Politecnico di Torino

College of Computer Engineering, Cinema and Mechatronics

Master's Degree in Mechatronic Engineering

Master's Degree Thesis

Evaluation of ADC Modules for Real-time Control of Electric Motors in Automotive Systems

Supervisor:
Ph.D. Sarah AZIMI

Candidate:
Giuseppe COSTA

Co-supervisor:
Prof. Luca STERPONE

Academic year 2022-2023

Abstract

Today, analog-to-digital converters are vital components of contemporary technology. They play a significant role, particularly in the automotive sector, as hundreds of sensors constantly require converting analog signals into digital ones for further processing. To monitor electrical quantities and manage electric motors efficiently, ADCs are integrated into the system-on-a-chip to offer a compact and efficient solution. The control system must provide a quick and accurate response for electric motors while managing multiple signals simultaneously. With the aim of converting several signals at the same time from a DC/DC converter and maximizing performance in order to achieve a fast conversion, this research thesis first focused on analyzing existing microcontroller technologies. In order to determine the best solution, a comparison was made between the analog-to-digital conversion modules of the AURIX™ TC399 and other alternative board solutions. The results showed that the former was superior. The study then proceeded with a detailed evaluation of the various ADC modules existing on the board provided by Infineon, critically comparing EVADC (Enhanced-Versatile-Analog-to-Digital Converter) and EDSADC (Enhanced-Delta-Sigma-Analog-to-Digital Converter). After considering the balance between resolution and conversion time, the EVADC was chosen because it is quicker than the other type of converter. This fast conversion capability is crucial to ensure real-time control of currents and voltages, which in turn regulate the PWM signals used for precise control of the DC/DC converter's switches. A further contribution to this thesis was the successful synchronization of the GTM (Generic Timer Module) with the EVADC module. This enables the use of PWM signals generated by the TOM (Timer Output Module) as triggers for EVADC conversion processes. Finally, regarding the proposed method, an application was initially developed for the conversion of a single analog signal. Subsequently, two different approaches were introduced to calculate the actual

conversion times of the EVADC module. After selecting the optimal configuration, a final application was implemented that allows the simultaneous conversion of several signals.

Acknowledgements

I wish to express my profound thanks to my thesis supervisors, Sarah Azimi and Luca Sterpone, for their invaluable guidance and support throughout this thesis.
The contribution of my supervisor Sarah Azimi has been crucial for me. I am grateful for the valuable guidance and continuous support provided by her. I appreciated her patience in answering my questions and providing wise and enlightening recommendations.
I would also like to express my gratitude to Luca Sterpone, whose technical expertise and analytical approach enhanced my work and allowed me to perceive the subject in a new light.
Special thanks go to my family and friends for their love, support and endless patience. Without them, this accomplishment would have been much more challenging to attain.

Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Analog-to-Digital Converter
ARU	Advanced Routing Unit
ATOM	ARU-connected Timer Output Module
CALSTC	Calibration Sample Time Control
CCU	Capture Compare Units
CIC	Cyclic Integrating Comb
CMU	Clock Management Unit
CPU	Central Processing Unit
DAC	Digital-to-Analog Converter
DMA	Direct Memory Access
DSP	Digital Signal Processor
EDSADC	Enhanced Delta-Sigma Analog-to-Digital Converter
EOC	End of Conversion
EVADC	Enhanced Versatile Analog-to-Digital Converter
FIFO	First-In, First-Out
FIR	Finite Impulse Response

FTM	FlexTimer Module
GTM	Generic Timer Module
LSB	Least Significant Bit
MCU	Microcontroller Unit
MSB	Most Significant Bit
NR	Noise Reduction
NRS	Noise Reduction Step
PWM	Pulse-Width Modulation
RISC	Reduced Instruction Set Computer
SAR ADC	Successive Approximation Register Analog-to-Digital Converter
SOC	Start of Conversion
SOMB	Signal Output Mode Buffered Compare
SOMC	Signal Output Mode Compare
SOMI	Signal Output Mode Immediate
SOMP	Signal Output Mode PWM
SOMS	Signal Output Mode Serial
SPB	System Peripheral Bus
SRI	Shared Resource Interconnect
STC	Additional Sample Time
STM	System Timer Module
TIM	Timer Input Module
TRGMUX	Trigger Multiplexer

Chapter 1

Introduction

Nowadays, we live in an increasingly interconnected and digitized world, where analog-to-digital converters play a key role in converting analog quantities into digital ones that are later be used for further processing.
Advancement in technology, particularly in the automotive industry, has increased the need for systems that can provide precise, reliable and fast conversions to guarantee safety, efficiency and optimal performance.
Analog-to-digital converters (ADCs) play a crucial role, because they are connected to hundreds of sensors, such as pressure sensors, temperature sensors, engine speed sensors and so on, which require the constant conversion of analog signals into digital ones.
It is through this conversion process that the data becomes accessible for various applications, including digital control strategies, power management, current and voltage control, fuel consumption optimization, and much more.
In the automotive industry, ADCs are integrated into systems-on-a-chip (SoCs) achieving a high level of optimization, and control and this mark a significant step forward in the realization of compact and efficient solutions for monitoring electrical parameters.
Indeed, electric motors need careful management, and SoC-integrated ADCs are becoming essential instruments for this task.
The ADC in a SoC is not an isolated component, but part of a large group of processors, sensors and control systems. Synchronization between these parts on a single chip is consequently very important.

However, the integration of ADCs into system-on-a-chip presents different challenges. Control systems need to provide fast and precise responses for electric motors, and one important factor to consider is the trade-off between conversion speed and resolution. In essence, a power conversion system includes a battery, a DC/DC converter and an inverter. The latter two necessitate PWM signals to govern the current flow via the switches. The PWM signals are produced by precise regulation of voltage and current. To carry out this control, it is necessary to convert the signals that are picked up by the sensors in order to extrapolate useful data, and all these operations must be done in a very short time.

This is where this thesis project comes in, where the aim is to convert several signals simultaneously and in the shortest possible time. In order to attain the goal in the most efficient manner possible, this thesis first analyses the various microcontroller solutions available.

A comparison of the different options can be found in Chapter 2, with a special focus on the various conversion modules. Another aspect tackled in this study concerns the synchronization between the conversion module and the timer module, responsible for generating various PWM signals. The AURIX™ TC399 controller was selected, mainly for its conversion capabilities. After conducting a thorough examination of the different conversion and timer modules on the Infineon board in Chapter 3, a decision is ultimately reached regarding the modules to be used. Chapter 4 outlines this decision in detail, highlighting the preference for the EVADC module over the EDSADC module and the TOM module over the ATOM for generating PWM waves. Using the AURIX Development Studio environment, it was possible to first develop an application that allows the conversion of a single signal. After calculating the actual conversion times, using two different methods, it was possible to create a final application that allows several signals to be converted simultaneously with minimal conversion times.

Chapter 2

State of the art

2.1 ADC into System-on-Chip (SoC)

The analog-to-digital converter is crucial in any application that utilizes sensors since it acts as an interface between the analogue and digital domains. Over the years, demand has increased to integrate these devices into integrated circuits due to the significant advantages in terms of cost, power, and size. Nowadays, complete and very complex systems are integrated on one single die.

Briefly, a system-on-chip is an integrated circuit that includes most of a computer's components and other electronic systems. These components include central processing units (CPU), memory interfaces, input/output devices and interfaces. An SoC integrates a microcontroller, microprocessor or perhaps multiple processor cores.

The integration of an ADC in a SoC offers a significant solution. As it shares space with other functional blocks, the need for external blocks is reduced and the circuit layout is simplified. Moreover, the integration makes it possible to optimise power consumption by switching the ADC on or off based on the task at hand. Additionally, the distance between the ADC and digital circuits is minimised, thereby reducing noise and electromagnetic interference. This is of great importance in applications where size is restricted, such as in the automotive industry.

For performance, efficiency, and safety, synchronization between the different components and modules within a microcontroller is essential in modern automotive systems. In particular, in the case examined in this thesis work, i.e. the acquisition of signals from a DC/DC converter in such a way as to ensure real-time control, the

synchronization between the PWM signals and the ADCs is crucial.
PWM is a digital signal that regulates the output voltage from a DC source while limiting the power dissipated by the electrical system. Essentially, it is a kind of digital modulation that delivers a fluctuating voltage based on the proportion between the period of the positive pulse and the total time (duty cycle).
This is made easier when the ADC and PWM signal generation modules are on the same board. There are many microcontroller suppliers worldwide, and three of the top five are European companies: NXP, STMicroelectronics, and Infineon Technologies.

Looking at NXP and focusing on the modules relevant to this research, it appears that the widely used microcontroller family is the S32K1xx[1]. It incorporates an Arm™ Cortex-M4F/M0+ core, a 32-bit CPU, supports frequencies up to 112 MHz, and includes a fast external oscillator range of 4 to 40 MHz. For the analog-to-digital converters, there are two 12-bit ADC modules with up to 32 channels available per module. To calculate the total conversion time the following formula it is used:

$$\begin{aligned} ADC\ Total\ conversion\ time = {} & Sample\ Phase\ Time \\ & + \ Hold\ Phase\ Time \\ & + \ Compare\ Phase\ Time \\ & + \ Single\ or\ First\ continuous\ time\ adder \end{aligned} \tag{2.1}$$

Where: Sample Phase Time = Sample time + 1, Hold Phase = 1 ADC Cycle, Compare Phase Time = 28 ADC Cycles (12-bit Mode), Single or First continuous time adder (5 ADC cycles + 5 bus clock cycles).
In the datasheet of this microcontroller[2] it can also be found that $f_{conv\_max} = 36.25 Ksps$.
Regarding the generation of PWM signals, there are up to eight independent 16-bit FlexTimer (FTM) modules, providing up to 64 standard channels. Then, these signals can trigger the ADCs via the TRGMUX module.

[1][1]S32K1xx Series Reference Manual. URL: `https://usermanual.wiki/Document/S32K1xx20MCU20Family20Reference20ManualREV208.1780670565/html`.

[2][2]NXP Semiconductors Data Sheet. URL: `https://www.nxp.com/docs/en/data-sheet/S32K-DS.pdf`.

Turning to the Microcontrollers manufactured by STMicroelectronics, one of the most widely used families is the STM32F405xx and STM32F4057xx, which features an Arm® 32-bit Cortex®-M4 CPU[3]. It supports a frequency of 168 MHz with an external crystal oscillator ranging from 4 to 26 MHz. As for the analog-to-digital converters, there are three 12-bit converters. Each converter has 16 channels for the external inputs and two channels for internal communication. The total conversion time can be calculated using the following formula:

$$\begin{aligned} ADC\ Total\ conversion\ time = Sampling\ Time \\ + \ 12\ Cycle \end{aligned} \tag{2.2}$$

Where: $T_{conv\_min} = 500ns$, with $f_{ADC} = 30MHz$[4].
Moreover, about the generation of PWM signals, they possess up to 17 timers (TIM) in which 12 of them are 16-bit and the remaining are 32-bit. Specifically, the TIM1 and TIM8 modules have the capability to generate PWM waves which can be synchronised with the ADCs.

Finally, let's discuss Infineon's Microcontrollers, with the AURIX TC3xx family being a particularly popular choice. They are equipped with 6 TriCore™ CPUs running at 300 MHz and a 20 MHz crystal oscillator[5]. Analog-digital converters are available in two classes: EVADCs and EDSADCs, which will be thoroughly reviewed in later sections. Conversion times can be calculated as follows:

$$\begin{aligned} ADC\ Total\ conversion\ time = Sampling\ Time \\ + \ ADC\ conversion\ clock\ frequency \\ + \ Noise\ Reduction\ Time \\ + \ Post\ Calibration\ Time \end{aligned} \tag{2.3}$$

[3][3]STM32F405xx STM32F407xx Data Sheet. URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f405rg.html.

[4][4]RM0090 Reference manual. URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f405-415/documentation.html.

[5][5]Infineon TC39x Datasheet. URL: https://www.infineon.com/dgdl/Infineon-TC39x-DataSheet-v01_02-EN.pdf?fileId=5546d462712ef9b7017140bc3416145f.

In the User Manual [6] it can also be found that $t_{conv\_min} = 375ns$. For the generation of the PWM signals, there are various modules inside the Generic Timer Module (GTM), such as TOM (6 modules, 16 channels each, 16 bit counter value) and ATOM (12 modules, 8 channels each, 24 bit counter value). In this thesis work, the Microcontroller supplied by Infineon was chosen, further details will be provided in the following chapters.

2.2 ADC-PWM synchronization

The motor control unit is an embedded real-time application in the automotive sector, and it necessitates the integration of timing across multiple modules for the precise functioning of numerous sub-applications. The aim of this section is to demonstrate how previous studies tackled the issue of converting multiple analog signals by synchronizing the conversion with PWM signals. These processes utilizes boards from NXP and ST. In the previous studies[7] it is used the microcontroller S32K144. Since the current sampling point plays a crucial role in motor control applications, it is necessary to synchronize ADC and PWM and the associated modules. Two ADC groups (ADC0 and ADC1) are used to simultaneously convert two analog signals. The analog frequency of the converters is 40MHz. Once this is done, the FTM module is configured to generate six PWM signals with a 2 μs dead-time between them. The PWM waves are set with a duty cycle of 50% and a frequency of 10 kHz. As the clock of FTM module runs at 80 MHz, the period corresponding to 10 kHz is 8000 ticks, derived from the formula: 80MHz/10kHz. Then, thanks to the TRGMUX module, the FTM-ADC modules are interconnected. In fact, the PWM signals act as triggers for the ADC converters. Finally, equation 2.1 is used to calculate the total conversion time, which on this board is 1.3 μs. In the figure 2.1[8] it is possible to see the flowchart of the work done by previous studies.

[6][6]User Manual Part2 v02_00. URL: `https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_Part2-UserManual-v02_00-EN.pdf?fileId=5546d462712ef9b701717d35f8541d94`.

[7][7]Likitha, T., Venkatanarasimharao Medam, and G. Ranjani. "ADC-PWM Synchronization for Motor Control Unit Development." 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT). IEEE, 2022.

[8][7]Fig 3, Pag 4 from: Likitha, T., Venkatanarasimharao Medam, and G. Ranjani. "ADC-PWM Synchronization for Motor Control Unit Development." 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT). IEEE, 2022.

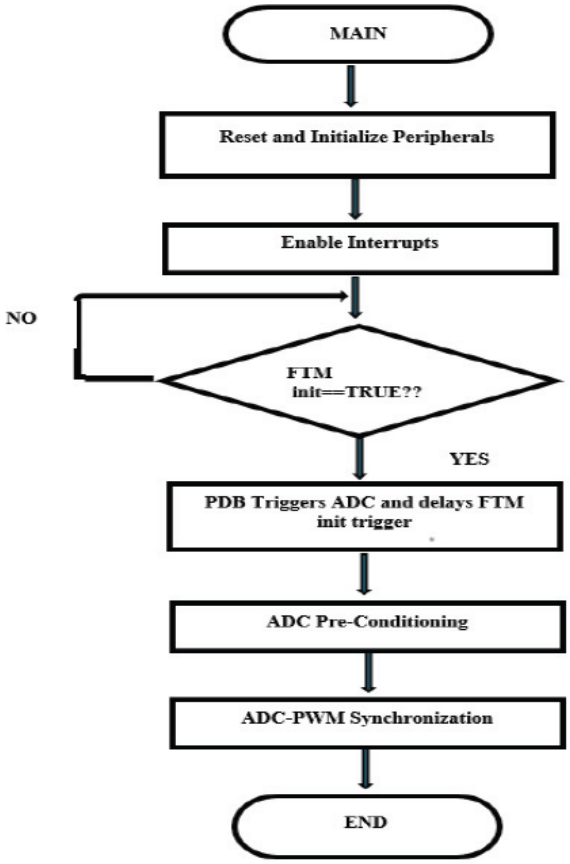

Figure 2.1: Flowchart for ADC-PWM Synchronization.

Another task involving multiple signal conversion and synchronization with PWM waves has been successfully completed using the STM32F407ZGT6 microcontroller[9]. What makes this work unique is that two different controllers are used to acquire signals, store results, and control the converters via PWM waves. The system requires nine ADC channels for sampling and conversion; hence, three ADC groups with three channels each are utilized. TIM 4 timer is used to generate the PWM signal. When a falling edge occurs, each ADC group begins to convert channel 1 simultaneously. After completion, they generate a DMA request, and the value is stored in SRAM. The conversion process then proceeds to channel 2 and 3. The system waits for the next trigger after all 9 channels have been converted. The work just described

[9][8]Wang, Bowen, et al. "Research on two-dimensional electric field measurement system for electrical tree detection in insulating materials." 2022 IEEE International Conference on High Voltage Engineering and Applications (ICHVE). IEEE, 2022.

concludes that up to 72 channels can be converted simultaneously. The conversion time is not specified but can be referred to from the board's datasheet, as mentioned in the previous paragraph. In the figure 2.2[10] there is the flowchart of the work done by previous studies.

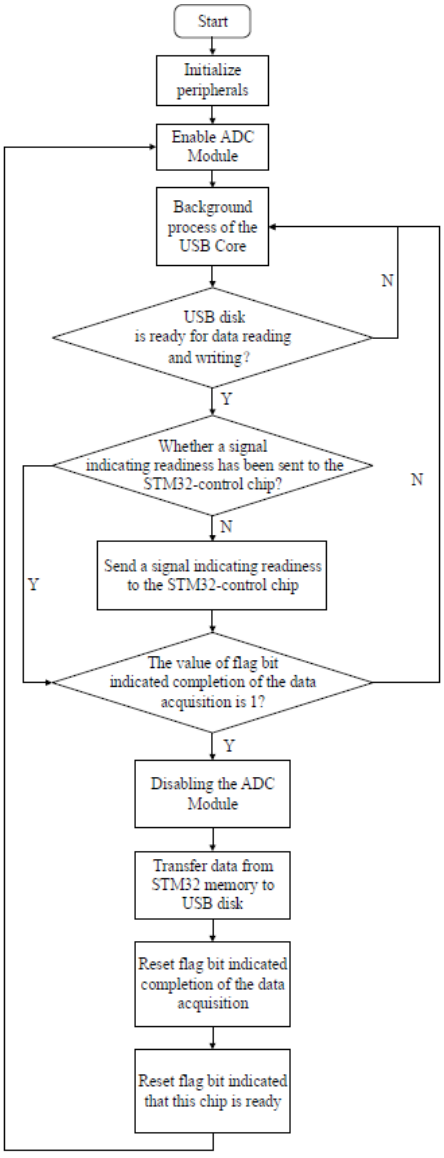

Figure 2.2: The program flow chart of STM32-acquisition and storage chip.

[10][8]Fig 7, Pag 3 from: Wang, Bowen, et al. "Research on two-dimensional electric field measurement system for electrical tree detection in insulating materials." 2022 IEEE International Conference on High Voltage Engineering and Applications (ICHVE). IEEE, 2022.

Chapter 3

Background

3.1 Analog-to-digital conversion

Analog-to-digital conversion is an electronic process in which an analog signal is changed into a digital one.
A quantity is analog if it can take any value within a certain range, whereas a quantity is digital if it can only take specific values.

Figure 3.1: Analog-to-digital conversion.

Quantities such as voltage, current, and time are analog. Thanks to modern technology, it has become easier to process digital signals.
Therefore, a conversion process of the input signals is required. Converting from a continuous quantity to a discrete one in time and amplitude involves a process of sampling and quantization, respectively.

3.1.1 Sampling

Considering a signal $x(t)$ with an amplitude spectrum $X(f)$, the conversion process must generate a sequence of values such that the signal trend is well represented.

Performing a discretization in time, the trend of $x(t)$ is represented by a series of samples $x[n] = x(nT_s)$ which are taken by sampling $x(t)$ with constant step T_s.
The amplitude of each sample is equal to the amplitude of the input signal at that precise instant of sampling.
We can obtain the sampled signal by multiplying the signal $x(t)$ by a Dirac series δ.

$$x_s(t) = x(t) \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) X(f) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} \delta(f - \frac{n}{T_s}) = \frac{1}{T_s} \sum_{n=-\infty}^{+\infty} X(f - \frac{n}{T_s}) \quad (3.1)$$

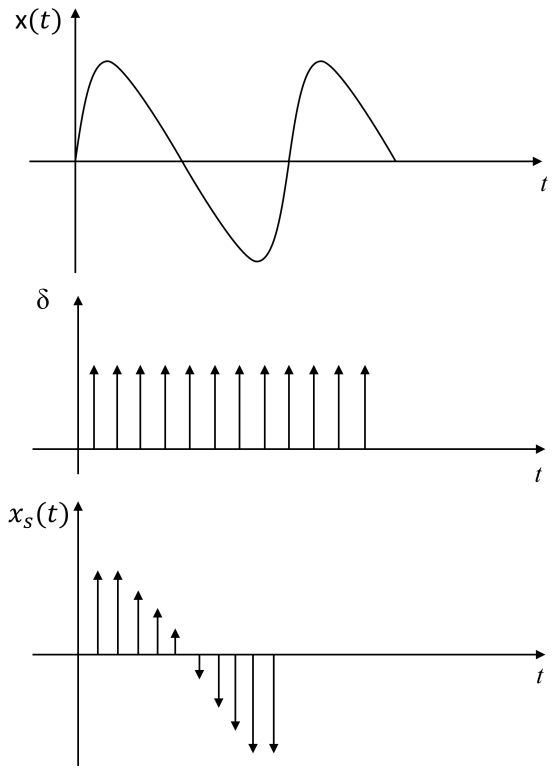

Figure 3.2: Sampling process.

To reconstruct the original signal, the signal spectrum must be considered. In particular, the baseband part of the spectrum must be isolated by means of a low-pass filter.
Isolation is only possible if the secondary spectra do not overlap, and this is achieved by choosing the right sampling frequency $f_s = \frac{1}{T_s}$.

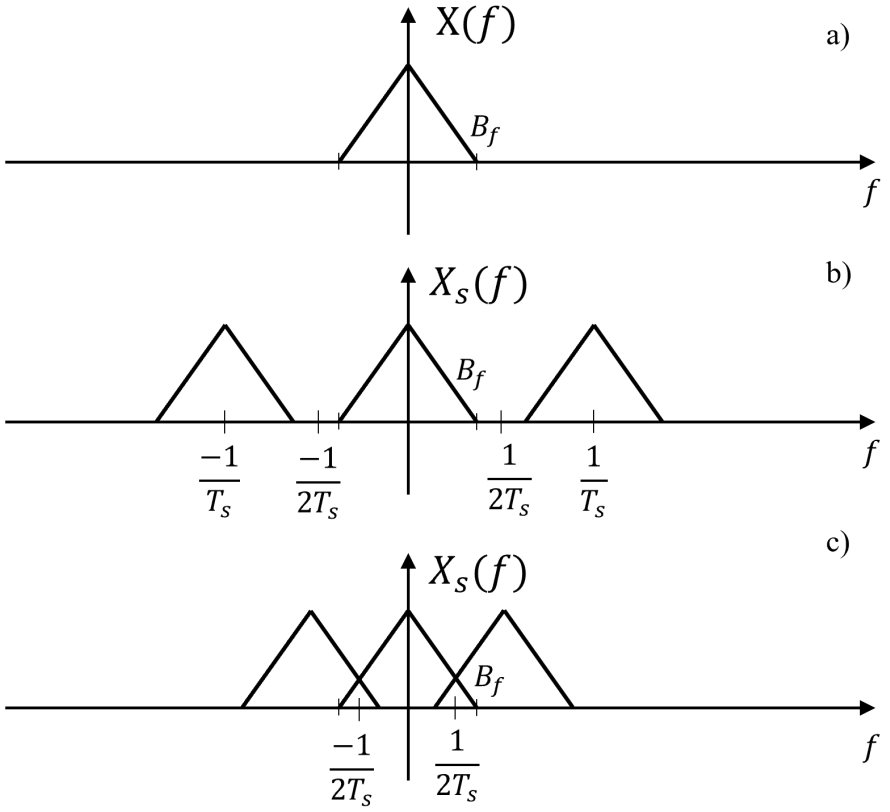

Figure 3.3: a) Baseband spectrum of the signal, b) Spectrum of the sampled signal, c) Aliasing phenomenon.

The phenomenon of overlapping spectra is called aliasing, which does not allow a correct reconstruction of the signal. To avoid this and to be able to isolate the part of the spectrum in the base band, it is necessary to respect the Sampling Theorem. The Sampling Theorem states that: a signal with a limited band B_f can be completely reconstructed from its samples if it has been sampled with frequency $f_s \geq 2B_f$, where $f_N = 2B_f$ is called Nyquist frequency.
Actually, sampling by delta train is not physically feasible. In fact, many converters require the signal to be permanently present at the input for a certain time. The circuit capable of sampling and holding the signal constant for the required time is called sample and hold.

3.1.2 Quantization

As previously stated, an analog quantity can assume infinite values within a specific range while a numerical quantity can only take on a finite set of values.
The analog signal values need to be quantized by grouping them into a certain number of levels known as quantization levels, and each level corresponds to a digital value. At this stage of the conversion, an error is inevitably introduced due to approximations, called quantization error.

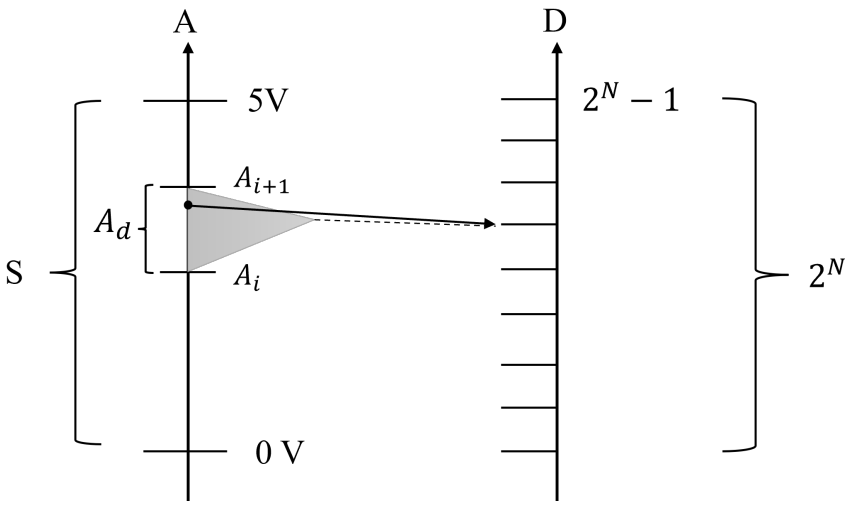

Figure 3.4: Quantization.

Uniform quantization is used to provide a numerical value to the quantization error. The uniform quantization is a type of quantization where the analog quantity's entire range is divided into equal portions.
Let us consider an analog quantity A that varies continuously within the S range, which we need to map to a 2^N bit numerical quantity D, as demonstrated in figure 3.4.
Only A_i values will be mapped exactly, while intermediate values of A will be mapped to the D value corresponding to the closest A_i. Considering that $A_d = A_{i+1} - A_i = \frac{s}{2^N} = 1LSB$, it follows that $|\epsilon_q| \leq \frac{1}{2}LSB = \frac{s}{2^N}$. It must be said that quantization error, unlike sampling error, is irreversible.

3.2 Analog-to-digital converters (ADCs)

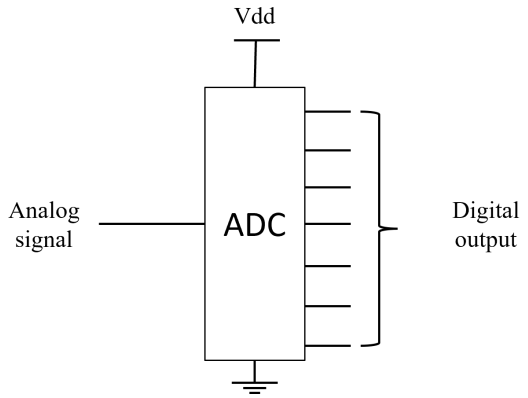

Figure 3.5: Analog to digital converter.

The conversion process is executed by the analog-to-digital converter (ADC), which is an electronic integrated circuit. The conversion can be done in several ways, in fact there are different types of ADC available. The main feature of ADC are sample rate and bit resolution.
The sample rate refers to the number of samples per second, and its unit of measurement is Hertz or samples per second. It is an indication of how fast an ADC can convert the signal.
Whereas, the resolution of the ADC relates to the number of quantization bits. The resolution of the ADC provides the number of bits necessary to encode the value of the relevant analog signal. Increased ADC resolution results in more bits and a more accurate representation of the analog signal.
The are different types of analog to digital converters, such as: Dual Slope A/D Converter, Flash A/D Converter, Semi-flash ADC, Sigma-Delta ADC, Pipelined ADC, Successive Approximation Register (SAR) A/D Converter.
Among these converters, the Sigma-Delta and the SAR will be discussed in more detail.

Sigma-Delta ADC

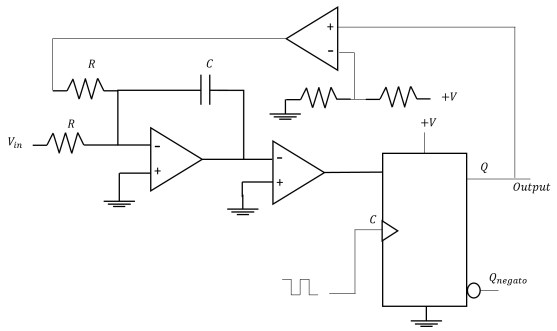

Figure 3.6: Sigma-Delta ADC.

In a $\Delta\Sigma$ converter, the analog input signal is connected to an integrator, which generates a slope at the output corresponding to the input magnitude. This ramp voltage is subsequently compared with the earth potential by a comparator. The comparator produces an output bit ('high' or 'low') based on the positive or negative value of the integrator output. The output of the comparator is then fed back into a D-type flip-flop and fed back into another input channel of the integrator, to drive the integrator towards a 0-volt output. The final comparator is necessary to transform the flip-flop's sole polarity 0V / 5V logic-level output voltage to a +V / -V voltage signal for transmission to the integrator. If the integrator output is positive, the initial comparator emits a 'high' signal to the input D of the flip-flop. At the following clock pulse, the 'high' signal will be discharged from line Q into the non-inverting input of the last comparator. This final comparator saturates in the positive direction when an input voltage above the threshold voltage of 1/2 +V is detected. This +V feedback signal tends to drive the integrator output in a negative direction. When the output voltage becomes negative, the feedback loop sends a correction signal (-V) to the upper integrator input to drive it in the positive direction. This is the delta-sigma concept in action: the first comparator detects a difference (Δ) between the integrator output and zero volts. The integrator sums (Σ) the comparator output with the analog input signal.

SAR ADC converter

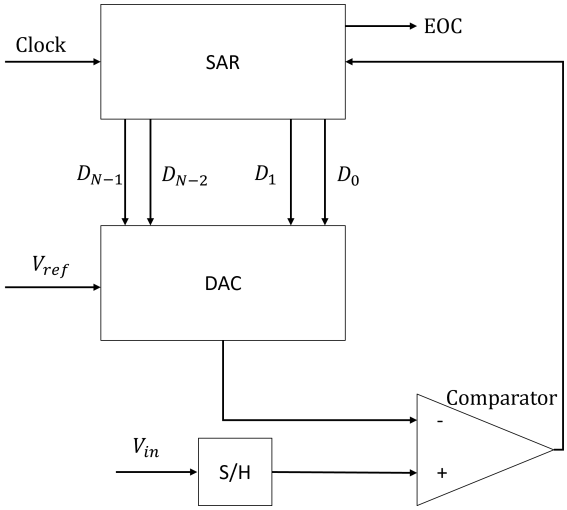

Figure 3.7: SAR ADC.

To find the digital representation of an analog input, the SAR ADC use successive approximation. There are three core functional components in a SAR ADC: a DAC that generates the comparison voltage, a SAR logic that configures the DAC and a comparator that makes the decision. A SAR conversion includes a sampling phase where V_{in} is sampled on the DAC and a conversion phase where N comparison are made for an N-bit A/D conversion. The sample-and-hold circuit is used to acquire the input voltage. Initially, the most significant bit (MSB) of the successive approximation register is set to a digital 1, and this code is supplied to the DAC. The equivalent analog voltage ($\frac{V_{ref}}{2}$) is the supplied to the comparator circuit, where it is compared with the sampled input voltage. In case this voltage exceeds V_{in}, the comparator resets the bit in the SAR; otherwise, the bit remains 1. The next bit is then set to 1, and the same test is repeated. This process continues until every bit has been tested. At the end of conversion (EOC), the SAR outputs the resulting code, which is the digital approximation of the sampled input voltage. In a SAR ADC there are various types of errors that affects the resolution. One of these is the sampling error, in which there are two major error sources in the sampling process: clock jitter and sampling noise. Then there are reference and DAC noise and also the comparator noise.

3.3 Microcontroller AURIX™ TC399

The AURIX™ TriCore™ microcontroller family, with its embedded safety and security features, is a platform ideal for a wide range of automotive and industrial applications. Combining the elements of a RISC processor core, a microcontroller, and a DSP in a single MCU design, AURIX™ TriCore™ microcontrollers are a versatile choice. The automotive applications include control of electrical and hybrid vehicles, but also control of combustion engine. For security it includes control of braking system, airbags and for the new connected cars there are advanced driver assistance systems. As previously mentioned, in this thesis, the TriBoard AURIX™ TC399 has been used. TriCore is unified, 32-bit Microcontroller DSP optimized for real-time embedded systems. It is a high performance microcontroller with multiple TriCore CPUs, program and data memories, buses, bus arbitration, interrupt system, DMA controller and a powerful set of on-chip peripherals such as serial controllers timer units, and analog-to-digital converters. All these peripheral units are connected to the TriCore CPUs/system via a System Peripheral Bus (SPB) and a Shared Resource Interconnect (SRI). The TriCore processor architecture combines three powerful technologies within one processing unit, achieving new levels of power, speed, and economy for embedded applications:

- Reduced Instruction Set Computing (RISC) processor architecture
- Digital Signal Processing (DSP) operations and addressing modes
- On-chip memories and peripherals

The RISC load/store architecture provides high computational bandwidth with low system cost. DSP operations and addressing modes provide the computational power necessary to efficiently analyze complex real-world signals. On-chip memory and peripherals are designed to support even the most demanding high bandwidth real-time embedded control-systems tasks.

3.4 Data acquisition

Regarding the signal acquisition, there are two distinct types of converters in the used board: EDSADC and EVADC. These two will be examined in the upcoming

sections, with particular attention given to the EVADC. The latter has a significant impact on enhancing performance, particularly in relation to conversion time.

3.4.1 EDSADC

The Enhanced Delta-Sigma Analog-to-Digital Converter (EDSADC) of the AURIX™ TC3X9 provides a set of up to 14 analog input channels in a single module, connected to on-chip modulators using the Delta/Sigma conversion principle.

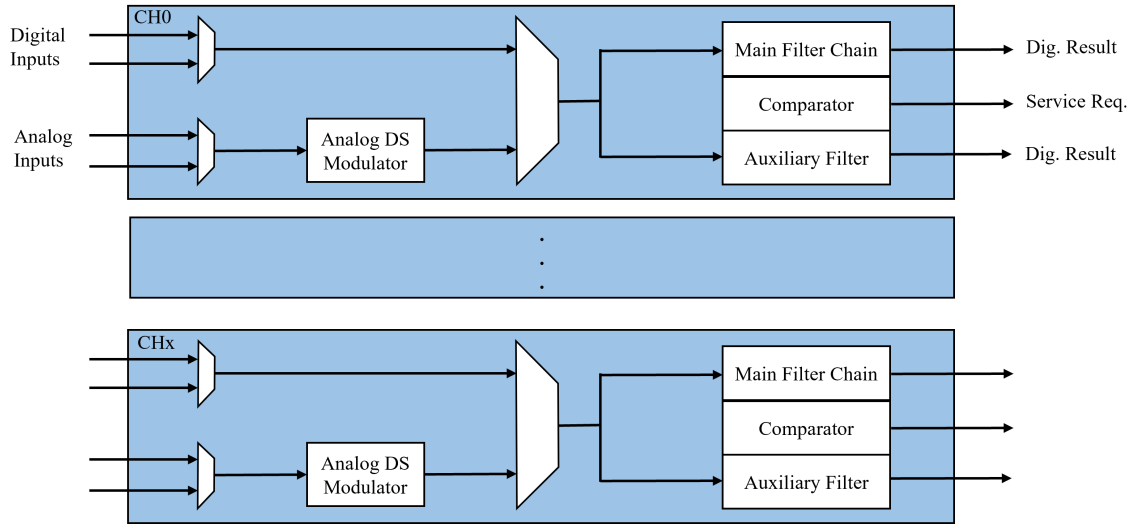

Figure 3.8: EDSADC Module.

The EDSADC module supports up to 40 MHz sampling rate and the resolution of the channels is 13 bit. Each channel can operate independent of the others. There is a demodulator, also called filter chain, composed by: the Cyclic Integrating Comb (CIC) filter that provides the basic filtering and decimation with a selectable decimation rate, two Finite Impulse Response (FIR) filters that allow signal shaping by attenuating the upper frequencies of the signal spectrum and it is also present the high-pass filter which provides offset compensation by removing the DC component of the input signal. The final result can be stored in a dedicated specific result register. EDSADC has higher accuracy with respect to EVADC and is widely used in engine pressure measurement, fuel injection control, but it is slower.

3.4.2 EVADC

The Enhanced Versatile Analog-to-Digital Converter (EVADC) provides a series of analog input channels connected to several clusters of Analog/Digital Converters using the Successive Approximation Register (SAR) principle to convert analog input values to discrete digital values. Each converter of the ADC clusters can operate independent of the others.

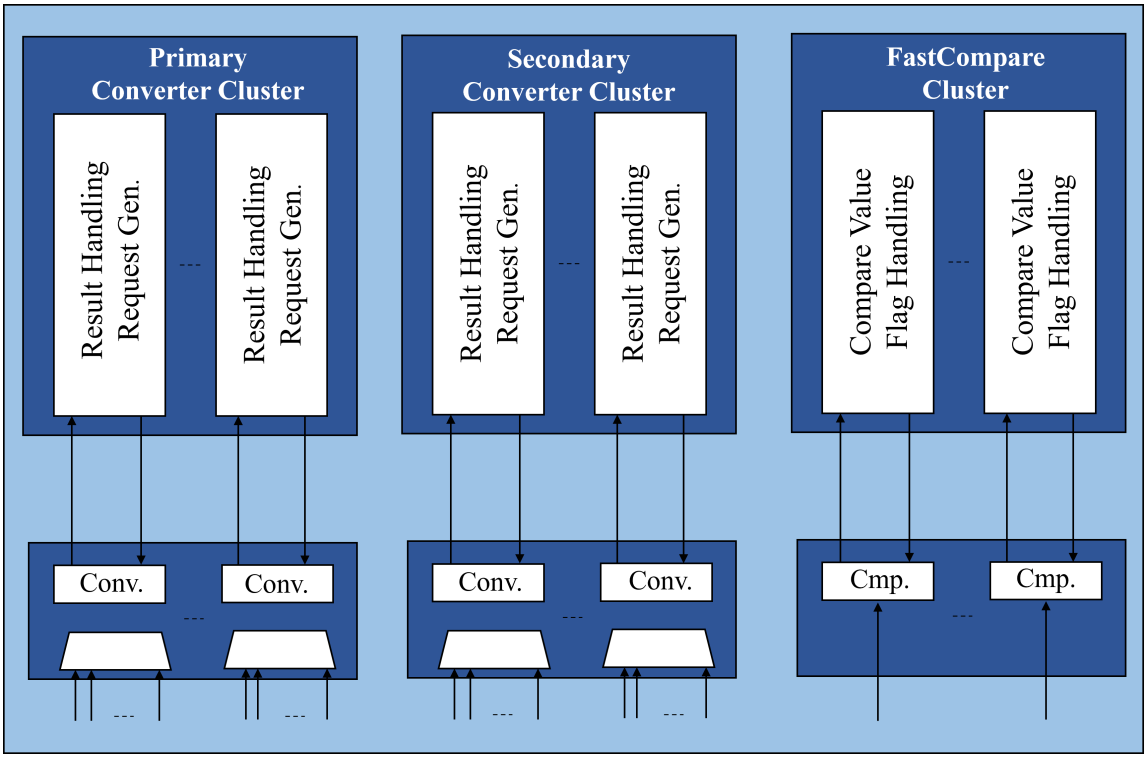

Figure 3.9: Analog-to-Digital Converter (EVADC) clusters.

As shown in the figure 3.9, three clusters are available:

- Primary converter cluster provide 8:1 input multiplexer and deliver the minimum conversion time down below 0.5 μs. In this cluster are present eight groups with eight channels each and a resolution of 12 bit;
- Secondary converter cluster provide 16:1 input multiplexers and require a higher sample time leading to an increased conversion time, but down below 1 μs. In this cluster are present four groups with sixteen channels each and a resolution of 12 bit;

- Fast converter channels provide one dedicated input channel each and deliver a compare time down below 0.2 μs. In this cluster are present 8 channels with a resolution of 10 bit;

The ADC kernel is composed by a set of functional units that can be configured to the requirements of a given application.

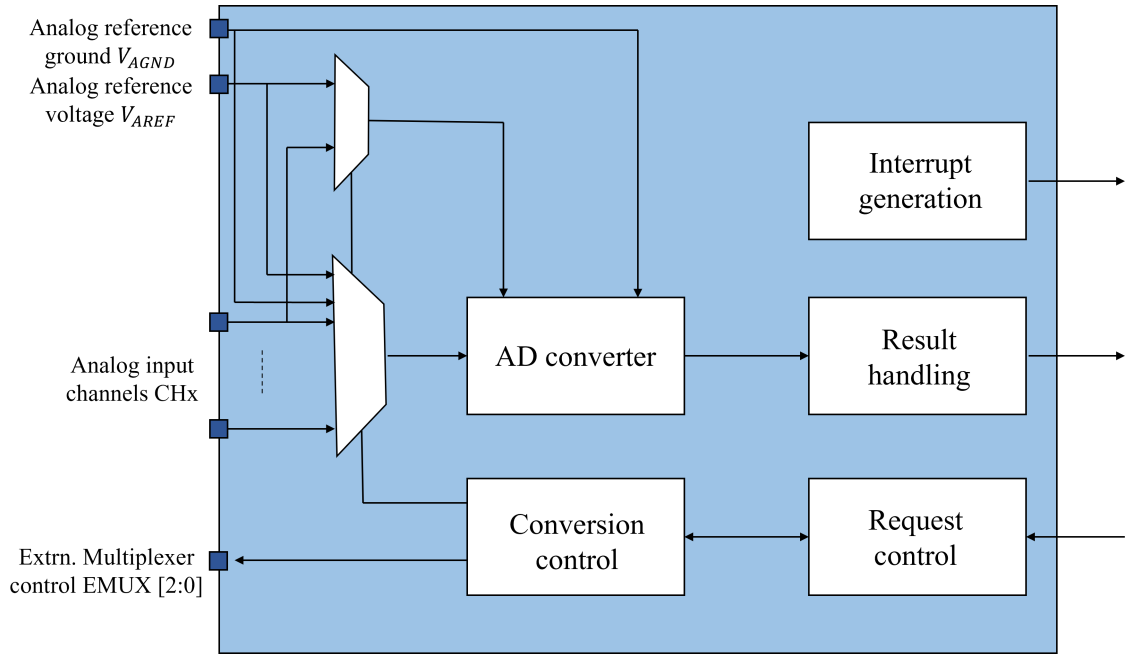

Figure 3.10: ADC kernel block.

Input channel selection

The analog input multiplexer selects one of the available analog inputs (CH0-CHx) to be converted. The priorities of these sources can be configured.

Analog/Digital converter

The voltage of the chosen input is first sampled before generating the digital result bits. A sample and hold unit is connected to a Successive Approximation Register (SAR) converter to produce the desired digital output.

Conversion control

In this block several conversion parameters can be configured, such us: sample time, conversion mode (Noise Reduction Level), clock frequency, start-up calibration and post-calibration. Thus the input channels can be adjusted to the type of sensor (or other analog sources) connected to the ADC.

Result handling

The conversion result of each analog input channel can be directed to one of 16 group-specific registers. At this point several actions can be performed such as the wait-for-read mode that avoids data loss due to result overwrite, or maybe the result registers can be concatenated to build FIFO structures. Another action can be the data reduction mode that can automatically add up to sixteen conversion results before issuing a service request, and a FIR or IIR filter can be enabled that preprocesses the conversion results.

Interrupt generation

Several ADC events can issue service request to CPU or DMA, such as source events which indicate the completion of a conversion sequence in the corresponding request source, channel events which indicate the completion of a conversion for a certain channel or result events that indicate the availability of new result data in the corresponding result register.

Request source control

All conversion requests can be enabled simultaneously with the use of an arbiter that resolves concurrent conversion requests from various sources. These requests can be triggered either by an external signal, on-chip signals or software. At fixed intervals, the arbiter will scan through the sources of these requests and choose the one with the highest priority. This selected conversion request is then directed towards the converter for the sampling and conversion of the requested channel. For each group's request source, the trigger and gating unit generates external triggers from one of 15 selectable trigger inputs (REQTRx[O:A]) and gating inputs from one of 16 selectable

inputs (REQGTx[P:A]). The trigger and gating unit generates trigger events from the selected internal or external trigger and gating signals.

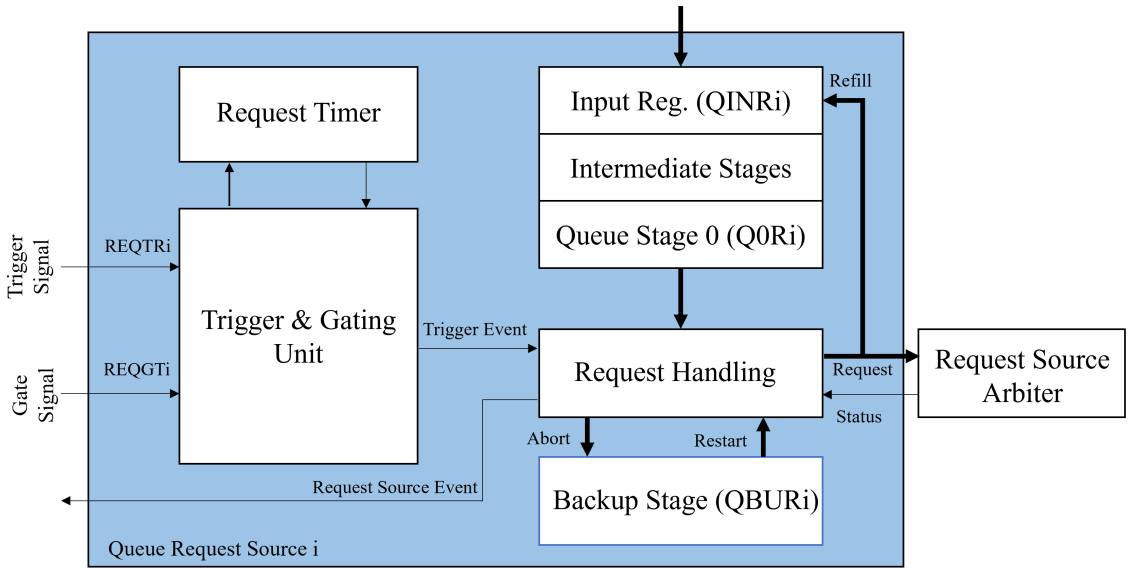

Figure 3.11: Queued request source.

Conversion timing

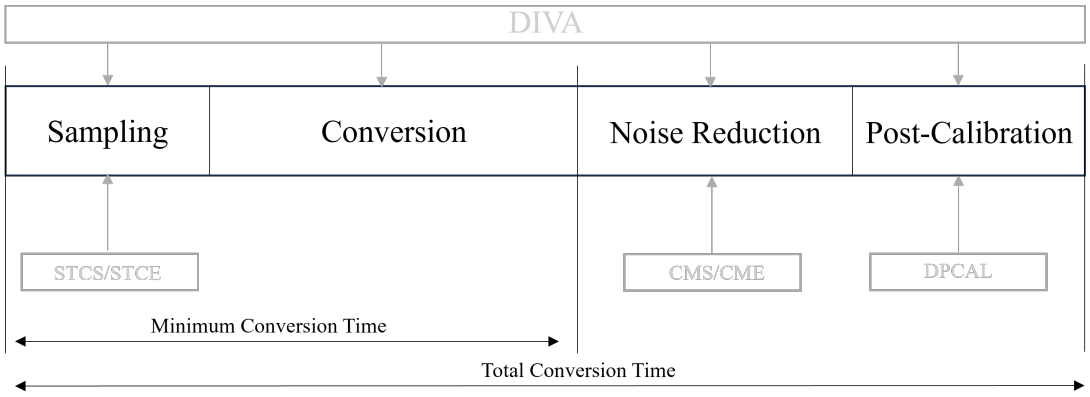

Figure 3.12: Conversion phase.

The timing of conversion encompasses not only the duration for digitization of the result, but also the time for signal sampling, calibration, and other internal steps. These factors can all be configured by the application. As shown in the figure 3.12, the

overall conversion time comprises a minimum conversion period and extra time. The former is split into sampling time and conversion time, which use the SAR principle to produce digital results. The latter includes operations like noise reduction, post-calibration and other internal steps (not depicted in the diagram) that are essential for state machine synchronization. More in detail, the conversion timing depends on the following factors that can be changed by user:

- The ADC conversion clock frequency: $f_{ADCI} = \frac{f_{ADC}}{DIVA+1}$
- The selected sample time: $t_s = (2 + STC)t_{ADCI}$, where (STC = additional sample time)
- The actual result generation (12 bit)
- The duration of the selected noise reduction conversion steps: NRS x t_{NR} , where NR = 0,1,3,7.
- The post-calibration time t_{PC}, if enabled
- Internal steps done at module clock speed

So the final formula to compute the conversion time is:

$$t_{C12} = (2 + STC) \cdot t_{ADCI} + 13 \cdot t_{ADCI} + NRS \cdot t_{NR} + t_{PC} + 3 \cdot t_{ADC} \tag{3.2}$$

where:

- $t_{NR} = 4 \cdot t_{ADCI} + 3 \cdot t_{ADC}$, per conversion step
- $t_{PC} = (4 + 2 \cdot CALSTC) \cdot t_{ADCI} + 5 \cdot t_{ADCI}$, if enabled, otherwise 0, where (CALSTC = Calibration Sample Time Control).

3.5 GTM

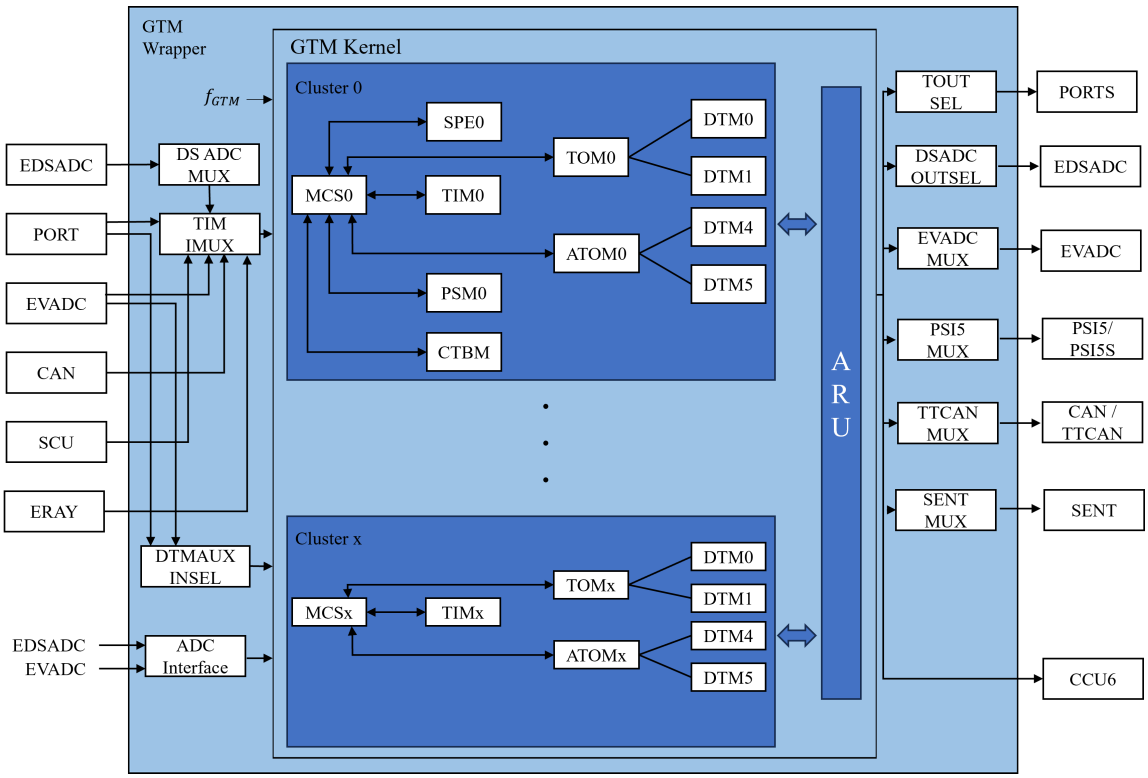

Figure 3.13: GTM Architecture.

As shown in figure 3.13, the Generic Timer Module (GTM) has a module framework with submodules of varying functionalities. These submodules can be combined in a configurable way to create a complex timer module that supports different application domains and different classes within an application domain. Dedicated hardware submodules are positioned around a central routing unit, known as the Advanced Routing Unit (ARU). The ARU can flexibly connect the submodules, whose connectivity is programmable using software, and therefore, can be configured during runtime. Nevertheless, the GTM is designed to unload the CPU or a peripheral core from a high interrupt load. The hardware sub-modules have specific functions, for example there are timer input modules that can capture and analyze incoming signals including timestamps. Other sub-modules have a more versatile design and can perform typical timer tasks, such as generating PWM signals. About this last topic, the generation PWM signals, two of the modules dedicated to this are the

TOM and the ATOM.

3.5.1 TOM

The Timer Output Module (TOM) presents up to sixteen independent channels. It is used to generate simple PWM signals, which can be dependent or independent of each other. The TOM has a dedicated PWM time base counter which is 16-bit wide, and the architecture of the channels it is shown in the figure 3.14.

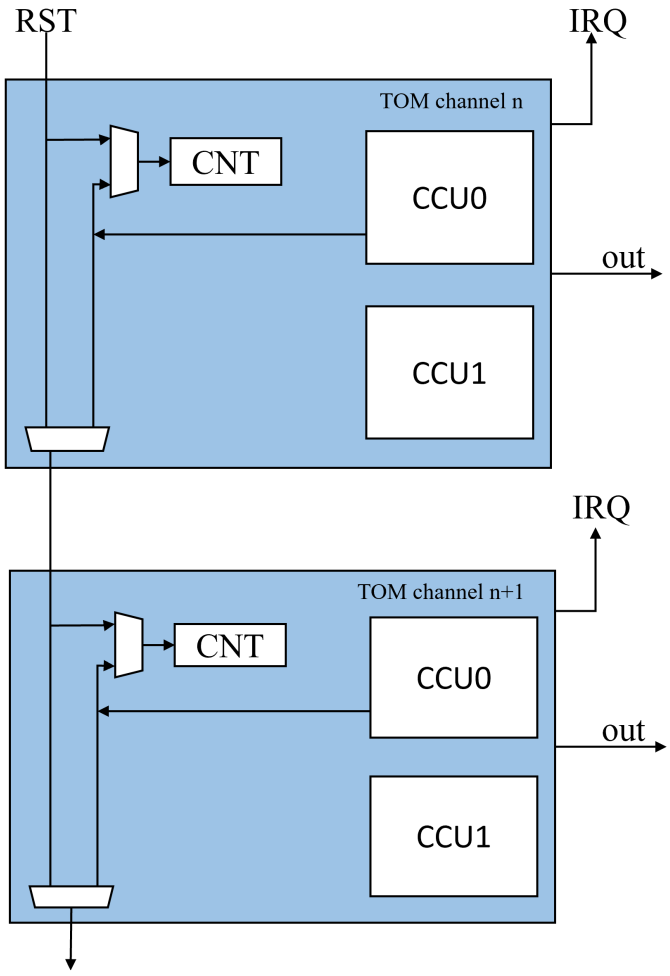

Figure 3.14: TOM Channel architecture.

The counter frequency can be selected from one of five prescaler clocks offered by the Clock Management Unit (CMU), which are: $2^0, 2^4, 2^8, 2^{12}, 2^{16}$. As can be seen, there

are two capture compare units for comparing the counter to a customizable value. CCU0 determines the duration of the PWM period, and CCU1 defines the duration of the duty cycle[1].

GTM to EVADC Connection

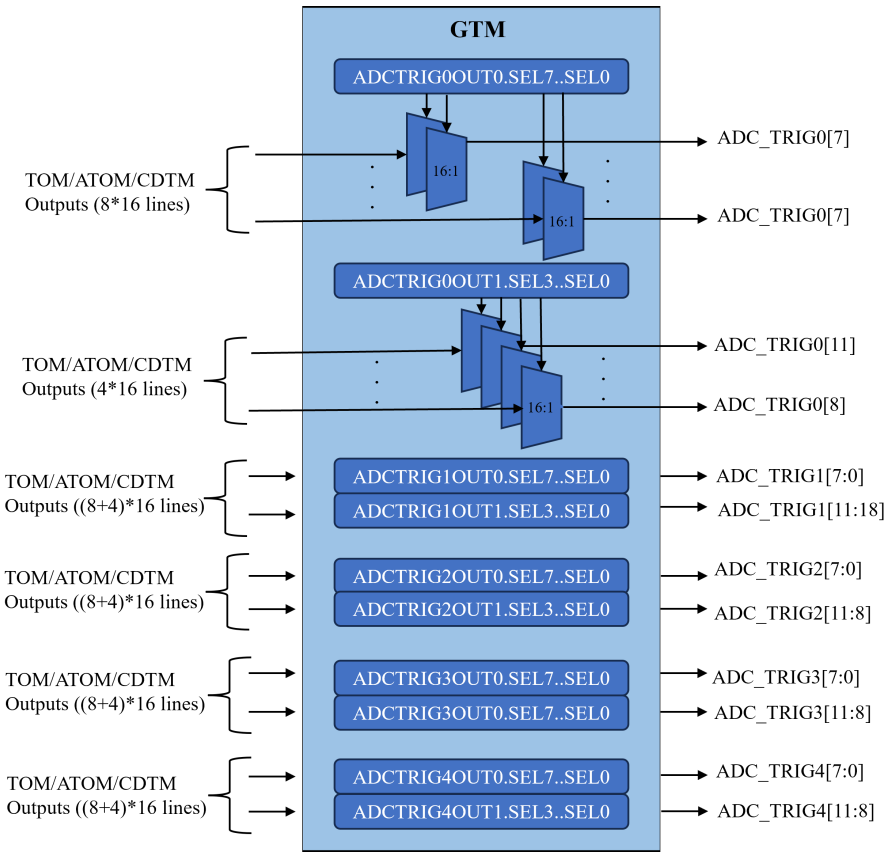

Figure 3.15: GTM to EVADC connections.

As can be seen in figure 3.15, the GTM can communicate directly with the EVADC. The PWM signals can initiate the start of the ADC conversion process by setting specific bits. More information about this will be provided in the following chapter.

[1][9]GTM-Cookbook: Overview and Application examples. URL: `https://www.bosch-semiconductors.com/media/ip_modules/pdf_2/gtm/gtm_cookbook_v05.pdf`.

3.5.2 ATOM

The ARU-connected Timer Output Module (ATOM) is able to generate complex output signals without CPU interaction due to its connectivity to the ARU. With respect to the TOM module, the ATOM presents 8 independent channels, but with a resolution of 24 bit. In the figure below it is possible to see the architecture of the channel. As the TOM submodule each ATOM channel is comprised of two compare units, its own 24 bit time base and shadow registers for independent operation of signal output and parameter reload[2].

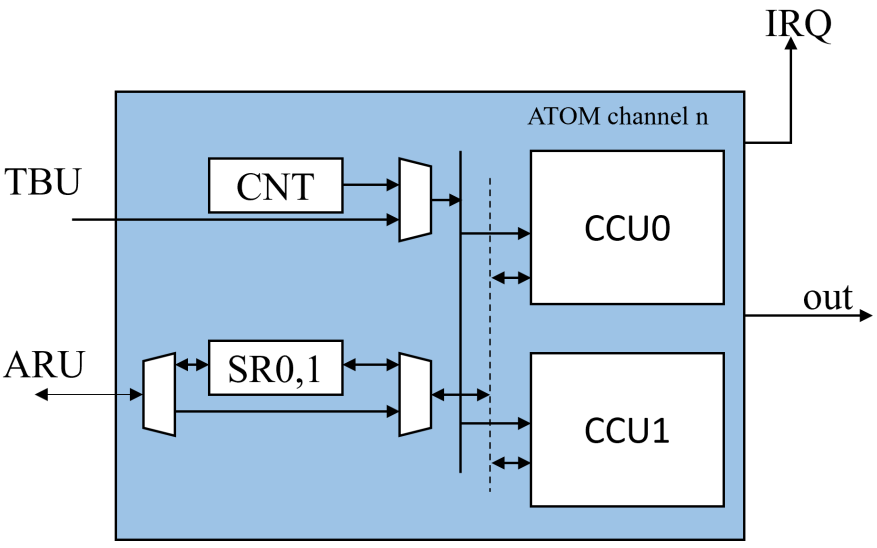

Figure 3.16: ATOM Channel architecture.

Each ATOM channel can operate in four operation modes that are: Signal Output Mode PWM (SOMP), Signal Output Mode Compare (SOMC), Signal Output Mode Immediate (SOMI), Signal Output Mode Serial (SOMS), Signal Output Mode Buffered Compare (SOMB). The initial option concerns utilizing the ATOM channel as a PWM output channel, which aligns with the focus of this study.

[2][9]GTM-Cookbook: Overview and Application examples. URL: `https://www.bosch-semiconductors.com/media/ip_modules/pdf_2/gtm/gtm_cookbook_v05.pdf`.

Chapter 4

Proposed methodology

4.1 Scope of the work

DC/DC Converter

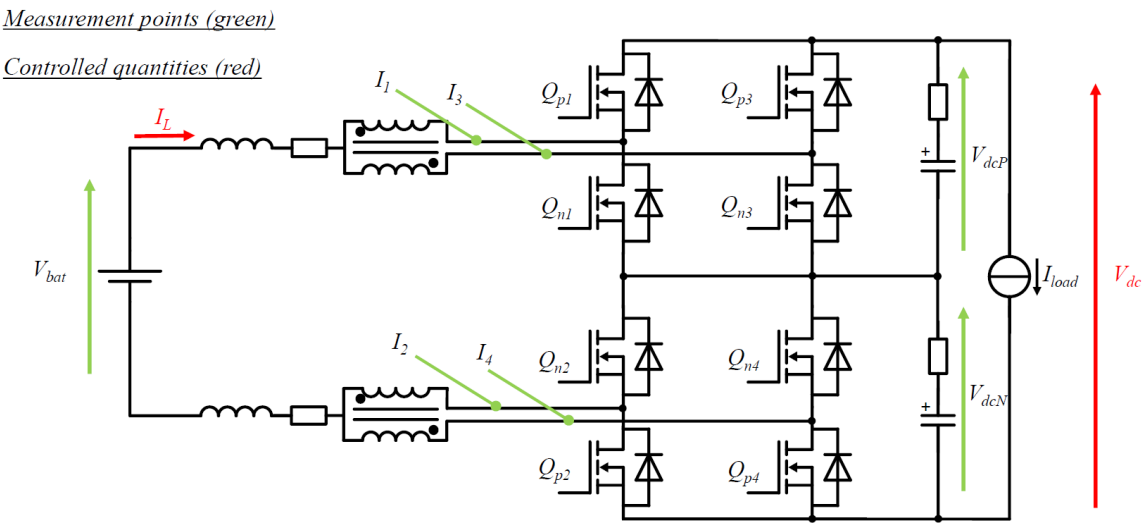

Figure 4.1: DC/DC Converter.

The objective of this thesis is to simultaneously convert seven quantities, coming from a DC/DC converter, highlighted in green in figure 4.1. This conversion should occur as swiftly as possible, as the current and voltage controllers must subsequently be applied. The duty-cycle values of the PWM waves will be determined by the

controllers' outputs, which will subsequently adjust the DC/DC converter's switches. Therefore, the sampling point and start of conversion is a fundamental aspect. Those who determine the start of conversion at a specific point will be the PWM waves themselves.

Consequently, the modules deployed within the microcontroller need to be synchronized. To accomplish this task effectively, it is crucial to select the optimal microcontroller for the job. Additionally, you must carefully choose a conversion module that guarantees swift conversion times and a timer module that ensures perfect synchronization between the produced PWM waves and the ADCs.

Sampling synchronization

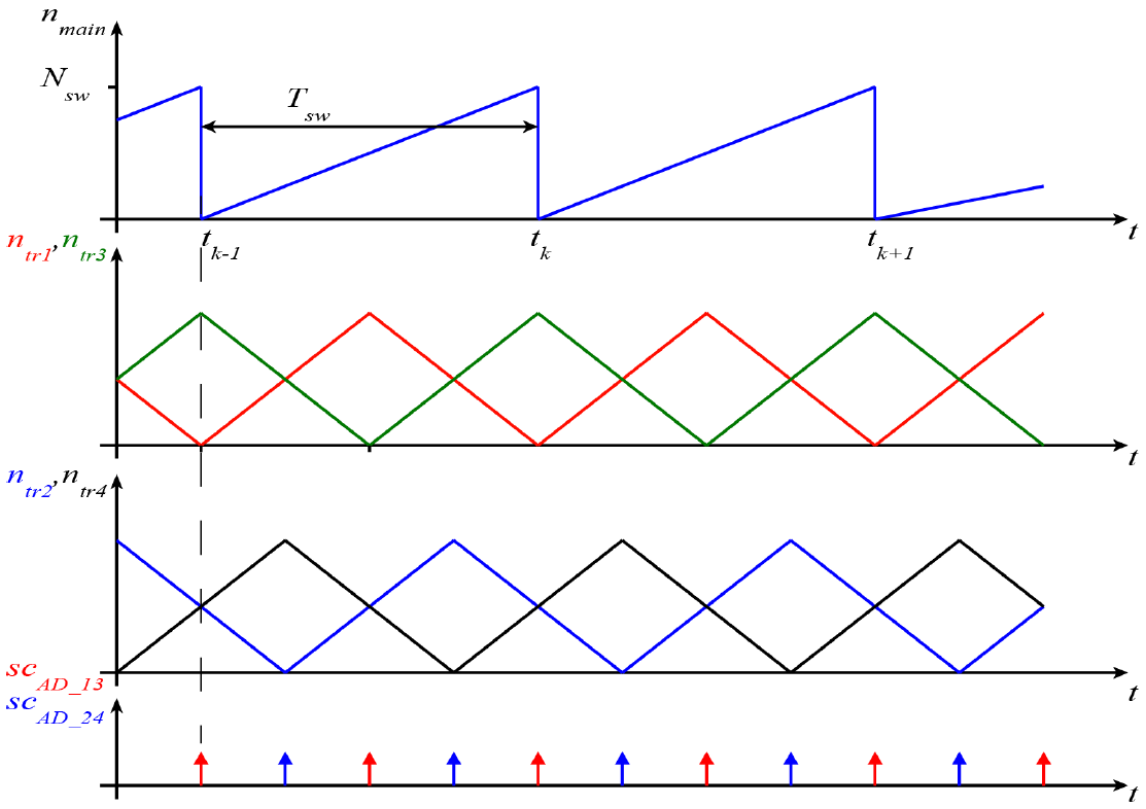

Figure 4.2: Sampling synchronization.

Given the characteristics of the ripple generated by PWM operation, it is necessary to implement double sampling in the PWM period. Each vertex of the carrier is

appropriate for sampling. In each PWM period, two samples of each feedback quantity are obtainable. The final two samples of each quantity are aggregated to derive the representative quantity for use in the control routine.

4.2 Microcontroller selection

After conducting a thorough analysis of available microcontroller options and considering various key technical factors, the AURIX™ TC399 microcontroller was ultimately chosen. This choice was made after reviewing the main contenders from Chapter 2. Its exceptional operating frequency, which far exceeded its competitors, was a major contributing factor in this decision. This feature was essential for the application as speed of execution is crucial to achieve desired performance.
Further benefits of the Infineon microcontroller include the ability to acquire multiple separate signals, indeed the AURIX™ TC399 contains more independent ADC modules, facilitating a parallel acquisition strategy that avoids overlaps and ensures accuracy. This is very important, because in the final application there is the need of acquiring seven signals from a DC/DC converter simultaneously.
Regarding the PWM signal generation, although competitors such as STM32F405xx and S32K1xx offered solutions of the same level, the primary objective of this work was reliable and fast signal acquisition.
So, the choice to utilise AURIX™ TC399 was made following a thorough decision-making process, during which the application's specific demands were thoughtfully assessed.

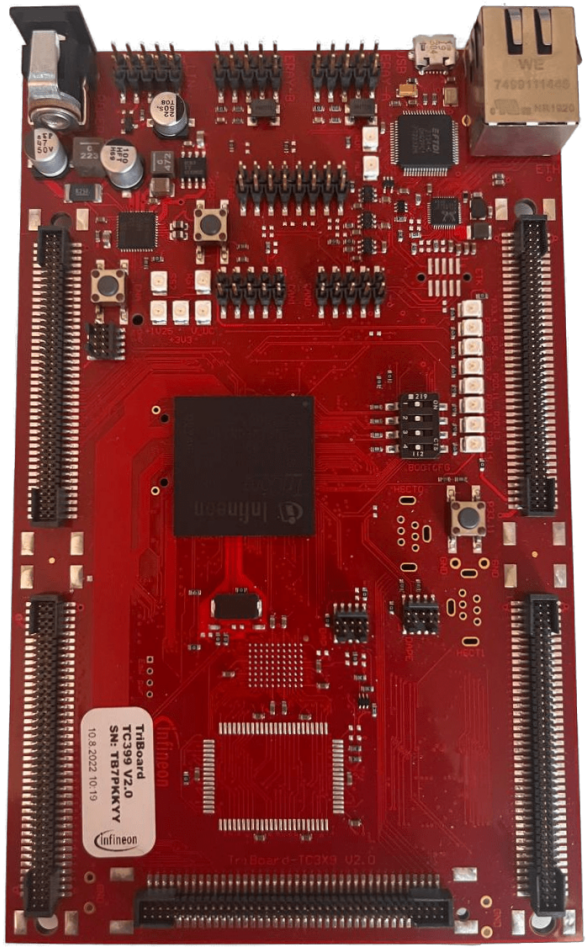

Figure 4.3: Microcontroller AURIX™ TC399.

4.3 Choice of analog-to-digital converter

As discussed earlier, the AURIX™ TC3X9 microcontroller possesses two different analog-to-digital converter modules, namely the Enhanced Versatile Analog-to-Digital Converter (EVADC) and the Enhanced Delta-Sigma Analog-to-Digital Converter (EDSADC). The former it is based on a SAR analog to digital converter, while the latter it is based on a Sigma-Delta ADC.

The decision-making process between the two types of converters was not random, but

a thorough analysis was conducted, taking into consideration their specific features. This evaluation aimed to maintain high precision and short conversion times. After all, quick and responsive system control is crucial to the success of the final application. When making this decision, the assistance provided by a scientific article was extremely valuable[1]. It employed a Sigma-Delta ADC to regulate an electric motor, requiring precise measurements of currents and voltages for optimal performance.
Nevertheless, the paper specifies how this kind of converters introduce latency in the system. It can lead to phase lag, reduce the system's gain/phase margin and cause instability. For these reasons, SAR converters are often the preferred choice due to the high conversion speeds and their low cost.
As previously discussed, the two converter technologies in the chosen microcontroller differ only in their resolution by one bit.
So, in the final application, where the TOM module and ADC module synchronize, the option of sacrifice one bit of resolution for faster conversion was considered the best.

4.4 TOM vs ATOM

In this research, a comparison was required between two timed signal generation modules: the Timer Output Module (TOM) and the ARU-connected Timer Output Module (ATOM). The aim was to determine which module was the most suitable for the final application. Each module presents unique advantages. As described before, the TOM has up to sixteen independent channels and offers the possibility of generating simple PWM signals, which can be either dependent or independent of each other, but has a dedicated counter for the PWM time base, which is 16 bits wide. However, when compared to the TOM module, the ARU-connected Timer Output Module (ATOM), which is connected to the ARU, can generate more intricate output signals without any CPU interaction. The ATOM, although, has only 8 autonomous channels but provides an impressive 24-bit resolution. Despite the higher resolution offered by the ATOM, it was decided to use the TOM for this thesis work. This

[1][10]Li, Chen, et al. "Analysis and Compensation of Sigma-Delta ADC Latency for High Performance Motor Control and Diagnosis." IEEE Transactions on Industry Applications 59.1 (2022): 873-885.

decision was driven by the need to have more channels available with a view to creating an application where 9 to 16 PWM waves are required. These must be generated using the same timer module, as using two separate modules would have led to delays in the synchronization of the waves. This is why for this type of application it is not possible to use two different ATOM modules to generate 16 PWM waves. It should be noted that there would not have been a significant difference in using either the ATOM or the TOM for the final implementation of this project, as only two PWM waves will be produced, but again, it was decided to implement the TOM for future operations.

4.5 Simple application

The purpose of this straightforward application is to develop an application that can efficiently convert a single analog signal in a continuous manner, without requiring a trigger. The particular feature of this application is that it is possible to modify the various parameters (number of group, number of channel, sample time, analogue frequency, noise reduction level and post-calibration), directly from the main before running the code. Below it is reported the pseudo-code.

```
void initEVADC(Group, Channel, Sample_time, Analog_frequency,
   Noise_reduction_level, PostCalibration){

%----------------------------GROUP-----------------------------
IfxEvadc_Adc_GroupConfig adcGroupConfig;
IfxEvadc_Adc_initGroupConfig(&adcGroupConfig, &g_evadc);

    adcGroupConfig.groupId = Group;
    adcGroupConfig.sampleTime = Sample_time;
    adcGroupConfig.conversionMode = Noise_reduction_level;
    adcGroupConfig.analogFrequency = Analog_frequency;
    adcGroupConfig.gatingMode = IfxEvadc_GatingMode_always;
    adcGroupConfig.PC = PostCalibration; % 0 Enabled, 1 Disabled
%----------------------------CHANNEL-----------------------------
    adcChannelConfig.channelId = Channel;
    adcChannelConfig.resultRegister = Channel;

void readEVADC(void)
```

```
{
    Ifx_EVADC_G_RES conversionResult;
    % Wait for valid result
    do{
          conversionResult = IfxEvadc_Adc_getResult(&Channel);
    }while(!conversionResult.B.VF);
    g_result = conversionResult; % Store result
}
```

As it can be observed, the primary aspects of configuring an EVADC module consist of group and channel setup. Within the group, the following parameters are established:

- group number
- sample time and analog frequency
- enable or disable post-calibration
- the level of noise reduction is imposed

In the channel, only the number and corresponding result register are set. The readEVADC function is used to read the result register as soon as it is considered a valid result.

4.6 Conversion time computation

Regarding the conversion times, it is not possible to rely exclusively on the values described in the datasheet. This is because the calculations are carried out theoretically or the conditions under which the various modules were tested are ideal. Therefore, it is necessary to verify the conversion times by physical use of the available board. Two different methods were implemented for this experiment: the first is a so-called external measurement, while the second is an internal measurement.

Method 1 - External measurement

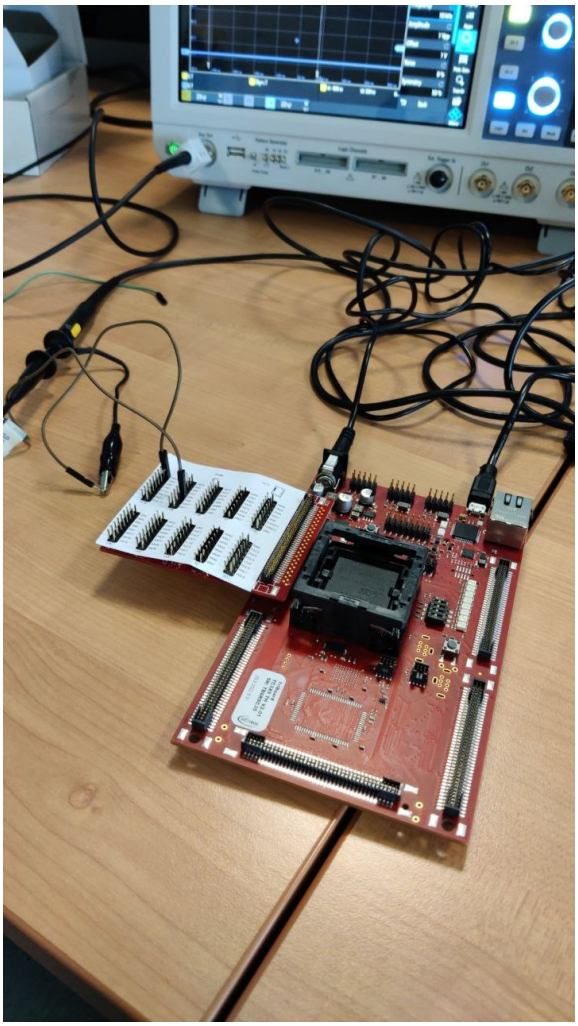

Figure 4.4: Set-up of method 1.

Firstly, the code is written in the C language using the AURIX™ Development Studio environment. The code is then uploaded to the board via the USB port. Next, the AURIX™ TC399 board is connected to the oscilloscope through pin 13, port 3, as shown in figure 4.4.

As can be noted from the code provided below, the function *IfxPort_setPinMode(PIN, IfxPort_Mode_outputPushPullGeneral)* sets the pin 13 to push-pull, general-purpose output.

This code is part of a larger function (readEVADC) that reads the output register

value once a valid conversion occurs. Moreover, pin 13 is set 'high' at the beginning of the conversion and 'low' when it ends. In this way, a periodic square wave can be seen on the oscilloscope and the conversion time can be deduced from it. The signal to be converted is transmitted to the board using a signal generator.

```
% This function read the value of EVADC register and set the PIN
void readEVADC(void)
{
    Ifx_EVADC_G_RES conversionResult;
    IfxPort_setPinMode(PIN, IfxPort_Mode_outputPushPullGeneral);

    % Wait for valid result
    do{
        % Read the result of the channel
        conversionResult = IfxEvadc_Adc_getResult(&g_adcChannel);
        IfxPort_setPinState(PIN, IfxPort_State_high); % Set PIN High
    }while(!conversionResult.B.VF);
    IfxPort_setPinState(PIN, IfxPort_State_low);      % Set PIN Low
    g_result = conversionResult;                      % Store result
}
```

Method 2 – Internal measurement

A different approach to calculating conversion times eliminates the need for an oscilloscope and only requires a signal generator. This is possible because a timing measurement is performed internally on the board using the System Timer Module (STM)[2]. In particular, the following algorithm is specified, which is further elaborated in the code presented below:

- an interrupt is defined when the conversion ends
- thanks to the function of the STM module, *IfxStm_get(IFXSTM_DEFAULT_TIMER) & TIME_INFINITE* , the number of ticks from the beginning to the end of the conversion is counted

[2][11]Pag 70 from: Ceglia, Simone. Sviluppo di una scheda elettronica digitale nell'ambito della sicurezza industriale= Development of a digital electronic board in the field of industrial safety. Diss. Politecnico di Torino, 2023.

- another function of the STM, *IfxStm_getTicksFromMicroseconds(BSP_DEFAULT_TIMER, 1)*, returns the number of ticks present in a microsecond
- finally, the time base is reconstructed using the algorithm form lines 7 to 10

```
% Take the number of ticks
ticks[j] = (uint64)IfxStm_get(IFXSTM_DEFAULT_TIMER) & TIME_INFINITE;
% Number of ticks in 1us
ticks_in_a_us = IfxStm_getTicksFromMicroseconds(BSP_DEFAULT_TIMER,1)

% Reconstruct the base time
for(i=1; i<=NUMBER_OF_SAMPLES; i++) {
    ticks_between_samples = ticks[i] - ticks[i-1];
    us_between_samples = ticks_between_samples/ticks_in_a_us;
}
```

Application 1

This application is designed to calculate conversion times using Method 1. A simple code is implemented where a single group and a single channel of the EVADC module is set up and a free-running conversion is performed without triggering. Twelve tests are performed in which the following parameters are changed: post-calibration, noise reduction level and analogue frequency, while the sample time is kept constant. The objective of these tests is to compare the conversion times on the datasheet with the 'real' conversion times that we can observe using the oscilloscope. Below, a pseudo-code outlining the general set-up of the application is provided.

```
 %---------------------------EVADC GROUP-------------------------
IfxEvadc_Adc_GroupConfig adcGroupConfig;
IfxEvadc_Adc_initGroupConfig(&adcGroupConfig, &g_evadc);

    adcGroupConfig.groupId = 0;
    adcGroupConfig.sampleTime = 100e-9; % 100 ns
    adcGroupConfig.conversionMode = Noise_reduction_level;
    adcGroupConfig.analogFrequency = Analog_frequency;
    adcGroupConfig.PostCalibration = PostCalibration;
% Post-calibration = 0 -> Enabled;  Post-calibration = 1 -> Disabled
    adcGroupConfig.PreCalibration = Precalibration;
    adcGroupConfig. triggerConfig.gatingMode = Always;
```

```
    adcGroupConfig. triggerConfig.triggerMode = NoTrigger;
 %--------------------------EVADC CHANNEL-------------------------
IfxEvadc_Adc_ChannelConfig adcChannelConfig;
IfxEvadc_Adc_initChannelConfig(&adcChannelConfig, &g_adcGroup);
    adcChannelConfig.channelId = 0;
    adcChannelConfig.resultRegister = 0;
```

Group 0, channel 0, result register 0 of EVADC module has been set. The Noise Reduction Level can vary from 0 to 3 depending on the test conducted, same for the post-calibration which can be enabled or disabled. The analog frequency can take a value of either 20MHz, 26.7MHz, 40MHz, or 53.3MHz. Additionally, there is a pre-calibration parameter that can be enabled or disabled. Although it does not impact the final conversion time, it holds significant importance. In fact, by enabling pre-calibration, it is possible to disable post-calibration, thereby reducing conversion times.

Application 2

The application 2 is designed to calculate conversion times using Method 2. A simple code is implemented where a single group and a single channel of the EVADC module is set up and a free-running conversion is performed without triggering. Twelve tests are also performed in which the same parameters as before are changed: post-calibration, noise reduction level and analog frequency, while the sample time is kept again constant. The objective of these tests is to compare the conversion times on the datasheet with the 'real' conversion times. Below, a pseudo-code outlining the general set-up of the application is provided.

```
 %--------------------------EVADC GROUP-------------------------*/
IfxEvadc_Adc_GroupConfig adcGroupConfig;
IfxEvadc_Adc_initGroupConfig(&adcGroupConfig, &g_evadc);

    adcGroupConfig.groupId = 0;
    adcGroupConfig.sampleTime = 100e-9; // 100 ns
    adcGroupConfig.conversionMode = Noise_reduction_level;
    adcGroupConfig.analogFrequency = Analog_frequency;
    adcGroupConfig.PostCalibration = PostCalibration;
% Post-calibration = 0 -> Enabled;  Post-calibration = 1 -> Disabled
    adcGroupConfig.PreCalibration = Precalibration;
```

```
12      adcGroupConfig. triggerConfig.gatingMode = Always;
13      adcGroupConfig. triggerConfig.triggerMode = NoTrigger;
14  %---------------------------EVADC CHANNEL--------------------------
15  IfxEvadc_Adc_ChannelConfig adcChannelConfig;
16  IfxEvadc_Adc_initChannelConfig(&adcChannelConfig, &g_adcGroup);
17      adcChannelConfig.channelId = 0;
18      adcChannelConfig.resultRegister = 0;
```

4.7 Final application

For the final application there are seven analog input signals arriving from a DC/DC converter. Four of these signals are currents (I_1, I_2, I_3, I_4), while the other three are voltages ($V_{batt}, V_{dcP}, V_{dcN}$), as shown in the figure 4.1. Therefore, seven distinct groups and seven channels of the EVADC module have been established. In this application the conversion is not free-running, but the sample point is determined by a trigger that originates from the PWM signals. For this reason a module of the Timer Output Module (TOM) is configured.

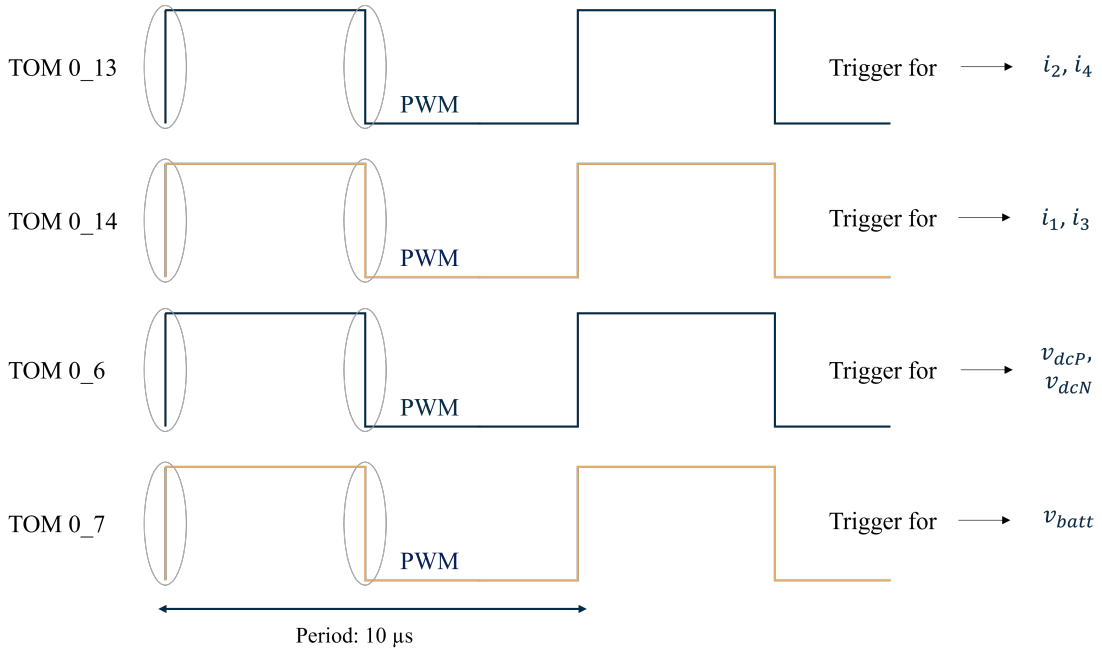

Figure 4.5: PWM signals

As displayed in the figure 4.5, four PWM with a period of 10 μs are generated. TOM 0 channel 14 generate a PWM which serves as a trigger for I_1, I_3 Similarly, TOM 0 channel 13 generates a PWM and is used as a trigger for I_2, I_4. Sampling can be conducted at each edge (rising and falling) of the square wave, resulting in a sample taken every 5 μs. The square wave corresponds to the triangular carriers that moves up and down as shown in the figure 4.2. Sampling in both wavefronts equals sampling at the apex of each triangle and at the zero point. Actually, when the sampling of the ADC synchronizes with the PWM timer clock, triggering the sampling of the ADC when the timer counter reaches zero, it makes the ADC susceptible to the switching noise induced when the region starts to operate. For this reason, it is recommended to synchronize the ADC sampling with the PWM at the timer counter peak[3]. In this application, it was determined that sampling twice per period is necessary as two samples must be taken and then averaged. Regarding the voltage sampling $(V_{dcP}, V_{batt}, V_{dcN})$, unlike with currents, there are no specific constraints on where to take the samples. The goal is to produce two more PWMs under the same conditions as the previous PWMs. It is essential that all PWMs are generated within the same TOM cluster to avoid any synchronization issues between the waves.
Now the challenge is to find a configuration that can achieve quick conversion times without compromising on resolution and synchronization with PWM signals.
Below, it is reported the pseudo-code of the set-up.

```
                        TOM CONFIGURATION
void init_GTM_TOM(void){
    % Enable GTM and FXU clock (fixed clock unit 100MHz)
    IfxGtm_enable(tom0_14);
    IfxGtm_Cmu_enableClocks(tom0_14, IFXGTM_CMU_CLKEN_FXCLK);

    g_tomConfig[0].tom = 0;           % TOM 0
    g_tomConfig[0].tomChannel =14;    % Channel 14
    g_tomConfig[0].period = 1000;     % Period: 100kHz = 1000 ticks
    g_tomConfig[0].dutyCycle = 500;   % Duty Cycle: 50
    % Trigger the Request Timer of the ADC
    IfxGtm_Trig_toEVadc(tom0_14, IfxGtm_Trig_AdcGroup_1,
```

[3][12]Zhang, Zhe, et al. "Optimized digital implementation of carrier-based randomized discontinuous PWM technique for active front end (AFE) drives." 2019 IEEE Energy Conversion Congress and Exposition (ECCE). IEEE, 2019.

```
    IfxGtm_Trig_AdcTrig_0,IfxGtm_Trig_AdcTrigSource_tom0,
    IfxGtm_Trig_AdcTrigChannel_14);          %i1
13   IfxGtm_Trig_toEVadc(tom0_14, IfxGtm_Trig_AdcGroup_3,
    IfxGtm_Trig_AdcTrig_0,IfxGtm_Trig_AdcTrigSource_tom0,
    IfxGtm_Trig_AdcTrigChannel_14);          %i3
14   %-------------------------------------------------------------
15   % Enable GTM and FXU clock (fixed clock unit 100MHz)
16   Ifx_GTM *tom0_13 = &MODULE_GTM;
17   IfxGtm_enable(tom0_13);
18   IfxGtm_Cmu_enableClocks(tom0_13, IFXGTM_CMU_CLKEN_FXCLK);
19
20   g_tomConfig[1].tom = 0;           % TOM 0
21   g_tomConfig[1].tomChannel = 13;   % Channel 13
22   g_tomConfig[1].period = 1000;     % Period: 100kHz = 1000 ticks
23   g_tomConfig[1].dutyCycle = 500;   % Duty Cycle: 50
24
25   % Trigger the Request Timer of the ADC
26   IfxGtm_Trig_toEVadc(tom0_13, IfxGtm_Trig_AdcGroup_2,
    IfxGtm_Trig_AdcTrig_0,IfxGtm_Trig_AdcTrigSource_tom0,
    IfxGtm_Trig_AdcTrigChannel_13);        %i2
27   IfxGtm_Trig_toEVadc(tom0_13, IfxGtm_Trig_AdcGroup_4,
    IfxGtm_Trig_AdcTrig_0,IfxGtm_Trig_AdcTrigSource_tom0,
    IfxGtm_Trig_AdcTrigChannel_13);        %i4
28   %-------------------------------------------------------------
29   % Enable GTM and FXU clock (fixed clock unit 100MHz)
30   Ifx_GTM *tom0_6 = &MODULE_GTM;
31   IfxGtm_enable(tom0_6);
32   IfxGtm_Cmu_enableClocks(tom0_6, IFXGTM_CMU_CLKEN_FXCLK);
33
34   g_tomConfig[2].tom = 0;           % TOM 0
35   g_tomConfig[2].tomChannel = 6;    % Channel 6
36   g_tomConfig[2].period = 1000;     % Period: 100kHz = 1000 ticks
37   g_tomConfig[2].dutyCycle = 500;   % Duty Cycle: 50
38
39   % Trigger the Request Timer of the ADC
40   IfxGtm_Trig_toEVadc(tom0_6, IfxGtm_Trig_AdcGroup_5,
    IfxGtm_Trig_AdcTrig_0,IfxGtm_Trig_AdcTrigSource_tom0,
    IfxGtm_Trig_AdcTrigChannel_6);          %Vdc_P
41   IfxGtm_Trig_toEVadc(tom0_6, IfxGtm_Trig_AdcGroup_6,
    IfxGtm_Trig_AdcTrig_0,IfxGtm_Trig_AdcTrigSource_tom0,
```

```
    IfxGtm_Trig_AdcTrigChannel_6);        %Vdc_N
42        %-------------------------------------------------------------
43     % Enable GTM and FXU clock (fixed clock unit 100MHz)
44     Ifx_GTM *tom0_7 = &MODULE_GTM;
45     IfxGtm_enable(tom0_7);
46     IfxGtm_Cmu_enableClocks(tom0_7, IFXGTM_CMU_CLKEN_FXCLK);
47
48     g_tomConfig[3].tom = 0;            % TOM 0
49     g_tomConfig[3].tomChannel = 7;     % Channel 7
50     g_tomConfig[3].period = 1000;      % Period: 100kHz = 1000 ticks
51     g_tomConfig[3].dutyCycle = 500;    % Duty Cycle: 50
52
53     % Trigger the Request Timer of the ADC
54     IfxGtm_Trig_toEVadc(tom0_7, IfxGtm_Trig_AdcGroup_7,
    IfxGtm_Trig_AdcTrig_0,IfxGtm_Trig_AdcTrigSource_tom0,
    IfxGtm_Trig_AdcTrigChannel_7);          %Vbatt
55 }
```

For the TOM module, the Fixed Clock Unit is first set to 100MHz. This value is useful for adjusting the period and duty cycle of the PWM signal. The period must be set in ticks and in this case a frequency of 100kHz is required, which is equal to 1000 thanks to the formula: 100 MHz/100kHz. To send the trigger to EVADC, the function *IfxGtm_Trig_toEVadc* is used, which inputs are:

- trigger source, i.e. module and channel of TOM
- destination, the EVADC group number
- type of the trigger, in this case for TOM 0 channel 6,7,13,14, is ADC trigger 0

```
1                       EVADC CONFIGURATION
2 void init_EVADC(void){
3
4     % Create and initializes module configuration
5     IfxEvadc_Adc_Config adcConfig;
6     IfxEvadc_Adc_initModuleConfig(&adcConfig, &MODULE_EVADC);
7     %                          GROUPS
8     % GROUP 1,3,5,7 - Primary Group
9     adcGroupConfig.groupId = 1,3,5,7;
10    adcGroupConfig.queueRequest.gatingSource = 0;
```

```
    adcGroupConfig.queueRequest.triggerMode = Upon Any Edge;
    adcGroupConfig.queueRequest.triggerSource = 8;
    adcGroupConfig. PostCalibration = Disabled;
    adcGroupConfig.sampleTime = 100e-9;
    adcGroupConfig.conversionMode = NRL 0 ;
    adcGroupConfig.analogFrequency = 53.3 MHz;
    adcGroupConfig.startupCalibration = Enabled;
    g_adcGroup.group->Q[0].QINR.B.EXTR = 1; % ->
    % -> A conversion request is only issued by a trigger event

    % GROUP 2,4,6 - Primary Group
    adcGroupConfig.groupId = 2,4,6;
    adcGroupConfig.queueRequest.gatingSource = 0;
    adcGroupConfig.queueRequest.triggerMode = Upon Any Edge;
    adcGroupConfig.queueRequest.triggerSource = 8;
    adcGroupConfig. PostCalibration = Disabled;
    adcGroupConfig.sampleTime = 100e-9;
    adcGroupConfig.conversionMode = NRL 0 ;
    adcGroupConfig.analogFrequency = 53.3 MHz;
    adcGroupConfig.startupCalibration = Enabled;
    g_adcGroup.group->Q[0].QINR.B.EXTR = 1; % ->
    % -> A conversion request is only issued by a trigger event

    %                          CHANNELS
    % Create and initializes the channel configuration
    IfxEvadc_Adc_ChannelConfig adcChannelConfig[CHANNEL_NUMBER];

    % GROUP 1 - Primary Group
    for(i=0; i<NUM_GROUPS; i++){
        adcChannelConfig[0].channelId = IfxEvadc_ChannelId_0;
        adcChannelConfig[0].resultRegister =
   IfxEvadc_ChannelResult_0;
    }
}
```

For the EVADC, the above configuration has been chosen to ensure the shortest conversion time possible. In addition to the number of groups and channels, the configuration includes the following settings:

- sample rate of 100 ns

- analog frequency of 53.3MHz
- noise reduction level set to 0
- pre-calibration enabled, which allows the disabling of post-calibration
- post-calibration disabled

For what regard the trigger, firstly the command *Q[0].QINR.B.EXTR = 1* is set to allow conversion only when there is an external trigger. Next, the trigger mode is set to be upon any edge, with the trigger and gating sources set to 8 and 0 respectively. These values are taken from the User Manual[4]. Below it is reported the flow chart of the application.

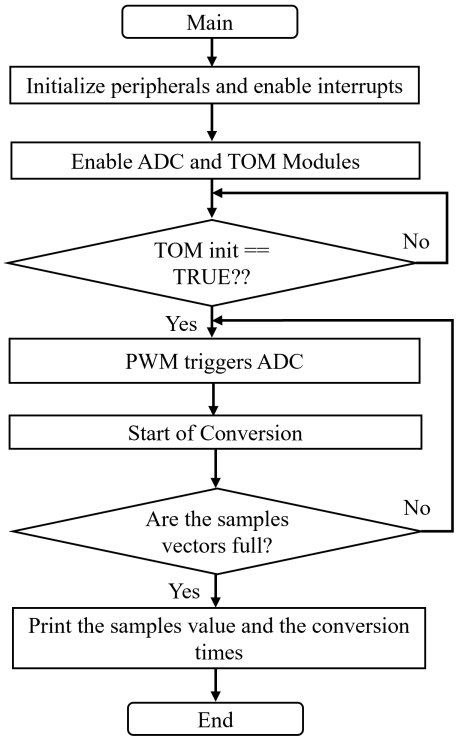

Figure 4.6: Flow chart final application

To clarify the point regarding calibration, some explanations must be provided. Calibration automatically corrects inconsistencies caused by process variation and aging,

[4][13]User Manual v01. URL: `https://www.infineon.com/dgdl/Infineon-AURIX_TC39x-UserManual-v02_00-EN.pdf?fileId=5546d462712ef9b7017182d371dc1d95`.

ensuring exact results throughout operation. An initial Pre-Calibration is necessary once after resetting all converters that are enabled. Subsequently, calibration cycles can offset the impacts of shifting external parameters, if not deactivated. The converter itself does not require any further calibration. Post-Calibration steps can be added to each conversion. Since aging causes very slow variation and post-calibration leaves the capacitor discharged, it is possible to disable post-calibration, especially since most applications undergo periodic restarts that repeat start-up calibration.

Chapter 5

Results

5.1 Result of the simple application

The objective of this application is to verify the accurate conversion of an analog signal. A signal generator's output is connected to pin AN0 of the board, followed by the setting of group 0 and channel 0 of the EVADC module in AURIX Development Studio. As other parameters such as sample time, noise reduction level, analogue frequency, and post-calibration are irrelevant to this application, they are randomly set.

The application is working correctly and this can be noticed by checking the value of the result register of the EVADC (EVADC_G0RES0). To confirm the values of the registers, the execution of the application was interrupted and the values of the resisters were checked.

Table 5.1: EVADC register G0RES0 – Simple Application

EVADC_G0RES0	0x800007a2	
RESULT	0x7a2	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	1 – No new result available	31: Valid Flag

5.2 Result of the application 1

The objective of this application is to utilise a single group and a single channel of the EVADC to convert an analogue signal, with a focus on the conversion times employing method 1. Different settings of the EVADC are tested by changing the following parameters: post-calibration, noise reduction level and analogue frequency, while keeping the sample time constant.
The following results are shown in the table below.

Table 5.2: Conversion Timing Considering External Measurement – Method 1

Test	Analog Frequency (MHz)	Sample time (ns)	Noise Reduction Level	Post-calibration	Datasheet Value (ns)	Mean Value (ns)
1	20.0	100	3	Yes	1656.25	2640
2	20.0	100	0	Yes	1000	1140
3	20.0	100	0	No	768.75	796.1
4	26.7	100	3	Yes	1306.25	2025
5	26.7	100	0	Yes	800	826
6	26.7	100	0	No	618.75	570.7
7	40.0	100	3	Yes	931.25	1480
8	40.0	100	0	Yes	575	662.3
9	40.0	100	0	No	443.75	640.7
10	53.3	100	3	Yes	800	1150
11	53.3	100	0	Yes	518.75	661.7
12	53.3	100	0	No	375	365

It should be noted that some values differ from those specified on the datasheet, and in some cases, the difference is quite significant. One plausible explanation is that a simple, unshielded wire connects the oscilloscope to the board for testing, which makes measurements susceptible to noise. Moreover, an additional time must be taken into account that makes this method of measurement less accurate. This delay time is the time it takes for the pin to go high and then low. Another drawback of this approach is that it permits only a limited number of measurements. In fact, the data presented in the table represents an average of 20 readings. Anyway, it is evident that the EVADC configuration with the shortest conversion times is that of test 12, while the longest is the one in test 1.

5.3 Results of the application 2

As for the outcomes of Application 2, the objective is to once again transform an analog signal using EVADC's single group and single channel, with a particular focus on the conversion times computed this time with Method 2.

Table 5.3: Conversion Timing Considering Internal Measurement – Method 2

Test	Analog Frequency (MHz)	Sample time (ns)	Noise Reduction Level	Post-calibration	Datasheet Value (ns)	Mean Value (ns)
1	20.0	100	3	Yes	1656.25	2637.5
2	20.0	100	0	Yes	1000	1106.2
3	20.0	100	0	No	768.75	775
4	26.7	100	3	Yes	1306.25	2025
5	26.7	100	0	Yes	800	843.8
6	26.7	100	0	No	618.75	587.5
7	40.0	100	3	Yes	931.25	1462.5
8	40.0	100	0	Yes	575	631.3
9	40.0	100	0	No	443.75	450
10	53.3	100	3	Yes	800	1162.5
11	53.3	100	0	Yes	518.75	506.2
12	53.3	100	0	No	375	362.5

Tests were performed with a constant sample time, varied post-calibration, analog frequency, and noise reduction levels. The results are more precise compared to application 1 as it eliminates the additional noise component caused by the wire connections between the board and oscilloscope. The results were printed out in a text file and then averaged over 2000 samples, suggesting the potential for taking more measurements than method 1. We observe that this also affirms that test 12 has the shortest conversion time, whereas test 1 has the longest conversion time.

5.4 Result of the final application

After setting all groups and channels for EVADC, it is verified whether the application is working correct by interrupting the execution of the application and checking the registers where the results are written from EVADC_G1RES0 to EVADC_G7RES0. From the following tables it is possible to see that the application works, in particular this can be seen from the fact that the RESULT field is different from 0.

Table 5.4: EVADC register G1RES0 – Final Application

EVADC_G1RES0	0x800007ad	
RESULT	0x7ad	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	1 – No new result available	31: Valid Flag

Table 5.5: EVADC register G2RES0 – Final Application

EVADC_G2RES0	0x7ae	
RESULT	0x7ae	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	0 – No new result available	31: Valid Flag

Table 5.6: EVADC register G3RES0 – Final Application

EVADC_G3RES0	0x77d	
RESULT	0x77d	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	0 – No new result available	31: Valid Flag

Table 5.7: EVADC register G4RES0 – Final Application

EVADC_G4RES0	0x7ab	
RESULT	0x7ab	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	0 – No new result available	31: Valid Flag

Table 5.8: EVADC register G5RES0 – Final Application

EVADC_G5RES0	0x7ab	
RESULT	0x7ab	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	0 – No new result available	31: Valid Flag

Table 5.9: EVADC register G6RES0 – Final Application

EVADC_G6RES0	0x7a9	
RESULT	0x7a9	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	0 – No new result available	31: Valid Flag

Table 5.10: EVADC register G7RES0 – Final Application

EVADC_G7RES0	0x77e	
RESULT	0x77e	0-14: Result of Most Recent Conversion
DRC	0x0	16-18: Data Reduction Counter
CHNR	0x0	20-23: Channel Number
EMUX	0x0	25-26: External Multiplexer Setting
CRS	0 – Request source 0	28-28: Converted Request Source
VF	0 – No new result available	31: Valid Flag

To ensure the functionality of the application, a signal ramp was produced using a signal generator with a steady amplitude (2 Vpp) at various frequencies. The ramp was then conveyed to the board via AN0 pin. Subsequently, 2000 samples of the signal were acquired and the sampling time was calculated.
During the experiment, special attention was paid to verifying the conversion times, comparing them with the calculations performed using method 2 in test 3. The importance of this verification lay in the need to ensure that the EVADC groups and channels operated independently of each other without interference.
After confirming that the conversion times aligned with predictions, a signal reconstruction quality check was conducted. This task was completed by generating a graphical representation of the 2000 samples using MATLAB. The outcome was favourable, as the waveform was successfully reconstructed at various frequencies. This can be seen from figures 5.1, 5.2, 5.3, 5.4, where on the x-axis it is possible to see the quantisation levels, which in this case range from 800 to 2600. These values resulted from the 12-bit converter, which offers 4096 levels. Considering that the board operates at 5V, 1V is represented by 819.2 levels. For the current situation, the

wave has an amplitude of 2Vpp and ranges from 1V to 3V. Hence, the reconstructed wave will range from approximately 820 to 2450 levels.
It should be noted, however, that there is a slight noise in the acquired signal. This noise is unavoidable as the noise reduction level has been set to 0. However, it is important to point out that there are solutions to deal with this problem, such as the adoption of other noise reduction techniques, which can be implemented to improve the quality of acquisitions without compromising conversion time.

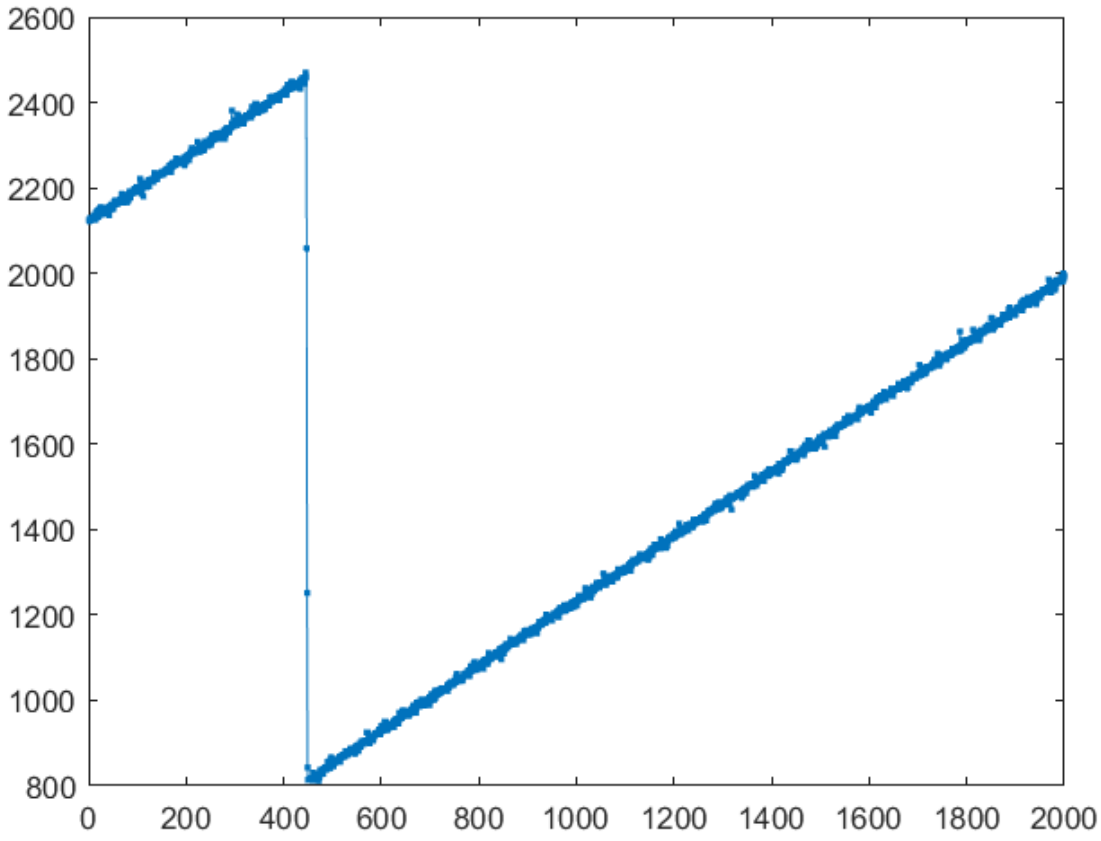

Figure 5.1: Ramp, 1 Vpp, 1kHz.

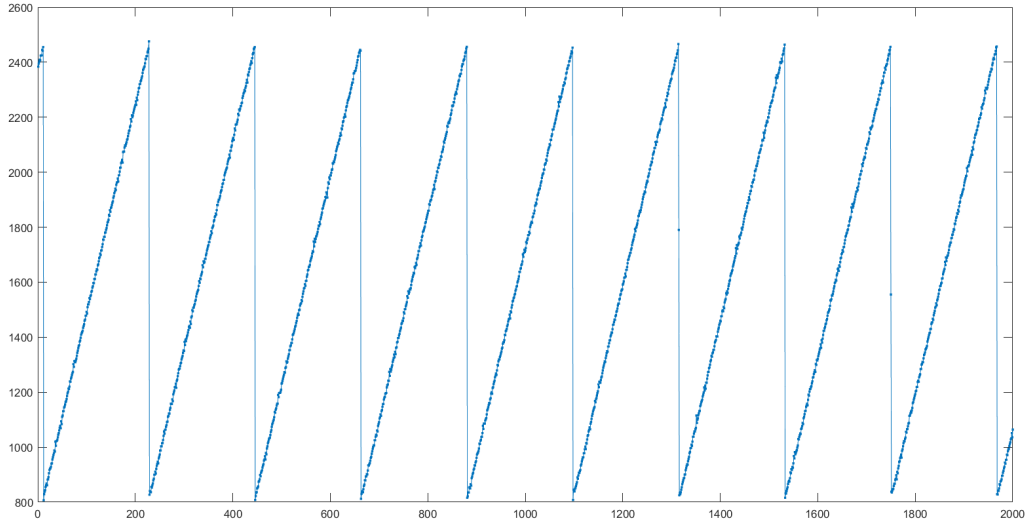

Figure 5.2: Ramp, 1 Vpp, 10kHz.

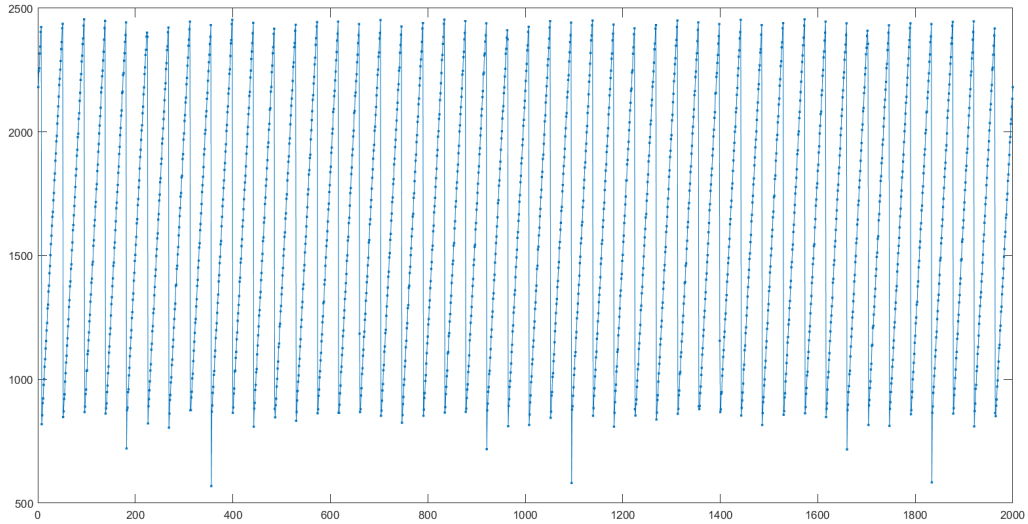

Figure 5.3: Ramp, 1 Vpp, 50kHz.

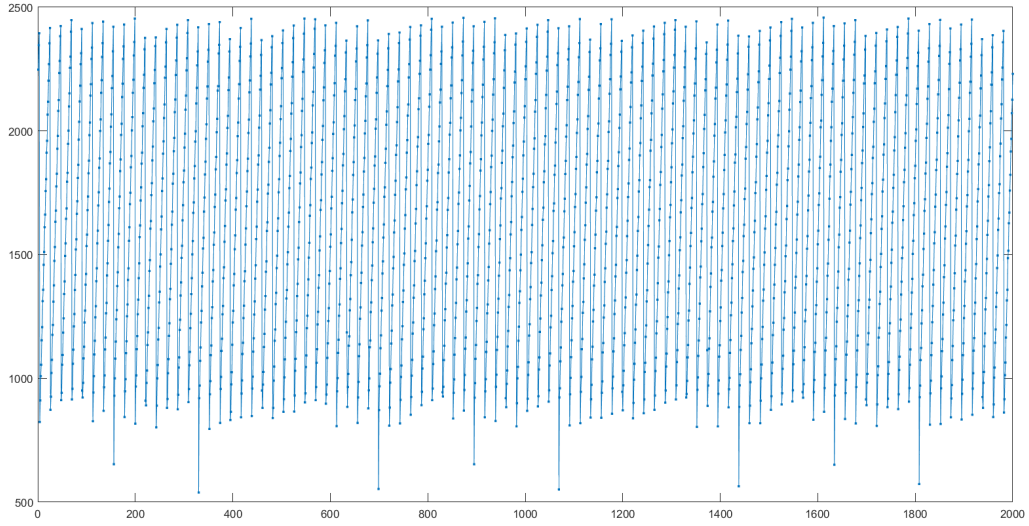

Figure 5.4: Ramp, 1 Vpp, 100kHz.

Solutions for noise reduction

In order to discuss this topic, the Infineon User Manual [6] is used, in particular the section dedicated to noise reduction techniques.
Analog input signals typically contain some noise. Noise may be generated within the sensor or along the signal path to the converter input. Digital blocks of the microcontroller may also create noise that could affect the converter operation. To mitigate this, the EVADC offers several techniques for noise attenuation.

Noise-Reduction Levels

Additional refining steps (1,3,7) can be selected to enhance the accuracy of the generated result value. The amount of noise reduction applied influences the timing of the conversion process.

Spread Early Sample Point

Unlike statistical noise, synchronous noise cannot be eliminated by oversampling. The Spread Early Sample Point feature reduces synchronous noise by shifting the end of the sample phase in pseudo-random steps. When this feature is enabled, the

sample phase is randomly shortened by up to 100 ns. However, this also affects the conversion timing.

Accumulated Conversions

The accumulation of 2 to 16 conversion results takes place to support the calculation of averages. The system automatically accumulates a set number of values, and service requests are generated only after the accumulation process is complete.

Mutual Interference of Converters

Since all converters utilize the common reference voltage V_{AREF} (except for the alternate reference), they can interfere with each other. Synchronising their operation can prevent this.

Chapter 6

Conclusion

The objective of this thesis was to identify the optimal analog-to-digital conversion module, capable of converting multiple quantities simultaneously and with maximum speed for ensuring real-time current and voltage control of an electric motor. The research concluded that the Enhanced Versatile Analog-to-Digital Converter (EVADC) is the most suitable module, integrated in the AURIX™ TC399 series microcontrollers. This study entailed a comprehensive assessment of the EVADC's performance in contrast to other options available in the market. The results clearly demonstrated that EVADC is a superior choice due to its combination of conversion speed and resolution.

Another important aspect was the synchronization of the Enhanced Versatile Analog-to-Digital Converter (EVADC) with the Generic Timer Module (GTM). Using Pulse-Width Modulation (PWM) signals generated by the Timer Output Modules (TOM) as triggers for the EVADC conversion processes resulted in an accurate and consistent solution.

The experiments and tests carried out during this research verified the correct functioning of the final developed application and the accuracy of the EVADC conversion times in accordance with predictions. Moreover, a verification of the signal reconstruction quality at various frequencies was conducted, emphasising that while the conversion speed was given priority, the signal reconstruction remained optimal.

6.1 Future Works

It is important to point out that, in addition to the results obtained in this research, there are interesting points that could pave the way for future advances.
One of these concern the synchronization of the PWM waves used to trigger the EVADC modules. Currently, these waves start simultaneously, but it could be advantageous to introduce a time shift between them. Particularly in regards to the waves that initiate the current conversions, as illustrated in the figure 6.1. This dead time interval would enable the electrical quantities to be sampled at the desired times, thereby allowing for optimal results.
Furthermore, it is important to note that the sensors cause a small delay in signal collection. To resolve this issue in future, sampling ahead or behind the vertices of the triangular waves acting as carriers could be worth exploring, as shown in the figure 6.2. A possible solution would be the use of the same timer module to generate both primary and secondary PWM waves. The primary waves would regulate the opening and closing of the switches of the DC/DC converter, while the secondary waves, that depend on the primary ones, would be merely used as triggers for the EVADC modules. This approach may provide an efficacious solution, but it is important to remain open to explore further alternatives, if necessary, to ensure optimal performance in real-time control of electrical systems.

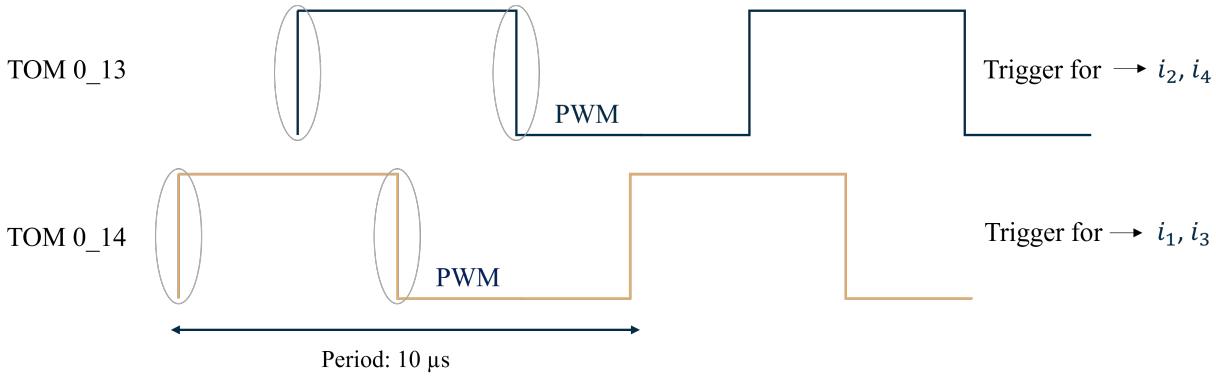

Figure 6.1: Shifted PWM

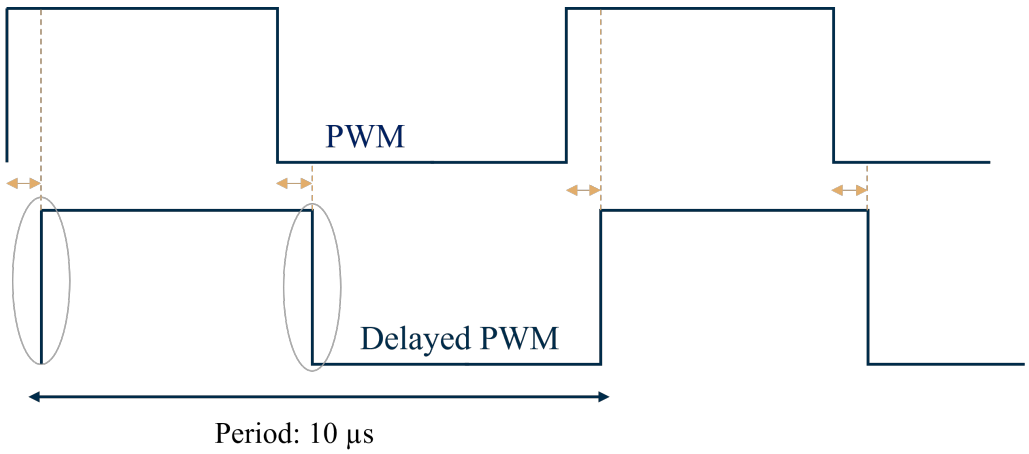

Figure 6.2: Delayed PWM

Bibliography

[1] *S32K1xx Series Reference Manual.* URL: https://usermanual.wiki/Document/S32K1xx20MCU20Family20Reference20ManualREV208.1780670565/html.

[2] *NXP Semiconductors Data Sheet.* URL: https://www.nxp.com/docs/en/data-sheet/S32K-DS.pdf.

[3] *STM32F405xx STM32F407xx Data Sheet.* URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f405rg.html.

[4] *RM0090 Reference manual.* URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f405-415/documentation.html.

[5] *Infineon TC39x Datasheet.* URL: https://www.infineon.com/dgdl/Infineon-TC39x-DataSheet-v01_02-EN.pdf?fileId=5546d462712ef9b7017140bc3416145f.

[6] *User Manual Part2 v02_00.* URL: https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_Part2-v02_00-EN.pdf?fileId=5546d462712ef9b701717d35f8541d94.

[7] T Likitha, Venkatanarasimharao Medam and G Ranjani. "ADC-PWM Synchronization for Motor Control Unit Development". In: *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*. 2022, pp. 1–6. DOI: 10.1109/GCAT55367.2022.9972020.

[8] Bowen Wang et al. "Research on two-dimensional electric field measurement system for electrical tree detection in insulating materials". In: *2022 IEEE International Conference on High Voltage Engineering and Applications (ICHVE)*. 2022, pp. 1–4. DOI: 10.1109/ICHVE53725.2022.9961620.

[9] *GTM-Cookbook: Overview and Application examples.* URL: https://www.bosch-semiconductors.com/media/ip_modules/pdf_2/gtm/gtm_cookbook_v05.pdf.

[10] Chen Li et al. “Analysis and Compensation of Sigma-Delta ADC Latency for High Performance Motor Control and Diagnosis”. In: *IEEE Transactions on Industry Applications* 59.1 (2023), pp. 873–885. DOI: 10.1109/TIA.2022.3210078.

[11] Simone Ceglia. “Sviluppo di una scheda elettronica digitale nell’ambito della sicurezza industriale= Development of a digital electronic board in the field of industrial safety”. PhD thesis. Politecnico di Torino, 2023.

[12] Zhe Zhang et al. “Optimized digital implementation of carrier-based randomized discontinuous PWM technique for active front end (AFE) drives”. In: *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE. 2019, pp. 4390–4394.

[13] *User Manual v01*. URL: https://www.infineon.com/dgdl/Infineon-AURIX_TC39x-UserManual-v02_00-EN.pdf?fileId=5546d462712ef9b7017182d371dc1d95.