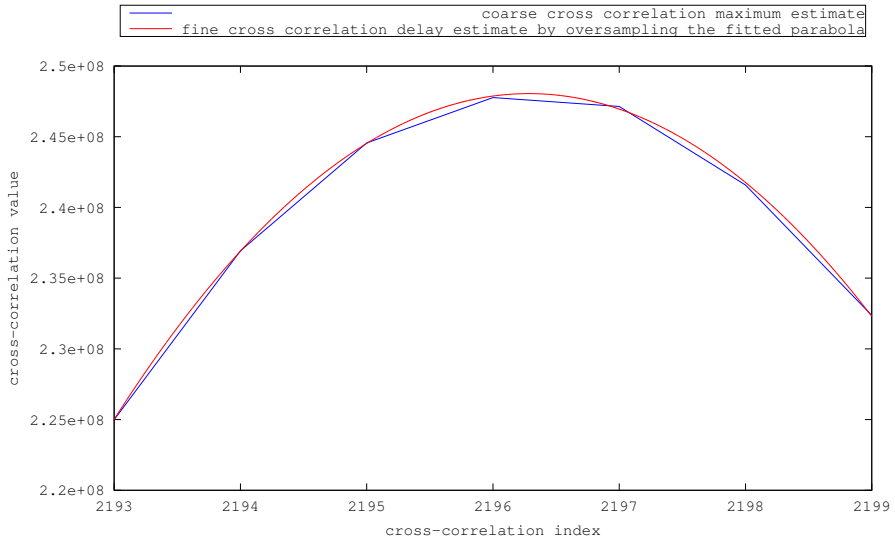

Figure 24: Fitting a quadratic expression around the local cross-correlation maxima increase its precision.

We execute a long term measurement aimed to show the validity of probing the wireless SAW sensor and measuring time delay. The measurement has been done while positioning the SAW sensor at 0.45 cm below the RADAR, and adapting the antennas for a test measurement in air. The length of each antenna strand was increased temporarily to $70cm$. Figure 25 shows that when applying a freezing spray on the sensor, the delay increases: this is due to a slowing down of the acoustic wave speed across the piezoelectric surface, taking longer periods to reach the mirrors.

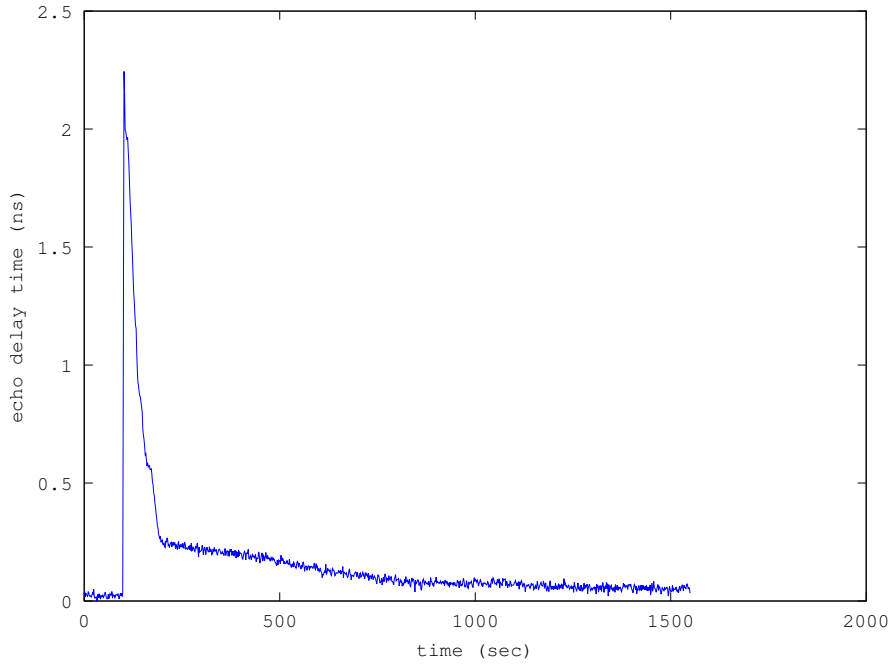

Figure 25: Long term measurement (26 minutes) of a SAW sensor, with the retrieval of a increase in time delay, due to the application of freezing spray. Standard deviation of the cross-correlation at the start is around 10.9 ps

The following section deals with the experimental measurements done with the antenna lengths adapted for a sand environment.

5 Experimental measurements

Time delay variation has been demonstrated in the previous section with the implementation of the cross-correlation algorithm.

The ground penetrating RADAR, however, needs testing in a real world scenario, such as detecting temperature variation in underground environments. An experimental 3.3 m-wide, 5.4-m long and 1.2 m deep sandbox has been installed for the previous characterization of commercial GPR equipment[6]. Figure 26 shows the sandbox, with embedded Poly(Vinyl Chloride) (PVC) pipes inserted in varying depth along the sandbox, allowing a fine control over the depth of the SAW sensor and its antenna[6].

The GPR is also shown connected through USB to a laptop, probing a SAW sensor directly beneath it. A stable laboratory power supply is used to generate high current across a power resistor, connected directly to the SAW sensor and a PT100 temperature sensor used to detect temperature variation between the SAW and the power resistor. Details of this experiment is discussed in section 5.2.

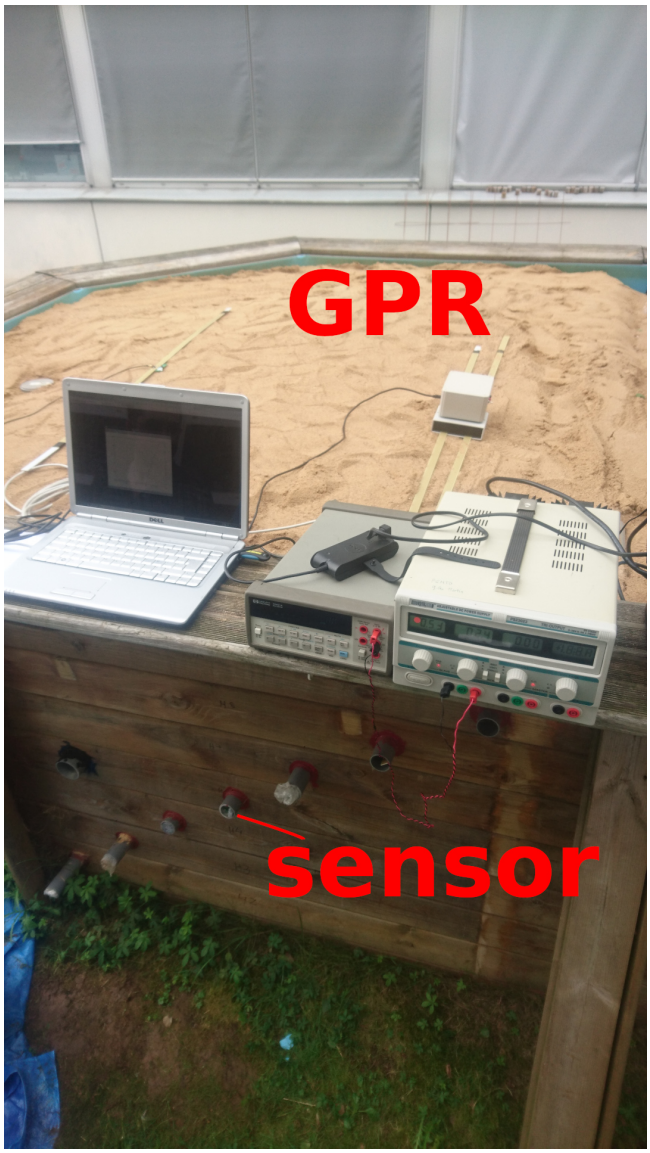

Figure 26: Measurement apparatus for buried wireless SAW sensors in sand.

The objectives of the measurements are:

- To verify the ability of the ground penetrating RADAR of reaching underground cooperative SAW sensors, and receiving sensor echoes with varying SAW sensor depths. The procedure of this experiment are discussed in section 5.1.
- Ensure the sustainability of the measurements on long periods of time, and the electronics consumption.
- Detect a physical quantity change in sub-terrain environments, in this case, temperature variation with the power resistor setup, discussed in section 5.2.

First, we start with a series of measurements on buried SAW sensors to confirm its ability of reaching targets beyond 50 cm in sand.

5.1 Buried sensors interrogation

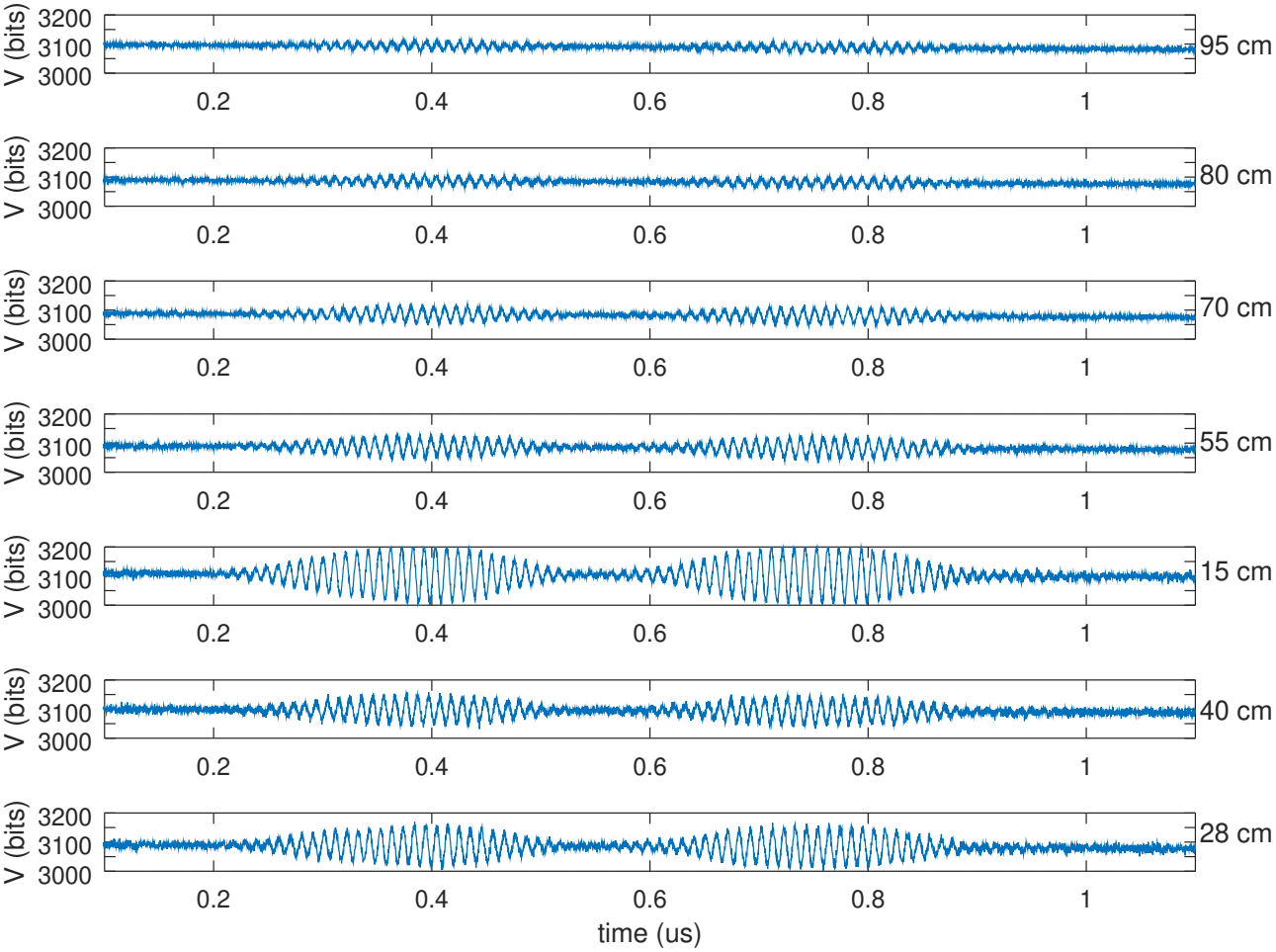

Figure 27: Measurements of SAW sensors on different depths

In this experiment, we probe a single underground sensor inserted in one of the PVC pipes at different depths in sand, with averaging 10 **successive** measurements each. Let's recall that each measurement lasts for $70.9ms$: 10 measurements have a duration of $0.7s$, yet the temperature gradient in sub-terrain environment is stable enough for this time duration. The signal retains its coherence even if we average the 10 measurements.

Figure 27 shows the received echoes for SAW sensors at varying depths: we have an effective decrease in Signal to Noise Ratio (SNR) when reaching lower and lower depths, upon a near complete attenuation of the returned signal at depths exceeding 1 m in sand. This is due to a fading pulse and returned signals when penetrating the sand material, even if homogeneous fine sand is used[6], due to the returned power decay as the fourth power of the distance in the monostatic RADAR equation.

Considering signal SNR, the best candidate for performing long time sensor testing is effectively with the maximum energy coupling with the sensor at 15 cm of depth. Yet, we need to demonstrate the ability to detect temperature (or other physical quantities) changes at depths exceeding 0.5 meters.

5.2 Measurement of temperature variations

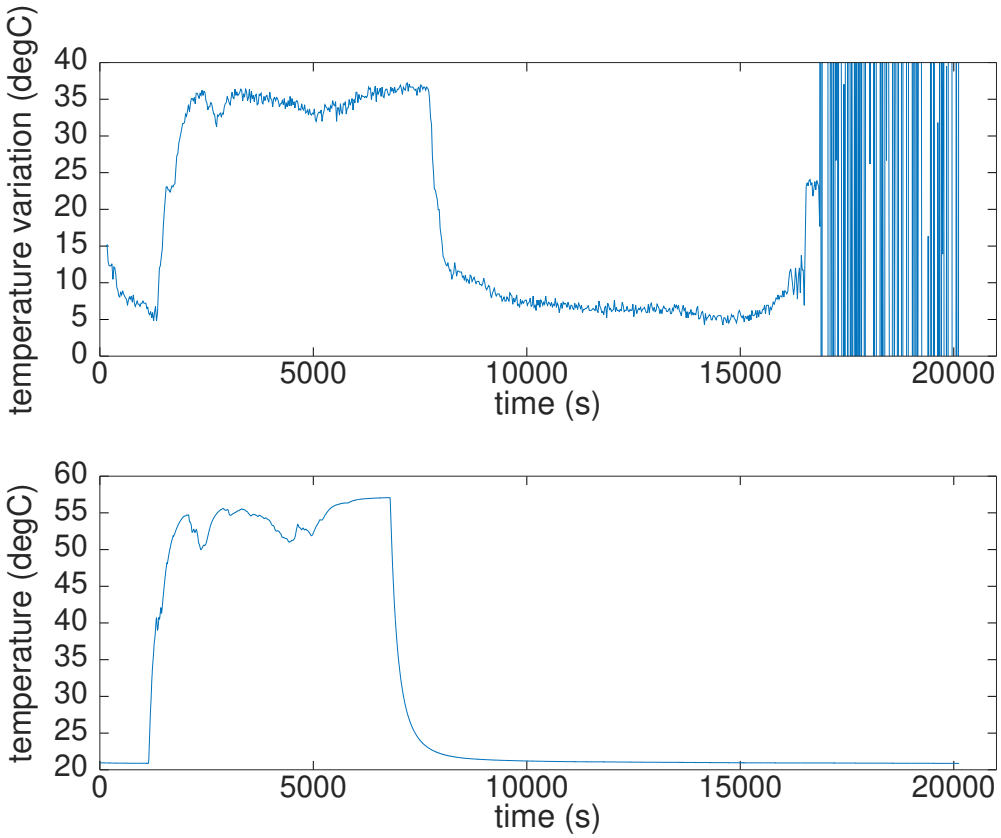

Figure 28: Long term measurement at 55 cm depth, of temperature variations introduced by a power resistor: the first graph shows temperature variation detected by the GPR up to 15000 seconds, and the second graph shows the current temperature of the SAW sensor with respect to time

We setup the GPR and the sensor at the depth of 55 cm, and launch 2 measurements at the same time: one through the usage of the GPR cross-correlation algorithm, modified to plot temperature variations (from echo time delays), and a second measurement used to record current temperature of the SAW sensor with PT100 temperature sensor. At the start of the procedure, the power resistor attached to the SAW sensor is unconnected to a power supply for 20 mins, then when connected, a high current flows within the power resistor, allowing it to reach temperatures up to 55°C as assessed by the reference PT100 temperature probe, then disconnected after 2 hours. We can clearly see the temperature variations detected by the SAW sensor, corresponding to the variation of the power resistor's temperature, therefore validating our ability to probe the wireless SAW sensor at 55 cm while detecting temperature variation.
While the sensor measurement can be validated, the random temperature variations in the transient response are due to power supply faults. The chaotic behavior of the RADAR beyond 15000 s (4 hours) is due to battery capacity depletion, which first introduce cross-correlation distortion, then failure of the avalanche transistor to trigger.
The initial drift, prior to the power resistor heating, is associated with a temperature drift of the reference quartz oscillator whose thermal management was not carefully addressed during the design. Nevertheless, this initial drift is a hundredfold lower than the one observed in the commercial Malå ProEx instrument.

Conclusion

Subsurface interrogation of wireless SAW transducers acting as cooperative GPR targets is an exciting approach to remote sensing. However commercial GPRs suffer from an extensive drift in their time base, rendering any attempt to probe underground SAW sensors useless. The main objective of this study was to establish a working GPR prototype capable of addressing the timebase derivation while providing accurate long term measurement of SAW sensors in underground environments.

A systematic approach to the developed pulsed RADAR has been deployed, with an emphasis on the general context and operating principles, such as the stroboscopic method, yielding design constraints.

The embedded RADAR system development is continued with the integration of a versatile real-time executive system, constituting a vital part of this project and the core of our contribution. A detailed analysis of the ground penetrating RADAR electronic architecture and implementation has been discussed and enhanced, with the design of the radiofrequency link and wideband antennas compatible with the spectral wideband operation of the SAW sensors. The cross-correlation algorithm has been developed to make use of the generated GPR prototype data, and the extraction of useful information from the signal, in our case a viable temperature variation measurement, extended over several hours, while demonstrating sufficient timebase stability.

The requirements of the study have all been met, while demonstrating the robustness of the embedded software development approach, additional work should be followed on the implementation of the cross-correlation algorithm directly on the RADAR receiver. This constitutes one of many perspectives for the elaboration of the project, like extending the current stroboscopic receiver to include the regulating chip and the high voltage source.

This study is rich in terms of versatility: it combines approaches from advanced embedded C software programming, real-time operating system programming, antenna and analog design, wideband pulsed architectures, advanced signal processing, RADAR monitoring and subsurface sensing, MEMS based acoustic resonators.

The research project has greatly extended my competence as an engineering student to quickly assess and correctly dissect complex problems, and to use my acquired knowledge and build upon it to solve engineering problems requiring a diversified collection of skills.

References

[1] FEMTO-ST, "The Time and Frequency Department" *https://www.femto-st.fr/en/Research-departments/TIME-and-FREQUENCY/Presentation* (June 2018)

[2] Mordor Intelligence *https://www.mordorintelligence.com/industry-reports/global-chemical-sensors-market-industry.* (February 2018)

[3] "UNDERGROUND: Systems for monitoring sub-surface pollution risks in oil industry infrastructures using passive sensors", *AAPG ANR 2018, Comité d'évaluation scientifique (CES 42) - PRCE.* (February 2018)

[4] J. M. Friedt, "Passive Cooperative Targets for Subsurface Physical and Chemical Measurements: A Systems Perspective", in *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 6, pp. 821-825, June 2017. (February 2018)

[5] J. M. Friedt et al., "Subsurface wireless chemical sensing strategy compatible with Ground Penetrating RADAR", *2017 9th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, Edinburgh, 2017, pp. 1-6. (February 2018)

[6] J.-M Friedt, G. Martin, G. Goavec-Mérou, D. Rabus, S. Alzuaga, L. Arapan, M. Sagnard, E. Carry. "Acoustic Transducers as Passive Cooperative Targets for Wireless Sensing of the Sub-Surface World: Challenges of Probing with Ground Penetrating RADAR", *MDPI Sensors 2018, 18(1), 246.* (February 2018)

[7] Minary, F et al. Note: A dual-chip stroboscopic pulsed RADAR for probing passive sensors. *The Review of scientific instruments 87 9 (2016): 096104.* (February 2018)

[8] D. Rabus, F. Minary, G. Martin, J.-M Friedt. "A high-stability dual-chip GPR for cooperative target probing", *GPR 2018.* (February 2018)

[9] A. Gushing, "Rapport de Stage : Réalisation d'un radar à pénétration de sol", *Bachelor thesis,University of Franche-Comté.* (February 2018)

[10] Linear Technology, "LTC1407-1/LTC1407A-1 Serial 12-Bit/14-Bit, 3 Msps Simultaneous Sampling ADCs with Shutdown", *LTC1407A-1 datasheet.* (February 2018)

[11] Microchip, "23A1024/23LC1024 1Mbit SPI Serial SRAM with SDI and SQI Interface", *23LC1024 datasheet.* (February 2018)

[12] "NuttX Real-Time Operating System", `http://www.nuttx.org/` . (February 2018)

[13] STMicroelectronics, "STM32F334xx advanced ARM based 32-bit MCUs", *RM0364 Reference manual.* (February 2018)

[14] STMicroelectronics, "AN4539 Application note", *HRTIM cookbook*, p. 10-12. (February 2018)

[15] M. Hage Hassan, "NuttX Stroboscopic Code Integration, Report and Code Reference".

[16] Minicircuits, "Surface Mount RF Transformer TC4-1WG2+", *TC4-1WG2+ datasheet.* (April 2018)

[17] Minicircuits, "Surface Mount RF Transformer T1-1H-KK81+", *T1-1H-KK81+ datasheet.* (April 2018)

[18] Hittie Microwave Corporation, "HMC349MS8G/349MS8GE High Isolation SPDT Non-Reflective Switch DC - 4 GHz" *HMC349MS8G datasheet, v02.0607.* (April 2018)

[19] Analog Devices, "HMC474MP86/474MP86E SiGe HBT Gain Block MMIC Amplifier, DC - 6 GHz", *HMC474MP86 datasheet, v01.0906.* (April 2018)

[20] Analog Devices, "HMC478MP86/478MP86E SiGe HBT Gain Block MMIC Amplifier, DC - 4 GHz", *HMC478MP86 datasheet, v03.0810.* (April 2018)

[21] Jim Williams, "High Speed Amplifier Techniques, A Designer Companion for Wideband Circuitry", *Linear Technology, Application Note 47, August 1991.* (April 2018)

[22] Diodes Incorporated "FMMT413 NPN Avalanche Transistor in SOT23", *FMMT413 avalanche transistor datasheet.* (April 2018)

[23] Minicircuits, "Surface Mount RF Transformer TC4-62H+", *TC4-62H+ datasheet.* (April 2018)

[24] Minicircuits, "Surface Mount RF Transformer TC4-25X+", *TC4-25X+ datasheet.* (April 2018)

[25] Diodes Incorporated, "ZHCS1000 Surface Mount Schottky Barrier Diode", *ZHCS1000 datasheet.* (May 2018)

[26] Texas Instruments, "LM340, LM340A and LM7805 Family Wide V_{IN} 1.5 A Fixed-Voltage Regulators", *LM7805 datasheet.* (May 2018)

[27] John W. Eaton, David Bateman, Søren Hauberg, Rik Wehbring (2014). "GNU Octave version 3.8.1 manual: a high-level interactive language for numerical computations."" *CreateSpace Independent Publishing Platform. ISBN 1441413006,* URL `http://www.gnu.org/software/octave/doc/interpreter/` (May 2018)

[28] R. S. Weis and T. K. Gaylord, "Lithium niobate: Summary of physical properties and crystal structure", *"Applied Physics A, Aug 01 1985, vol. 37. n.4* pages 191-203. (June 2018)

[29] Analog Devices, "A Technical Tutorial on Digital Signal Synthesis", `https://www.ieee.li/pdf/essay/dds.pdf` (June 2018)

[30] McKay, K. G., "Avalanche Breakdown in Silicon", *American Physical Society, Phys. Rev. vol 94. issue 4.* pages 877-884. (February 2018)

[31] Future Technology Devices International Ltd, "FT232R USB UART IC Datasheet version 2.13", *FTDI RT232 datasheet*

Annex A

A.1 Time delay accuracy calculation

Considering a SAW sensor with an operating frequency of 95 MHz, and a time delay between the echoes of 400 ns, as seen in the time domain analysis in figure 2.
The normalized sensor sensitivity can be expressed as:

$$S = \frac{\Delta_f}{f}\Big|_T = \frac{\Delta_v}{v}\Big|_T = \frac{\Delta_t}{t}\Big|_T = \frac{\Delta_\varphi}{\varphi}\Big|_T \tag{10}$$

We are mainly interested in the phase term $S = \frac{\Delta_\varphi}{\varphi}$. Considering the phase $\varphi = 2\pi ft = 76\pi$ for the corresponding frequency and time delay, phase variation Δ_φ can be therefore calculated

$$\Delta_\varphi = \varphi \times S \tag{11}$$

Since the current sensor sensitivity considered is of 70 ppm/K for the lithium niobate with a YXl/128° cut[6], $\Delta_\varphi = 76\pi \times 70 \times 10^{-6} = 0.0167 rad/k$.

Reconsidering equation 10, we can now calculate the time delay accuracy:

$$\Delta_t = t \times \frac{\Delta_\varphi}{\varphi} = 28ps \tag{12}$$

A.2 HRTIM configuration example

This code represents the implementation of the configuration of the HRTIMD CH1 (Trigger) in NuttX, in the low level driver layer.

This functionality is called by passing the "`STM32_HRTIM_MOD_ENABLE_HRTIMD_CH1`" command from a higher level application, through the **system call** input/output control (ioctl).

```
static int stm32_radar_ioctl(FAR struct file * filep, int cmd, unsigned long arg){
  switch(cmd) {
      case STM32_HRTIM_MOD_ENABLE_HRTIMD_CH1: {
        // Disable the timers for configuration
        HRTIM_OUTPUTS_ENABLE(hrtim, HRTIM_OUT_TIMD_CH1, false);

        // Sets the period of the HRTIMD GOLBAL (CH1 AND CH2)
        HRTIM_PER_SET(hrtim, HRTIM_TIMER_TIMD, 64516); // 0.5 us duty cycle;

        // Sets the Reset of the CH1 on CMP1
        HRTIM_CMP_SET(hrtim, HRTIM_TIMER_TIMD, HRTIM_CMP1, 2304); // Period: 14 us

        // Activate the HRTIMD CH1, depends on the arguments
        HRTIM_OUTPUTS_ENABLE(hrtim, HRTIM_OUT_TIMD_CH1, (bool)arg); // HRTIMD CH1
    Configuration
        break;
      }
}
```

A.3 Serial-USB connection

The FTDI chip on the receiver section of the RADAR allows a USB-serial connection to the USART1 of the STM32F334 microcontroller, which is vital for communication and

programming of the firmware into the FLASH region of the processor. To connect to the RADAR and the NuttX shell available, under a Linux operating system, we issue the following command:

```
screen /dev/ttyUSB0 460800 8N1
```

Where screen is a versatile toolkit used to connect to the serial port located at `/dev/ttyUSB0` under Linux. In the configuration of NuttX, we can program the maximum serial bus speed to be 460800 Bauds, thus allowing data transfer in the shortest amount of time to a computer, and with No parity 8 bits transmission packet, with 1 bit being the stop bit for the RS232 protocol.

Annex B

B.1 Updated Receiver Architecture

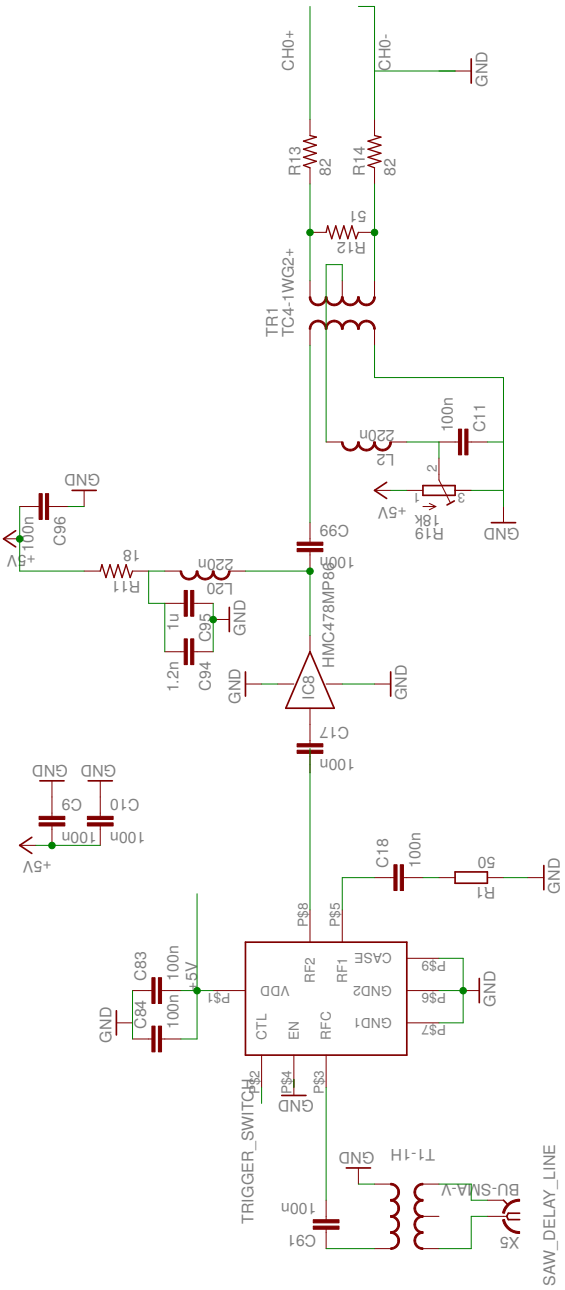

Figure 29: Updated receiver architecture

B.2 Circuit boards

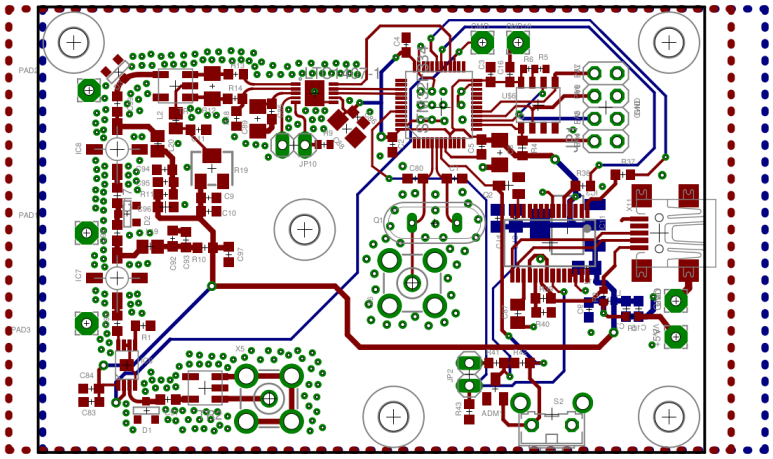

Figure 30: RADAR receiver board

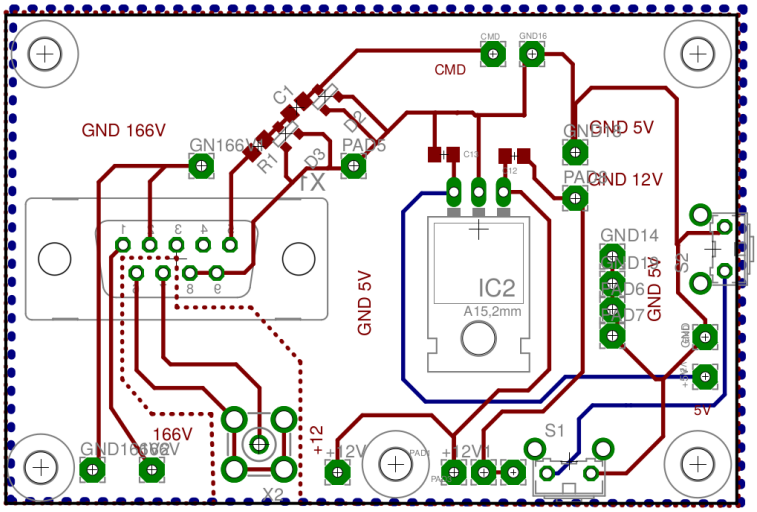

Figure 31: Regulator chip for the RADAR voltage regulator

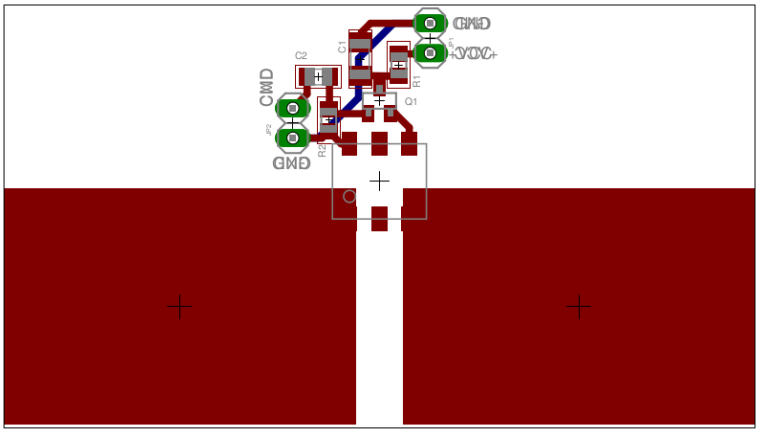

Figure 32: Emission stage using the pulsed avalanche transistor topology

Annex C

C.1 RADAR Casing

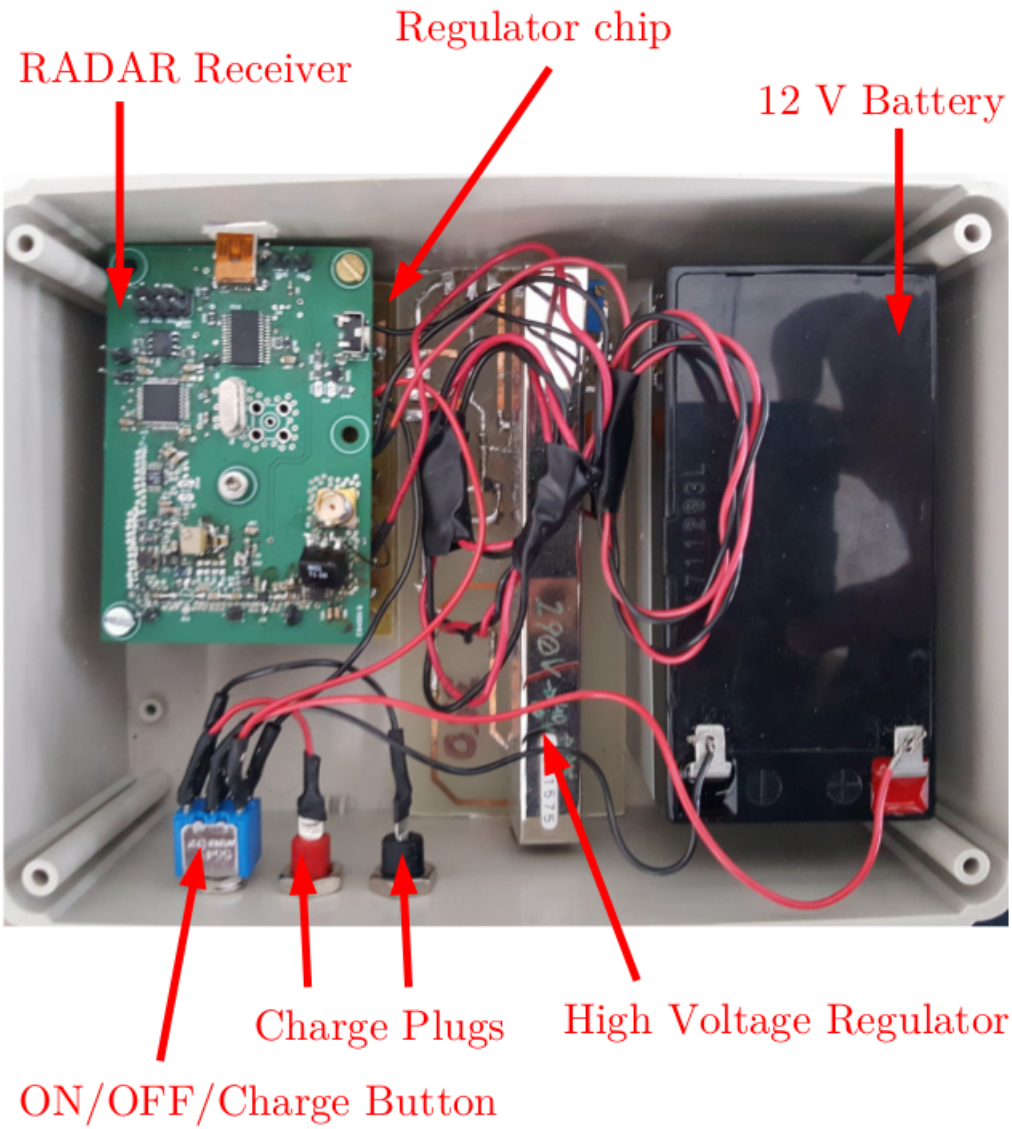

Figure 33: Radar casing showing the RADAR receiver, the regulator chip beneath it, the high voltage regulator and the 12 V battery

Figure 33 represents the RADAR prototype interior for the reception part. The DE connector is directly beneath the 2 electronic chips.

Annex D

D.1 Cross-correlation code

This section represents the cross-correlation code implemented in GNU Octave to calculate the time delay between the 2 echoes of the SAW sensor, and plotting the result directly.

```
clear all;
pkg load signal;
graphics_toolkit ("gnuplot");
file = fopen("log.advancedmodes.4");
measures=fscanf(file, "%x", [5070, 1000]);
fclose(file);

NMeasures = 1000;

printf("Reading Data...\n");

for data_index=1:NMeasures
  fs(:, data_index) = measures(1000:2950, data_index);
  ss(:, data_index) = measures(2950:4900, data_index);
end

printf("Calculating xcorrs...\n");

for index=1:NMeasures
  sig(:, index) = xcorr(fs(:, index) - mean(fs(:, index)), ss(:, index) - mean(ss(:, index)));
  plateau_x = find(sig(:,1) == max(sig(:,1)));
  colmax(:, index) = find(sig(plateau_x-10:plateau_x+10,index) == max(sig(plateau_x-10:
    plateau_x+10,index)));
end

colmax = colmax - colmax(:, 1);

printf("Calculating the variations..\n");

for index=1:NMeasures
  colmax(:, index) = colmax(:, index) + plateau_x;

  x(:, index) = colmax(:, index)-3:1:colmax(:, index)+3;
  finerx(:, index) = colmax(:, index)-3:0.001:colmax(:, index)+3;

  croppedsig(:, index)=sig(min(x(:,index)): max(x(:, index)), index);

  func(:, index) = polyfit(x(:, index), croppedsig(:, index), 2);
  approx(:, index) = polyval(func(:, index), finerx(:, index));
  [maxval current_column]=max(approx(:,index));
  finalvalue(index)=finerx(current_column,index);
end

sample_period=217E-3; % in nanoseconds
absolute_val = finalvalue - min(finalvalue);

time_between_measures = 1.55;
totaltime=1:time_between_measures:time_between_measures*NMeasures;
plot(totaltime, absolute_val.*sample_period, 'b');

ylabel("echo delay time (ns)");
xlabel('time (sec)');
```

Résumé

L'augmentation de la demande mondiale pour des solutions de mesures à distance des environnements et milieux industriels a menée au projet UNDERGROUND de l'ANR, avec une mission fondamentale de la création des méthodes pour détecter les variantes de quantités physiques comme la température et les fuites chimiques, ce qui a mené au développement des capteurs passifs interrogeables à distance. Le deuxième objectif de ce projet est de fiabiliser la mesure à distance de ces capteurs.

L'utilisation des RADARs à pénétration de sol commerciaux, a démontré leur capacité à interroger les capteurs sans fils passifs, mais leur dérive de la base de temps est assez importante, ce qui rend l'extraction de toute donnée par rapport à une variation de paramtètres physiques inutile. L'objectif de cette présente étude est la fabrication d'un RADAR à pénétration de sol capable d'interroger les capteurs, présentant une stabililité en terme de base de temps, ainsi qu'une portabilité et fidélité à long terme.

Le RADAR à pénétration de sol était prototypé, en partant d'une approche système qui consiste à delimiter le cahier de charge requis, l'approche vers une électronique embarquée avec le développement élaboré du système, mise en place effective du bilan de liason et le traitement de signal des informations acquises. Ce RADAR était testé en scénario réel sur des experiences pour détecter une variation en température en milieu sous-terrain, et correspond bien au cahier de charge en terme de la distance d'intérrogation, durée de vie et la précision de mesure.

Abstract

The global increase in demands for monitoring solutions in the context of remote and industrial underground sensing has led to the deployment of the ANR project UNDERGROUND, with a mission objective to develop of state-of-the-art sensors and methods to interrogate such sensors in the required environments. A key objective within UNDERGROUND is to establish a reliable technique for probing fabricated sensors to measure quantitative physical parameters like temperature, stress, or chemical leaks.

While commercial ground penetrating RADAR have been found to be capable of exciting the designed Surface Acoustic Wave sensors, their time base drift makes useful information extraction practically useless. The objective of this study was to establish a valid probing mechanism, a ground penetrating RADAR, capable of probing the SAW sensors with high degree of fidelity, portability, and high sustainability.

A ground penetrating RADAR has been prototyped, starting from a systematic approach of probing techniques, to embedded electronics and system development, to designing and enhancing the radiofrequency link budget, to the signal processing of incoming analog information. The RADAR has been tested on a real-world scenario of experimental temperature measurements and fits the required specifications of the project in terms of reliability, probing range, battery life, and measurement precision.