POLITECNICO DI TORINO

Politecnico
di Torino

Master's Degree in Electronic Engineering

Expansion board for digital out channels in HIL simulator controlled by CAN protocol

Relator:

Prof. Guido MASERA

Candidate:

Davide MILONE

Kineton Supervisor:

M.Sc. Giuseppe SELVA

2022 - 2023

Abstract

Nowadays, Hardware In The Loop (HIL) is the most used testing way for embedded ECUs on board the vehicles. Thanks to its versatility and reliability it is possible to make ECU tests as close as possible to reality.
The idea behind HIL is having a configurable system by means software interface in order to provide all of electrical stimuli needed to fully exercise the ECU. In this way the simulator is able to replicate electrical signals as close as possible to signal from real vehicle sensors and reading the ECU responses, ensuring the possibility to simulate communication protocols like CAN, LIN and FlexRay. HIL simulators can communicate with ECUs in different way depending on the ECU architecture. In general, part of them have available different boards that include I/O ports.
Generally, depending on the project or customer require, simulators are hardware customized for that specific purpose. Nevertheless, at the end of it or in case of customer change they can be reused but it is probable that by adapting the simulator to the new project, the number of channels is not enough. One of the possible solutions can be to add new boards (although there is physical space) on simulator, or at worst, buy a new customized one. In both cases this would require an high economic effort. The goal of this dissertation is to design a board that emulates some of the functionality of the real one of the simulator that is reliable and cheap. The key idea is to design an expansion board that communicates with simulator by CAN in one side and connected to the ECU to the other side. Since each of them can be connected by means different channels, the aim of this dissertation developed in Kineton, an engineering consultancy company based in Turin, will be focused only on those with digital output. The challenge is to design and develop the hardware needed to achieve the high-side, low-side and push-pull configurations at the performances required by a normal simulator that are a range frequency 0-100KHz, variable duty cycle 0-100% and range voltage 0-30V.

RINGRAZIAMENTI

È mia volontà ringraziare tutte le persone che mi hanno seguito e supportato durante tutto questo percorso. Un ringraziamento va alla società Kineton S.r.l. per avermi permesso lo svolgimento di questo lavoro di tesi, in particolar modo a Giuseppe Selva e a tutti i colleghi che mi hanno sempre fatto sentire a casa. Al mio relatore, professor Guido Masera per aver accettato di appoggiarmi in questo progetto fin dall'inizio e a tutti i docenti che sono stati per me fonte importante di informazione.
I miei piú sentiti ringraziamenti vanno alla mia famiglia, a mia madre e mio padre. Non esistono parole per descrivere quanto sia stata fondamentale per me la loro presenza nel raggiungimento di questo traguardo cosí tortuoso e faticoso. Grazie a voi se oggi posso dire di aver completato un importante capitolo della mia vita. Grazie per non avermi fatto mai mancare nulla.

A mia madre, che è stata linfa vitale durante tutto il percorso. Grazie ai tuoi insegnamenti sono riuscito ad affrontare e superare sempre con successo tutti i momenti più buii. Grazie per essere stata l'unica a non avermi MAI realmente abbandonato. A te devo tutto e non finiró mai di ringraziarti abbastanza.

E' mio desiderio, inoltre, ringraziare mia nonna Adriana per essersi sempre emozionata insieme a me ad ogni risultato d'esame. Grazie per esserti costantemente preoccupata di me e per averti sentito sempre vicina. Grazie per aver speso sempre belle parole di conforto nei momenti di bisogno. Grazie a Matteo, mio cugino, mio fratello, che seppur fisicamente distante é sempre stato presente attraverso semplici chiamate. Grazie a lui per essersi sempre preoccupato di tutto e per essersi messo sempre a disposizione. Grazie inoltre a tutto il resto della mia famiglia che seppur non direttamente si è sempre preoccupata di me.

A tutti i miei amici, infine, va la mia più sincera gratitudine; in particolar modo a Simone che purtroppo non ha potuto condividere con me questo traguardo ma che è stato il mio esempio di resilienza principale. A Giulio, Matteo e Gianluca per esserci sempre stati. Ai miei amici napoletani, seconda famiglia torinese, va un sincero ringraziamento per avermi regalato sempre momenti di felicità e spensieratezza. Spero di continuare a condividere con voi questi momenti anche in futuro.

A tutti voi GRAZIE!

ACKNOWLEDGEMENTS

It is my will to thank all the people who have followed and supported me throughout this journey. A thank you goes to the company Kineton S.r.l. for allowing me to carry out this thesis work, especially to Giuseppe Selva and all the colleagues who have always made me feel at home. Al Guido Masera for agreeing to support me in this project from the beginning and to all the teachers who have been for me important source of information.
My heartfelt thanks go to my family, my mother and my father. There are no words to describe how fundamental their presence in the achievement of this so tortuous and tiring goal. Thanks to you if today I can say that I have completed an important chapter of my life. Thanks for never missing anything.

To my mother, who has been the lifeblood all along. Thanks to your teachings I was able to face and always successfully overcome all the darker moments. Thank you for being the only one who NEVER really had me abandoned. I owe you everything and I will never stop thanking you enough.

It is my wish, moreover, to thank my grandmother Adriana for always getting excited with me to every test result. Thank you for constantly worried about me and for having always felt close to you. Thanks for always spending words of comfort in times of need. Thanks to Matthew, my cousin, my brother, who although physically distant has always been present through simple calls. Thanks to him for always caring about everything and for putting always available. Thanks also to all the rest of my family that although She didn't directly always worry about me.

Finally, to all my friends goes my most sincere gratitude; especially to Simone who unfortunately could not share with me this goal but that is was my main example of resilience. To Giulio, Matteo and Gianluca for being there always been. To my Neapolitan friends, Turin's second family, goes a sincere thanks for giving me always moments of happiness and carefree. I hope to continue to share these moments with you in the future.

To all of you THANK YOU!

Contents

Chapter 1

Introduction to ECUs

ECU stands for Engine Control Unit, which is the heart of vehicles. It is responsible for monitoring and controlling various systems within the engine to ensure optimal performance, efficiency, and safety. The ECU receives input from various sensors, such as the throttle position sensor, oxygen sensor, and coolant temperature sensor, and uses this information to adjust various parameters, such as fuel injection timing, ignition timing, and air/fuel mixture. The ECU also controls other systems, such as the transmission, ABS, and traction control. It is evident how nowadays the ECU's complexity is constantly growing in modern vehicles since the high number of sensors placed in a vehicle in order to have high performance and safety. In addition to monitoring and controlling engine and other systems, the ECU also stores diagnostic trouble codes (DTCs) which can be read using specialized diagnostic tools. These codes can help technicians identify and diagnose issues within the engine and other systems. Overall, the ECU plays a critical role in ensuring the smooth and efficient operation of modern automotive systems, and is an important component in ensuring the safety and reliability of vehicles.

1.1 Communication protocols

In order to make the ECU in communication with all sensors, actuator and more in general with all peripheral devices, a common communication protocol is needed. These protocols are essential for ensuring that different parts of the vehicle can exchange information effectively and operate together efficiently. Since a lot of data of different types is exchanged within a vehicle, different protocols exist. The main analyzed protocol are LIN and CAN.

1.1.1 Introduction to LIN protocol

LIN (Local Interconnect Network) is a serial protocol network used for communication between ECU and actuators and sensors that have to transmit or receive few information at low frequency. LIN is a broadcasting network that includes one master and up to 16 slaves. In this way after master send the message, only one slave per time that is selected by the ID message, can reply to the master itself. Below, some application of it:

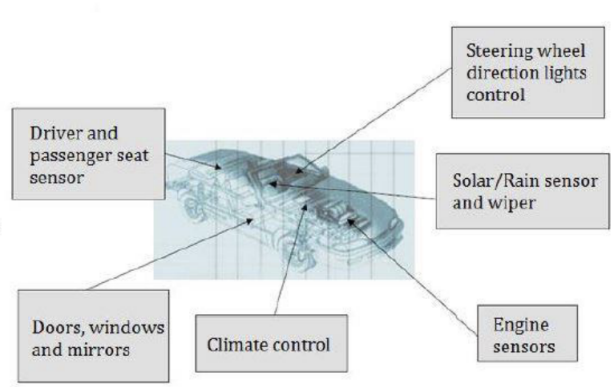


Figure 1.1: LIN Network applications

1.1.2 CAN protocol

CAN stands for Controller Area Network and is a serial communication bus designed for real-time applications. The first usage was in automotive field but later was also used for industrial applications. The idea behind was to make all the internal devices of a vehicle in communication with each other, reducing the number of interconnection wires and external disturb on it. About external disturbs, CAN interconnects a network of modules (or nodes) using two wire, twisted pair cable. The protocol is a serial, multimaster, multicast, which means that when the bus is free, any node can send a message (multimaster), and all nodes may receive and act on the message (multicast). In terms of communication speed, a bit rate up to 1 Mbit/s is possible in short networks lower than 40m. Anyway, "High speed" CAN is considered to be 500 kbit/s. The two twisted pair wires are classified as CAN_H and CAN_L and they carry the same information in the differential way.

The result is given by the difference between these two signals that provide 0V or 2V. The main advantage of that is the immunity to disturbs which led this kind of network to be used in noisy environments.

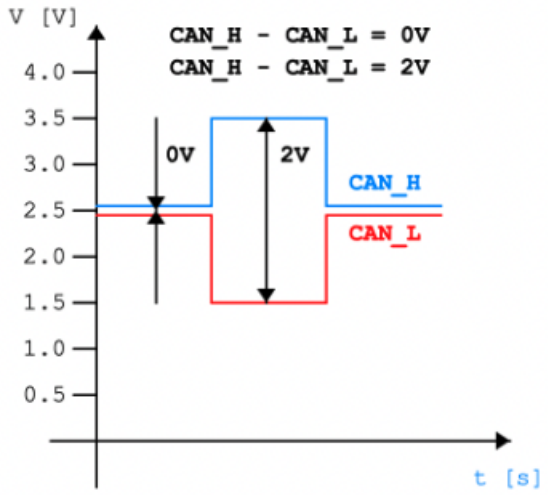


Figure 1.2: CAN_H and CAN_L amplitude

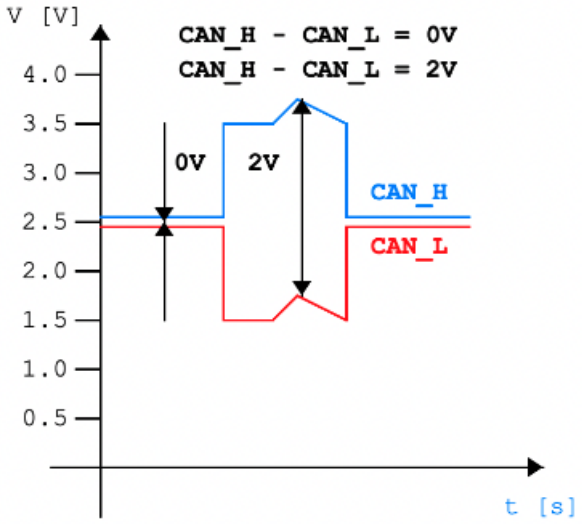


Figure 1.3: CAN_H and CAN_L disturbed

For the node to read the bus level correctly it is important that signal reflections are avoided. This is done by terminating the bus line with a termination resistor at both ends of the bus and by avoiding unnecessarily long stubs lines of the bus. The method of terminating your CAN hardware varies depending on the physical layer of your hardware: High-Speed, Low-Speed, Single- Wire, or Software-Selectable. For High-Speed CAN, both ends of the pair of signal wires (CAN H and CAN L)

must be terminated. The termination resistors on a cable should match the nominal impedance of the cable. ISO 11898 requires a cable with a nominal impedance of 120 ohms, and therefore 120 ohm resistors should be used for termination. For Low-Speed CAN, each device on the network needs a termination resistor for each data line: R(RTH) for CAN H and R(RTL) for CAN L.

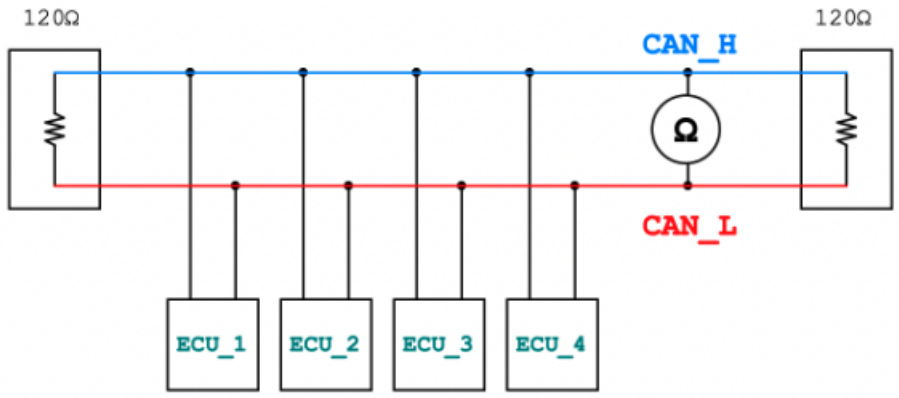


Figure 1.4: CAN Bus structure

About the protocol CAN specifies two logical states: recessive and dominant. According to ISO-11898-2, a differential voltage is used to represent recessive and dominant states (or bits). In the recessive state (usually logic 1), the differential voltage on CAN H and CAN L is less than the minimum threshold (<0.5V receiver input or <1.5V transmitter output). In the dominant state (logic 0), the differential voltage on CAN H and CAN L is greater than the minimum threshold.

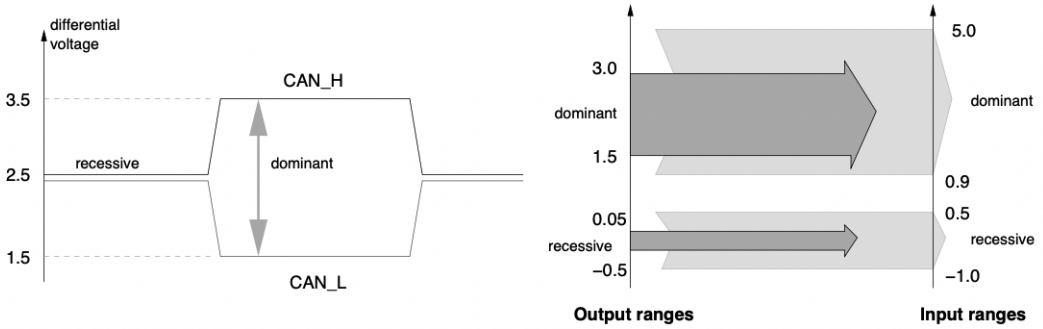


Figure 1.5: Encoding of dominant and recessive bits

The CAN Frame can be of four different kind:

• Data frame: are used to transmit information between a source node and one of more receivers.
• Remote frame: is used to request the transmission of a message with a given identifier from a remote node.
• Error frame: used for error transmission.
• Overload frame: if a CAN node receives messages faster than it can process them, then an Overload Frame will be generated to provide extra time between successive Data or Remote frames.
The Data Frame is organized as below:

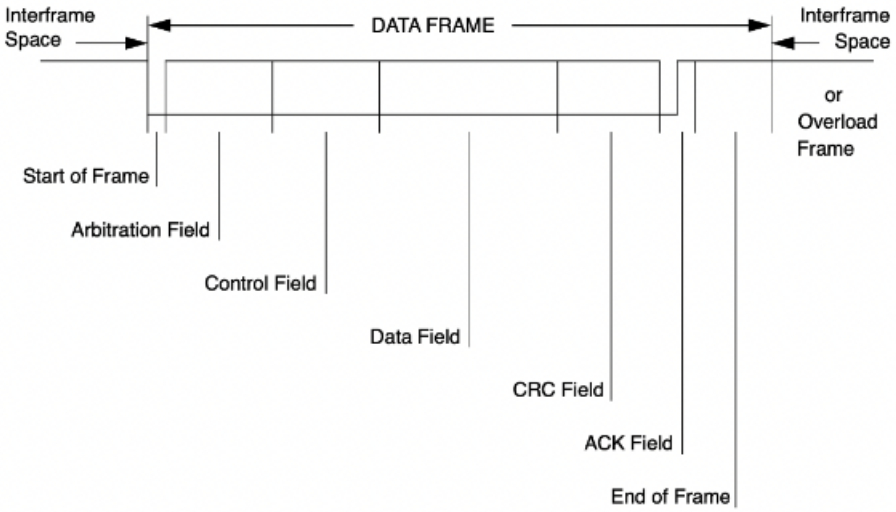


Figure 1.6: Data frame structure

The CAN data frame is composed of seven fields: Start of frame (SOF), arbitration, control, data, cyclical redundancy check (CRC), acknowledge (ACK) and end of frame (EOF). CAN message bits are referred to as “dominant” (0) or “recessive” (1). The SOF field consists of one dominant bit. All network nodes waiting to transmit synchronize with the SOF and begin transmitting at the same time.

Chapter 2

Introduction to HIL system

Hardware in the loop (HIL), was developed for two main reasons: time to market and complexity. Nowadays, thanks to that complex systems can be developed and tested in a fast and safe way by means custom plant configuration. In other words, the key idea is reading and providing to device under test, ECU in our case, all electrical stimuli needed for its function. The benefit is that testing can be done without damaging equipment or endangering lives. In the other hand, ECUs that have not yet been physically developed can also be simulated importing the simulink model on it. The main used software for the configuration of the HIL is called Configuration Desk that translates the inputs/outputs of the model that has to be provided by means Simulink Tool on the internal real hardware. The flexibility of the simulator allows the user to have also a practical feedback on the actuators by means the so called load plate. Depending on the specific project

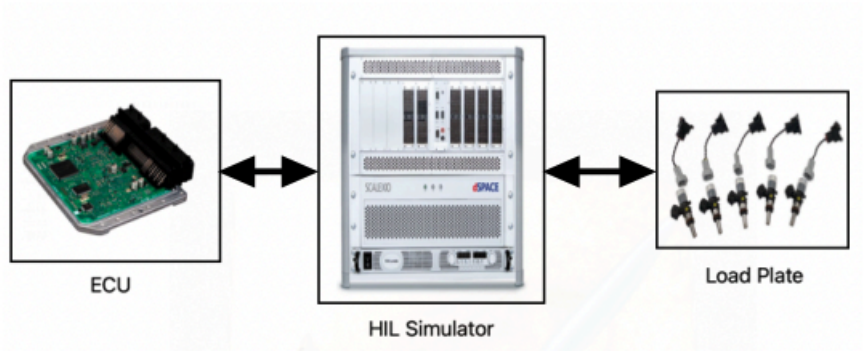


Figure 2.1: Block diagram of the test system architecture

Load Plate is not always present. Let's consider a generic electric engine. In that case there will not be the real one but it will be simulated by model.

2.1 DSpace Simulators

DSPACE provides devices and tools for developing, testing and calibrating electronic control units (ECUs) in the automotive, aerospace and medical engineering industries. In this section you will find an overview of the architecture of simulators in order to better understand how they interface with the real world. There are different types of simulators that differ from the point of view of hardware (single core RTP, multi-core RTP, different number of FPGAs, different kind and number of I/O boards).
Just to have a clear view of the most used simulator, below an example.

Figure 2.2: HIL simulator

Generally, all simulators interface with external environment through different signal types by means different slots unit or boards. Below the front view of a SCALEXIO system that includes a DS2703 6-Slot Unit and DS2680 I/O Unit.

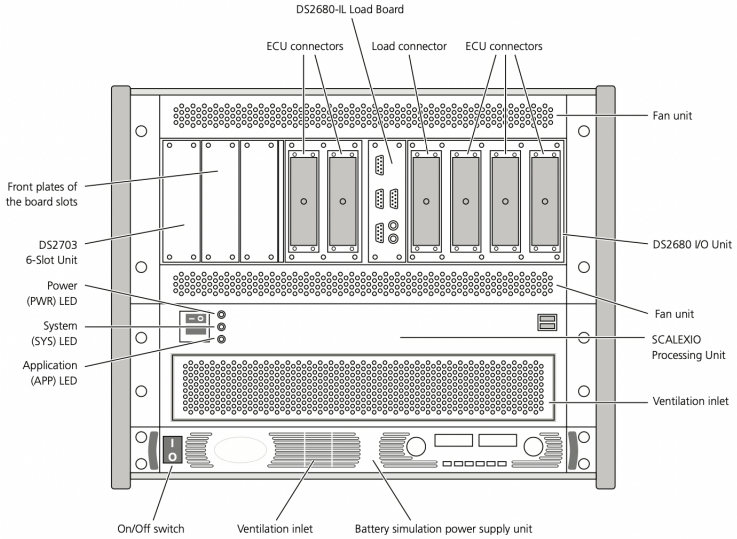


Figure 2.3: Front view of Scalexio system

Most of them are analogue, digital, resistive, power and communication. Power channels supply the ECU connected to the HIL simulator and represent the vehicle's battery. Resistive channels provide resistor values and, usually, simulates temperature sensors. Analog channels provide or acquire voltage signals between 0 V and battery voltage. The value can be whatever is inside this range. Digital channels provide or acquire voltage in the same range as the analogue channels, but assuming only 0 V or battery voltage. Moreover, digital channels can provide or acquire PWM signals (Pulse Width Modulation). This types work in the same range of digital signals and are periodic, so they are also frequency and duty cycle depending. Communication channels use typical vehicle protocols such as CAN and LIN.
Since the CAN or LIN message have to be simulated, simulators relies on DBC. They are files that include all messages in terms of bits that each control unit (node) receives and transmits. In the case of a transmitted message, the simulator will read that message from the DBC.

The main reference slot for this thesis project are the I/O boards which are part of simulators as mentioned above. In particular the specific DS2680 board.

Figure 2.4: DS2680 board

2.1.1 DS2680 overview

The DS2680 board is the most used in projects since it offers a large number of channels. Most of them are digital I/O, analog I/O, Flexible IN and Resistance OUT. Since the purpose of the thesis is focused on digital channels, the digital out architecture has been analyzed.

2.1.2 Digital out channel architecture

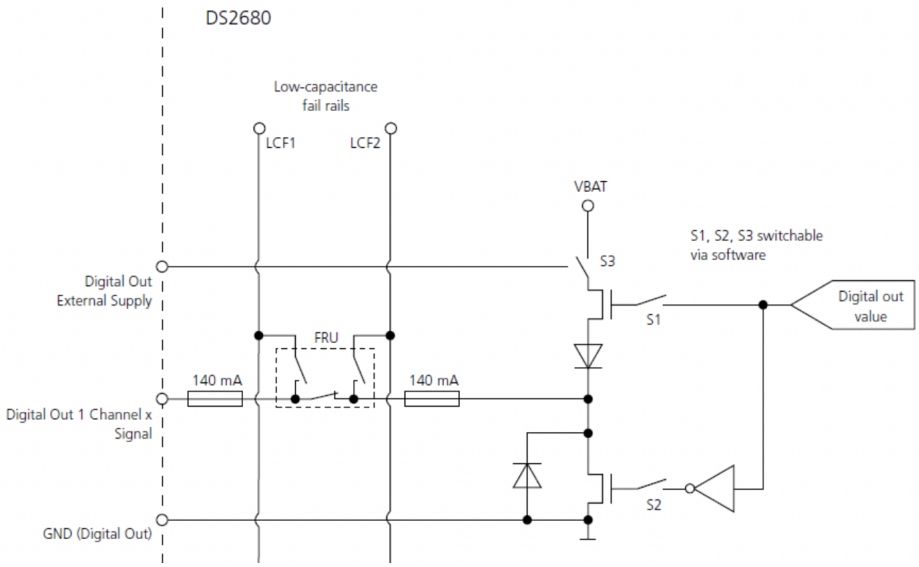


Figure 2.5: Digital OUT channel scheme of DS2680 board

The figure above, is the digital out channel scheme that represents the basic idea for our project. Of course, it is only a general scheme that shows as different configurations can be achieved by switching S1, S2 and S3 switches. All of them

are switched by software.
If only S1 is closed, the circuit works as a High-Side driver; on the contrary, if only S2 is closed, it works as a Low-Side driver. Finally, if both are closed, it works as a Push-Pull. S3 chooses from which source to take the voltage. VBAT means the internal simulator power supply or External Supply. The digital out value is what the simulator wants to send to ECU while the FRU part is used for faults generation. This last part was not considered in this thesis project.

Chapter 3

Digital output configuration solutions

Just to have an idea on where the board will be placed in the reality and from where the input signal comes, the figure 3.1 can be taken as reference.

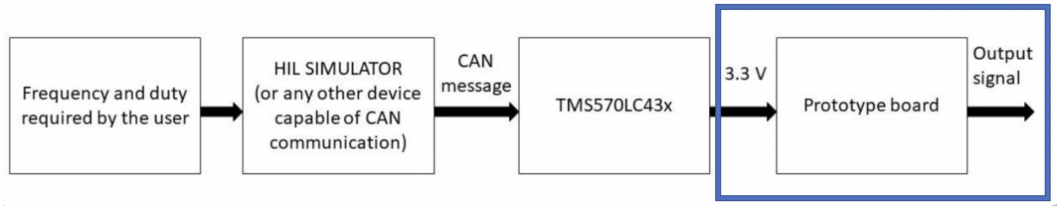


Figure 3.1: Configuration block diagram

The board communicates directly with ECU in one side and with simulator by means a microcontroller to the other. The latter, that has been previously programmed by another member of the company, receives a CAN message from the simulator and translates it into a PWM signal with a certain frequency and duty cycle according to DBC.
Since communication protocols such as CAN are simulated by the simulators, it is necessary to know how. The external user sends the message to be simulated to the simulator via so-called DBCs.
All the information relating to the messages that the simulator must transmit to the real ECU connected to it, and receive from itself, are contained in these so-called files which are databases. Thanks to these, the user can compose the message and its relative signal in terms of bits.

Name	Message	Multiplexing/...	Startbit	Leng...	Byte Order	Value Type	Initial...	Factor	Offset	Mini...	Maxi...	Unit
Board_A_Status_Signal	Board_A_Status	-	7	1	Motorola	Unsigned	0	1	0	0	1	
Enable_Short_Status	Potential_Selection...	-	0	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH10_sig_A_Sts	Signal_Faults_A_Sta...	-	9	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH11_sig_A_Sts	Signal_Faults_A_Sta...	-	10	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH12_sig_A_Sts	Signal_Faults_A_Sta...	-	11	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH13_sig_A_Sts	Signal_Faults_A_Sta...	-	12	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH14_sig_A_Sts	Signal_Faults_A_Sta...	-	13	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH15_sig_A_Sts	Signal_Faults_A_Sta...	-	14	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH16_sig_A_Sts	Signal_Faults_A_Sta...	-	32	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH17_sig_A_Sts	Signal_Faults_A_Sta...	-	33	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH18_sig_A_Sts	Signal_Faults_A_Sta...	-	34	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH19_sig_A_Sts	Signal_Faults_A_Sta...	-	35	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH1_sig_A_Sts	Signal_Faults_A_Sta...	-	0	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH20_sig_A_Sts	Signal_Faults_A_Sta...	-	36	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH21_sig_A_Sts	Signal_Faults_A_Sta...	-	37	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH22_sig_A_Sts	Signal_Faults_A_Sta...	-	38	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH23_sig_A_Sts	Signal_Faults_A_Sta...	-	39	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH24_sig_A_Sts	Signal_Faults_A_Sta...	-	40	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH25_sig_A_Sts	Signal_Faults_A_Sta...	-	41	1	Motorola	Unsigned	0	1	0	0	0	
OpenLoad_CH26_sig_A_Sts	Signal_Faults_A_Sta...	-	42	1	Motorola	Unsigned	0	1	0	0	0	

Figure 3.2: DBC Example

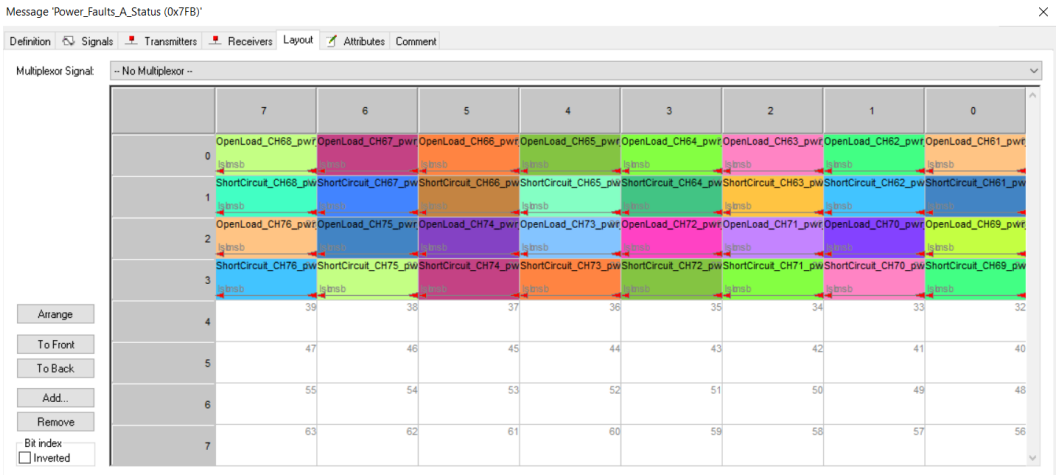


Figure 3.3: DBC Layout Example

In this project, the previously written DBC has been configured to communicate only frequency and duty cycle information. In this manner, the simulator provides frequency and duty cycle information to the microcontroller through CAN, which converts it into a PWM signal in the 0-3.3V range.
In this chapter three different solution will be presented. Each of them has been deeply analyzed. A detailed research was done on each of them to understand if it could be the solution to our case. The first two did not turn out to be so. The last one, "Gate driver with Bootstrapping architecture", the better one, was taken into consideration and carried forward in all phases of the project.

3.1 Separate configuration architecture

The key idea was having three different branches already configured in high side, low side and push pull and then was the microcontroller to decide which one to use by means a multiplexer and demultiplexer according to an external input from the user.

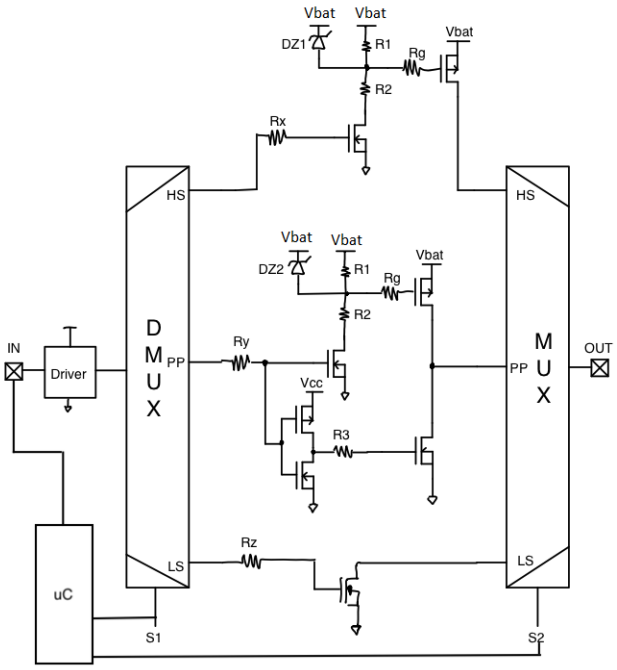


Figure 3.4: Separate configuration architecture

- **High Side Considerations**
  The initial idea was to use a PMOS to manage the High Side part controlled by a NMOS through the driver which would not be able to drive the PMOS gate directly to 30V in case of the output channel has to work at maximum voltage. In order for the PMOS to conduct, the condition on its threshold voltage must be satisfied. In this regard, it was decided to use a resistive divider with a reverse-biased 20V zener diode to guarantee a voltage between the gate and source during the switching time to do not exceed the maximum voltage supported by the PMOS itself (20V).
  Although at high frequencies the switching time of the PMOS with respect to the input signal began to be relevant, this configuration was acceptable. Certainly a good resistance values R1 and R2 is essential for a good operation.
- **Low Side Considerations**

No particular comments on this configuration except for a good choice of NMOS.

- **Push Pull Considerations**
  this was the most critical setup. First of all, the same High side configuration has been replicated in the upper part then an inverter (built with two MOSFETs) has been inserted to alternately drive the NMOS output. Unfortunately, due to different delays in the two branches, the switch of the high side and low side was not completely synchronized and this led to high short-circuit currents (switch currents) such as to lead to the destruction of the MOSFETs themselves. In order to compare the delay of the UP branch and DOWN branch in the Push Pull configuration, the reference circuit was analyzed through the Elmore delay model by dividing the two branches as shown in the figures below.

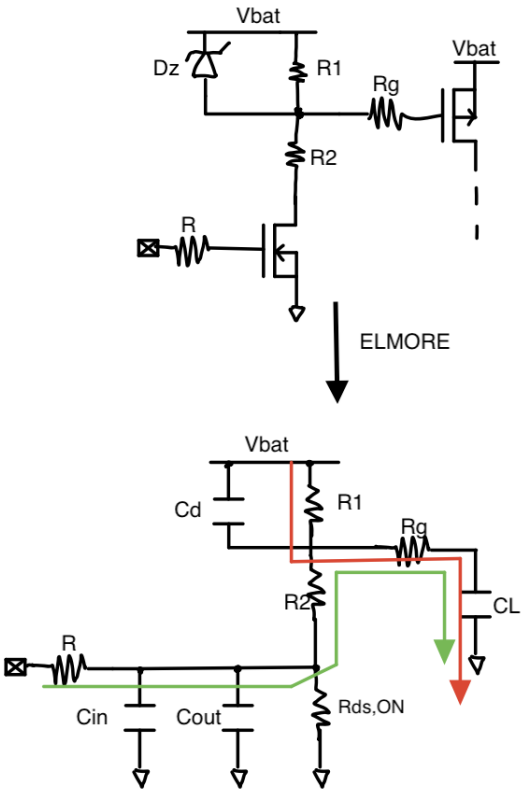


Figure 3.5: Elmore Delay branch UP

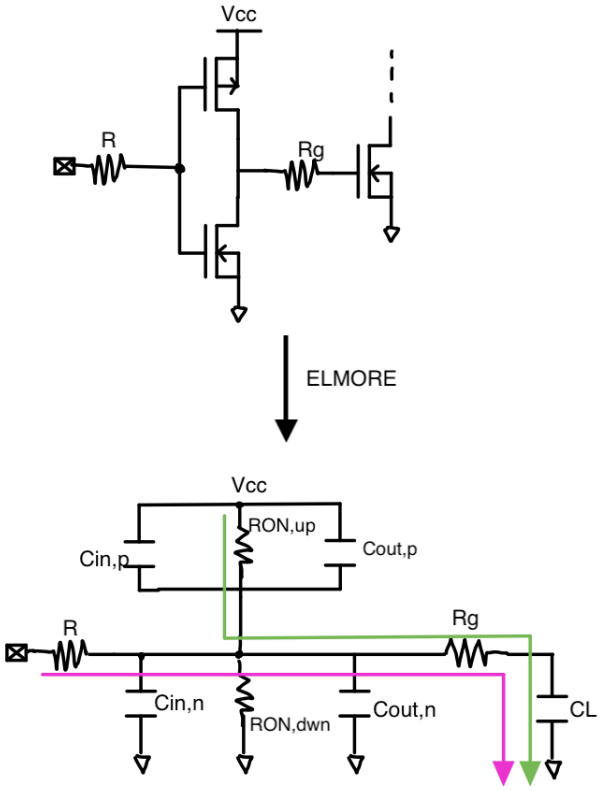


Figure 3.6: Elmore Delay branch DOWN

The two lines of different colors mark two different paths related to the path considered during the calculations to distinguish the two cases tpdON and tpdOFF subsequently calculated.
In order to better calculate the delays, the following MOSFETs were chosen as they complied with the specifications required by the project:

- PMOS SH8M41-P [6].
- Double NMOS NVMFD6H840NL [7].

From their respective datasheets, the following values needed to calculate the delay have been extrapolated:

- C_D = 7pF, C_IN = 2002pF, C_OUT = 249pF
- R = 10Ω, R1 = 100Ω, R2 = 512Ω, Rg = 10Ω, R_DS_ON = 6.9mΩ

The following equation used to calculate the two respective delay are reported below:

$$\tau_{pdON_{UP}} = \ln 2[R(C_{IN} + C_{OUT}) + (R + R_{DSON} + R_2 + R_G)C_L] \quad (3.1)$$

$$\tau_{pdOFF_{UP}} = \ln 2[(R_1 + R_G)(C_L + C_{OUT})] \quad (3.2)$$

$$\tau_{pdON_{down}} = \ln 2[R_{ON_{UP}}(C_{IN_p} + C_{OUT_p} + C_{OUT_n}) + (R_{ON_p} + R)C_L] \quad (3.3)$$

$$\tau_{pdOFF_{down}} = \ln 2[(RCINn + (R_{ON_{DWN}} + R)C_{OUT_n} + (2R + R_{ON})C_L)] \quad (3.4)$$

From the above equations, substituting the real values gives the following lags:

- $\tau_{pd}ON_{UP} = 384{,}4ns$, $\tau_{pd}ON_{DOWN} = 14{,}4ns$
- $\tau_{pd}OFF_{UP} = 137{,}4ns$, $\tau_{pd}OFF_{DOWN} = 32{,}8ns$

Comparing the results, it is evident the difference between the two cases in terms of the delay that led to the following drawbacks.

3.1.1 Drawbacks

- High Switch current with real components because of different delay.
  In a preliminary examination of the circuit, particularly in the Push Pull configuration, the signals that control the GATEs of the final stage take two different branches, resulting in a different delay. This led an issue related to the high switching currents between the MOS up and MOS down, which is justified by the fact that both MOSFETs are active for a short time period.
- Difficult management of the battery voltage at 30V for the High Side Branch.
  By using the zener diode to limitate the voltage drop on the high side Mosfet, at Vbat = 30V the high side would be always on.

3.1.2 Possible solutions

- Same delay of the signal that drives the UP and DOWN gate, minimizing the rise and fall time of the two pilot signals that turn on/off the PMOS and NMOS, reducing the time in which both are turned on.
- Change High Side configuration.

3.2 H Bridge architecture

This other solution uses the H-Bridge architecture which is still widely used in the automotive field to control motors. The idea was to adapt the integrated circuit LV8548MC [8].which has the configuration shown in figure 3.7 as close as possible to our case.

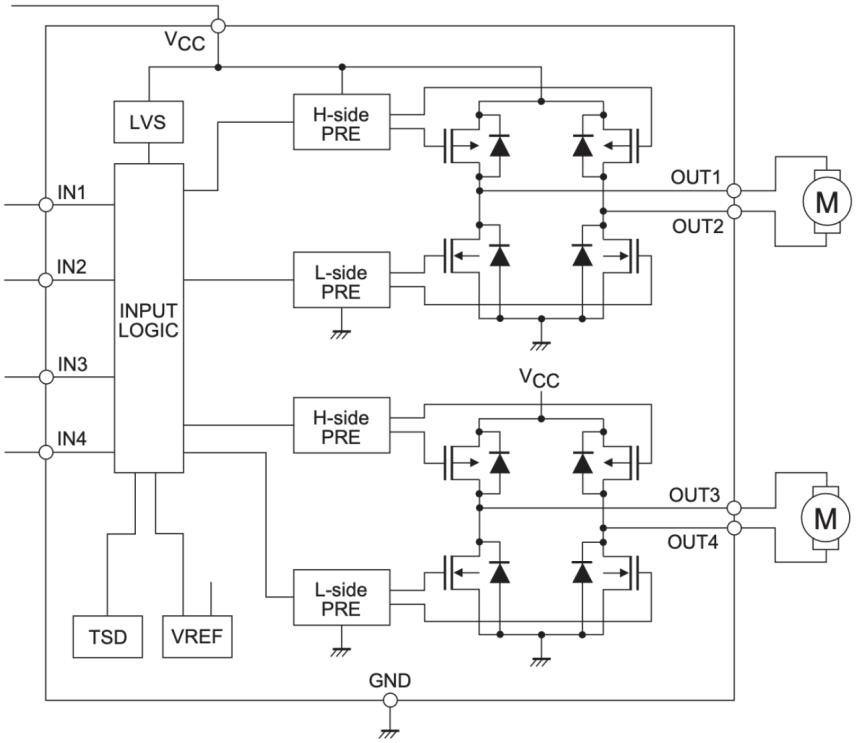


Figure 3.7: LV8548MC architecture

Input				Output				Remarks	
IN1	IN2	IN3	IN4	OUT1	OUT2	OUT3	OUT4		
L	L	L	L	OFF	OFF	OFF	OFF	Stand-by	
L	L			OFF	OFF			1CH	Stand-by
H	L			H	L				Forward
L	H			L	H				Reverse
H	H			L	L				Brake
		L	L			OFF	OFF	2CH	Stand-by
		H	L			H	L		Forward
		L	H			L	H		Reverse
		H	H			L	L		Brake

Figure 3.8: LV8548MC working table

Choosing only one of the two channels, the first activity was focused on understanding how to adapt the circuit to the real case in question. Observing the table, it shows how the outputs depend on the combination of the inputs. The idea was to control the input logic with the microcontroller and using only half of the bridge so using only OUT1.
Nevertheless, no analytical demonstrations have been made on the functionality but only an analysis and consideration on the table represented in figure 3.8. In this regard, in order to better adapt it to our case, the following considerations have been made for each case under examination:
- High Side
IN1 used for the input signal, IN2 must be grounded.
- Low Side
IN2 used for the input signal, IN1 must be grounded.
- Push Pull
IN1 used for the input signal, IN2 must be the inverse of IN1.

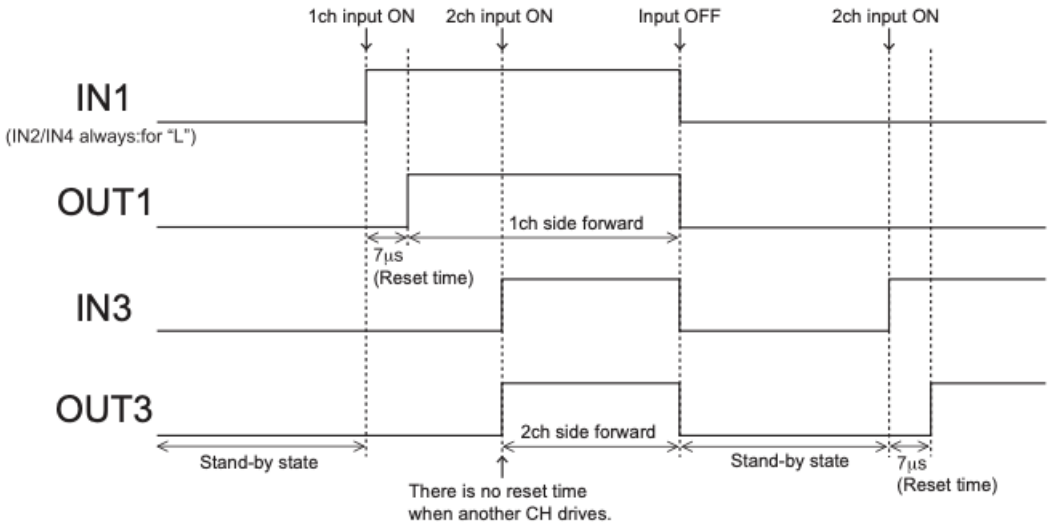


Figure 3.9: LV8548MC timing

3.2.1 Drawbacks

- Standby time.
  Analyzing the timing of the manufacturer shown in figure 3.9, a reset time equal to 7us is observed after the standby time. This means that in output you obtain the input signal after 7us. Since the input is a PWM signal, although it does not have a fixed duty cycle, every time the input signal goes to 0 it is necessary to wait for the reset time which leads to have an output signal with

different times compared to the input one.

- Output voltage limitated by the CMOS technology.
  Since the CMOS logic of the integrated circuit has a maximum output range of 20V, it is not possible to manage higher voltages as in our specific case (30V).

3.2.2 Possible solutions

Although additional full-bridge and half- bridge integrated circuits were analyzed, the main common issue experienced by most of the integrated devices explored on the market came out from the management of voltages higher than the standard 20V, such as 30V in our instance. For this reason, one of the possible challenge could be to designing a new custom Half-Bridge with robust mosfet and an appropriate driving logic.

3.3 Gate driver with Bootstrapping architecture

A different idea was exploiting another kind of working principle based on a bootstrap circuit around the main driver that includes a resistor, capacitor and diode. This is one of the most widely used methods to supply power to the high-side gate circuitry at low cost. Furthermore, the solution is simple and does not require many hardware components.

3.3.1 Working Principle

In order to be able to use an external voltage separate from the power supply voltage of the gate driver circuitry 15V which can be higher than the same as in our case 30V, the idea was to use the configuration shown in the figure below. Instead of using a PMOS as the High side, which would have been easily controllable if the maximum external voltage to be managed (Vbat) had been equal to the power supply voltage of the driver itself, the same NMOS was used as in the Low Side case. At this point, the external voltage applied to the NMOS drain may be higher but the problem remains on how to make the NMOS switch act as a switch. In this regard, the solution with bootstrapping helps us. The main purpose of the bootstrap circuit is to provide a positive bias voltage to the source of the high side gate so that it can be easily switched as a switch.
It is also important to highlight the importance of the gate driver which plays a fundamental role in driving the MOSFETs. In fact, it is able to provide higher output voltages and currents than the microcontroller alone.

The figure below represents the configuration used in this case and explains how it works.

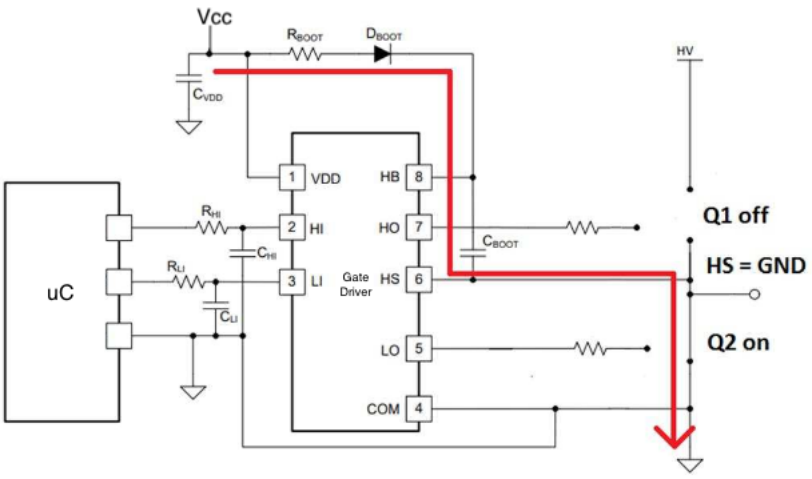


Figure 3.10: Bootstrap charging path

As soon as the low-side gate is switched ON, the source of the high-side gate is leaded to ground and in the capacitor will flow a current that will load itself at about the power supply voltage of the driver. There will be a voltage drop on the diode and bootstrap resistor.

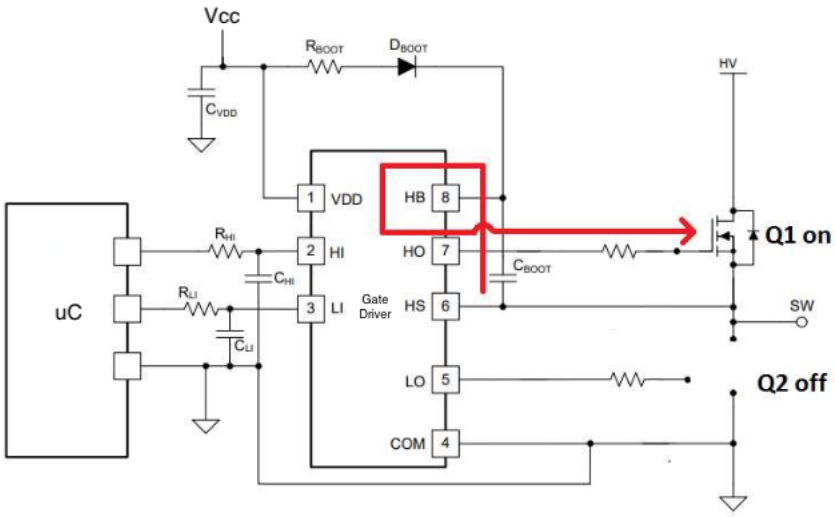


Figure 3.11: Bootstrap discharging path

At the moment when it is switched OFF and the high side gate ON, the source of the high side gate and the HS pin of the gate driver are pulled to the external high voltage bus (HV) and bootstrap capacitor discharges some of the stored voltage (accumulated during the charging sequence) to the high side gate through the HO and HS pins of the gate driver as shown in the figure above. In order to achieve three different configurations, the idea was to adapt this circuit separating each of them by means switches or as we will see in the PCB section by means jumper. The figure 3.16 shows better what has been said.
The operating principle mentioned so far is certainly correct if the channel is used in a push-pull configuration. Naturally, in order to be able to subsequently also verify High side and Low side operation, it is necessary to simulate the resistive load through a resistor following the configuration shown below.

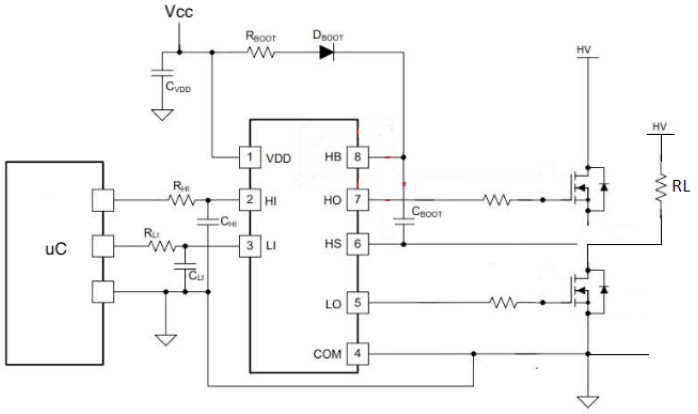


Figure 3.12: Low Side configuration

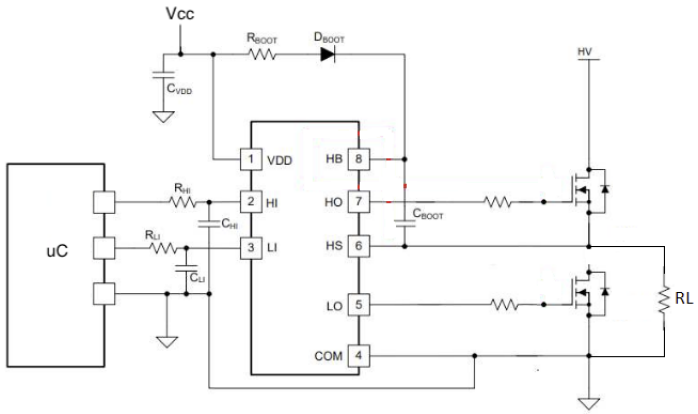


Figure 3.13: High Side configuration

Since no logic has been implemented to control the maximum current on the output load, the choice of resistor has been carried out following two conditions. The first, case of maximum voltage 30V, the second the smallest possible such as to be able to quickly discharge the high side capacitor. In this regard, considering the maximum voltage case, the maximum power required by the resistor itself was also evaluated. Therefore, by finding a trade-off between the maximum power that can be dissipated and the minimum value required, the resistance of 220 ohms [19] was found.

$$I = \frac{Vbat}{R} = \frac{30}{220} = 136mA \qquad P = R * I = 220\Omega * 136mA = 4W \tag{3.5}$$

So, a resistor of 220 ohm, 5W was considered for the simulations.

3.3.2 Bootstrap circuit selection

Cboot must be properly selected to ensure proper operation of the device. Choosing a capacitor too small can reault in a significant voltage drop when the charge is transferred to the gate of the high side MOS.
To limit this drop voltage is needed to force the charge Qboot of the bootstrap capacitor to be much larger than the total charge of the Qgate.

$$Q_{tot} = Q_g + QLS + ILK_{TOT} * (1 - D_{MIN}) * T_s \tag{3.6}$$

where Q_g is the gate charge from MOSFET's datasheet, QLS is the charge required by the internal level shifter of the driver that in our case has been neglected,D_{MIN} the minimum required duty cycle, T_s the switching period and finally ILK_{TOT} is the total leakage current that is:

$$ILK_{TOT} = IQBS + ILK \tag{3.7}$$

both of them are related to the driver. In particular IQBS takes into account the floating quiescent current and ILK the floating leackage current.
Anyway, a value of Q_{tot} that is often taken is 10 or 20 times larger.
In this project a factor 10 has been chosen. So, $Qboot = 10Q_{tot}$ that means

$$Cboot = \frac{Q_{tot} * 10}{Vboot} \tag{3.8}$$

Vboot was fixed to 2.7V and it corresponds to the voltage drop of the bootstrap capacitor Vdrop. In the figure below it can be seen better.

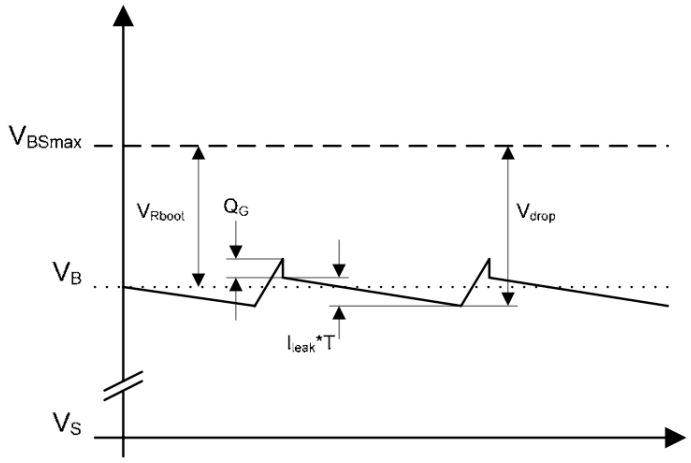


Figure 3.14: Vboot waveform

The charging and discharging of the bootstrap capacitor is represented by the VBS waveform as a function of time. During charging, the bootstrap resistor is in charge of the maximum current required by the capacitor. A portion of the charge (QG) is subsequently transferred to the high side MOSFET's gate and used to activate it. While the VBS ripple is only determined by the bootstrap capacitor, the average voltage drop (Vdrop) from the maximum achievable VBS (VBSMAX) is solely determined by the bootstrap resistor.
In order to find a possible value of Cboot all the previous equations were written in a matlab script as follows.

```
%%Cbot computation FROM INTERNATIONAL RECTIFIER (Infineon)%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%Manufacturer's data
%General
Vcc = 15; %Supply voltage of LOW-SIDE circuit
Vdrop =2.7; %Vdrop required
Dcycle = 0.1;
Rboot =10; %Bootsrap resistor
fs = 1000; %switching frequency

%From Mosfet's Datasheet IRF7380
Vgs =11;
Vds_ON =1.3; %Vdrop on MOS or Forward diode Voltage
QG =15*10^-9; %Total gate charge of high side MOS [C] (typical value)
ILK_GS =200*10^-9; %gate source leakage current

%From Driver's Datasheet IR2136
Vbs_uv_m =8.2; %if the driver has UVLO lock this is the minimum VBS voltage
IQBS =70*10^-6; %floating quiescent current
ILK =50*10^-6; %floating leakage current (it is 50uA)
QLS =0; %charge required by the internal level shifter

%From Bootstrap diode's Datasheet
ILK_diode =0; %bootstrap diode reverse bias leakage
Vboot_diode =0; %forward voltage drop across Bootstrap Diode

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

(a)

```
%formulas
Ts = 1/fs;
Vdrop_max = Vcc - Vgs - Vds_ON; %Maximum Vdrop
%ILK_TOT = ILK_GS + IQBS + ILK;
ILK_TOT = IQBS + ILK;
Dcycle_min = (((QG + QLS)*fs)+(200*10^-6))*(Rboot / Vdrop)
Qtot = QG + QLS + (ILK_TOT)*(1-Dcycle_min)*Ts;


Cboot_min = Qtot/Vdrop %take a greater value. This is the minimum one


DVBS = Qtot/Cboot_min

%Condition to check (Dmin > value)
value = (4*Rboot*Cboot_min)/Ts;
if(Dcycle > value)
    true = value
    %Vrboot = (((QG + QLS)*fs)+(200*10^-6))*(Rboot / Dcycle)
    DeltaVBS = Vdrop
else
    false = 1
    %Vrboot = (((QG + QLS)*fs)+(200*10^-6))*(Rboot / Dcycle)
    I_Rboot = (((QG + QLS)*fs) + ILK_TOT)/Dcycle;
    Vrboot = I_Rboot * Rboot;
    DeltaVBS = (2*Vdrop) - (2*Vrboot)
end

%VBS minimum computation
VBS_min = Vcc - Vds_ON - Vdrop

%Current through Rboot (average current)
I_Rboot = (((QG + QLS)*fs) + ILK_TOT)/Dcycle
%Voltage drop on Rboot
Vrboot = I_Rboot * Rboot
```

(b)

Figure 3.15: Matlab script

Because one of the elements on which the value of the bootstrap capacitance depends is the switching frequency (equation 3.6), which can have a value in the range (0-100)KHz, no single capacitance value matched this criteria. To solve the problem, two distinct capacitors were chosen, implying that the driver would drive two different channels. According to the MATLAB calculations, for low frequencies in the range (0 - 5)KHz, a capacitance of 10 nF [9].was chosen (first driver's channel), and for those in the range (5 - 100)KHz, a capacitance of 22 uF [10].was chosen (second driver's channel).
The circuit is depicted below, with the highlighted sections already mentioned.

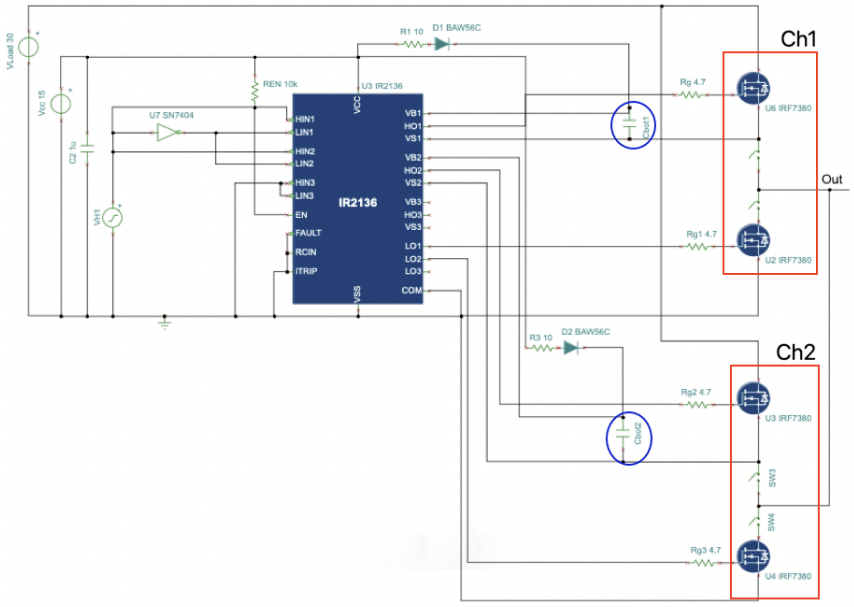


Figure 3.16: Driver's channels configuration

The choice of the gate driver IR2136 [11].that is showed in figure 3.16 will be discussed in the next section 3.3.3.
Nevertheless, regarding the bootstrap diode, particular attention has been paid to the forward voltage drop and minimum switch time as soon as possible for fast recovery or ultra-fast diode. It was chosen the BAW56 diode [23].
Rboot, the bootstrap resistor, is responsible for the current generated during the initial bootstrap charge and is normally between 5Ω and 10Ω in order to prevent Vb from falling below COM (ground). It is also in charge of charging timing. As a result, 10Ω [12]. was picked for this dissertation.

3.3.3 Driver and MOSFET selection

About the driver, the three channels MOS driver IR2136 [11].of Infineon was used thanks to its output electrical features. It is 600 V three-phase gate driver IC with enable, fault reporting and over current protection.
Nevertheless, as shown in the figure 3.16 only the enable pin has been connected and used but the fault and over current protection pins have been left unused and may be a possible future improvement.

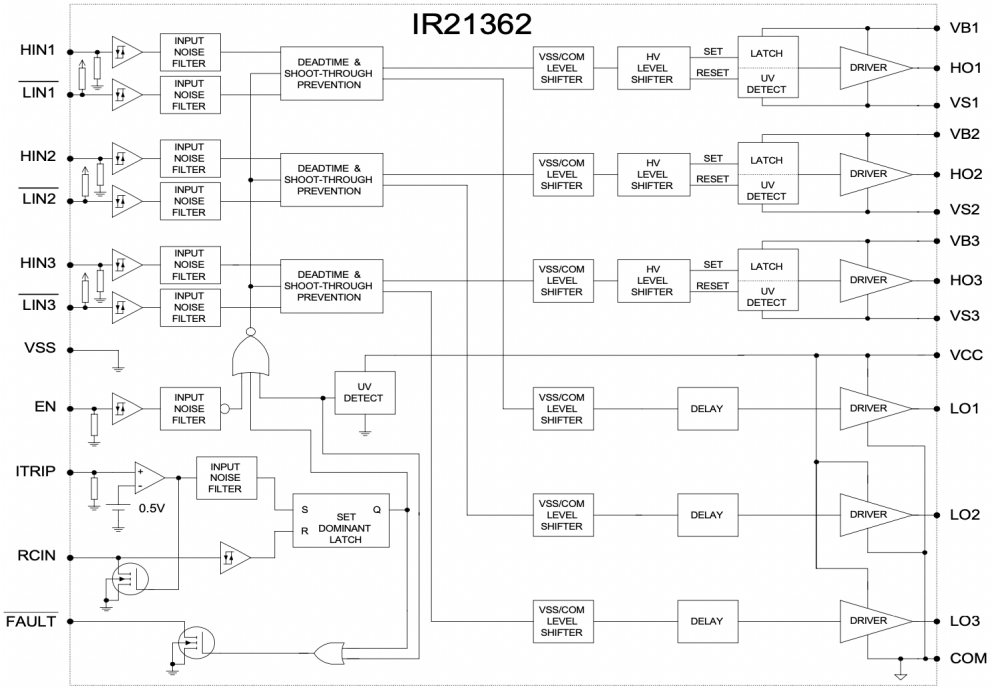


Figure 3.17: Driver functional block diagram

It presents high voltage, high speed power MOSFET and IGBT drivers with three independent high and low side referenced output channels. the inputs are compatible with CMOS or LSTTL logic. An external current sense resistor can be used as protection to turn off the six outputs of the integrated circuit in case of overcurrent.
The use of the driver is essential otherwise the microcontroller would not be able to command the mosfet configurations. The choice to use Infineon component was not accidental but to allow to simulate the real behavior of it on a Spice tool thanks to its wide components library. The online tool is free and can be used by anyone.
Regarding the choice of MOSFETs, a detailed analysis was made on the electrical and dynamic characteristics of each one. Particular attention has been paid to the RON and the drain current. Furthermore, in order to be able to simulate its real behavior using the Infineon Spice Tool, an integrated mosfet from the same manufacturer was considered. Since the board drives resistive loads, it is necessary to have a resistance RON as low as possible so as not to disturb the voltage drop across the load. Furthermore, a drain current of the order of Amperes has been considered to avoid the switch currents damage the device. This research led to the use of the IRF7380 [13]. integrated circuit.

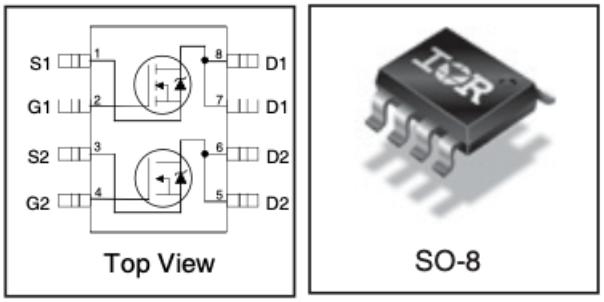


Figure 3.18: IRF7380 Dual MOS

3.3.4 Advantages and limitations

The main advantage as mentioned in the introduction is related to the cost and complexity that makes it a cheap and simple solution. Nevertheless, different limitations need to take into account.
At the same value of the bootstrap capacitance, as the switching frequency increases, the required capacitor charging time decreases and the capacitor is not able to charge completely and consequently the most high side is not able to switch completely because the charge that is transferred from the capacitor to the gate is not sufficient. This leads to a fail of the device.
The same goes for the duty cycle. Very small duties TON « T are not enough to charge the capacitor. On the contrary, very large TONs are not enough to completely discharge it.
Another issue that can occurs when the high side mos is turned off, as well as the low side, a negative voltage is recorded on its source, which can harm the driver. Because a small current goes through the MOS protection diode, Vs is negative and leads to a larger capacitor drop voltage.

3.4 PCB Design

The PCB design of the board was created using the KiCad software [14]. It already contains a lot of symbols and footprints but for this particular project some of them have been replaced with custom ones.
Moreover, on the schematic, many test points are added under the symbol "JG" and "JL": these are used to connect the oscilloscope's probe.

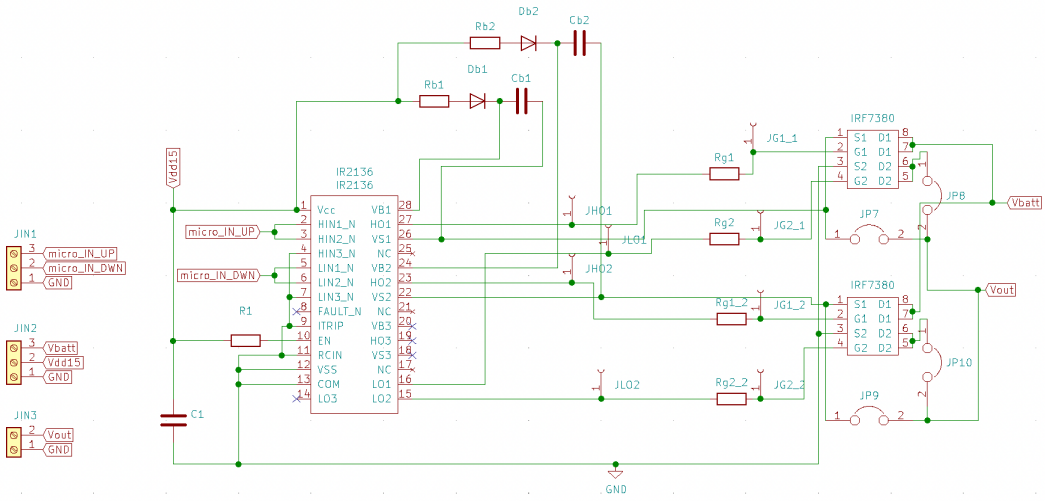


Figure 3.19: Kicad Schematic

In order to generate a final Layout, after the displacement of the components, the FreeRouting tool has been used [15] and its result is shown below. No particular board size constraints have been assigned but the minimum necessary value has been set.

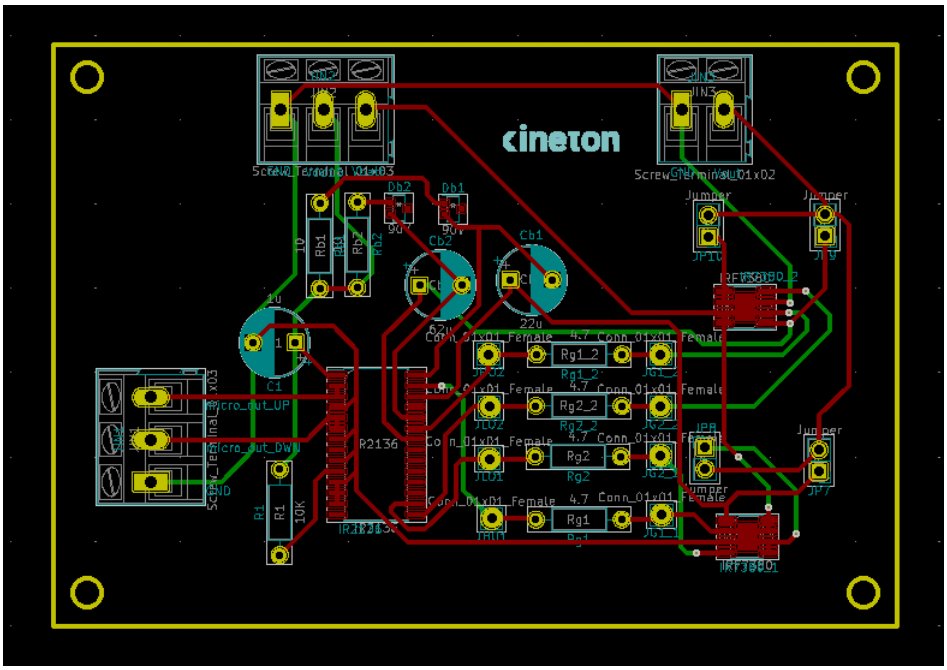


Figure 3.20: Kicad Layout

Four connectors JP7, JP8, JP9, JP10 are used like jumpers to configure the board in High side, Low Side and Push Pull according to a written map that user can consult. Three external connector JIN1, JIN2, JIN3 are used to connect the board with microcontroller (JIN1), external battery (JIN2) and ECU (JIN3).

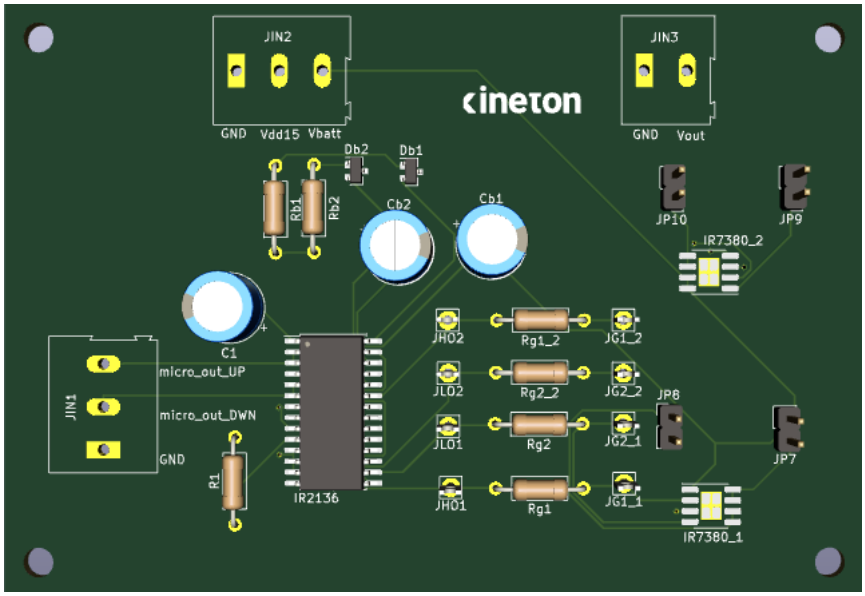


Figure 3.21: Expansion board for digital Out channels

3.5 Microcontroller programming

The Nucleo ST32F207ZGT board [16] was programmed by means STMCubeIDE [17] in different working conditions to test the board and so to replicate the PWM input signal, i.e. the one that the TMS570LC43 microcontroller [18] would send to the prototype board.
Although two differential signals are needed, the first has been simulated by programming the timer inside the 16bit Nucleo TIM1 board, the second instead depends on the first by means of the interrupt. Every time a rising or falling edge of the TIM1 is generated, an interrupt is set and generates the second differential signal.

```
void EXTI15_10_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI15_10_IRQn 0 */
    //if(HAL_GPIO_ReadPin(Micro_IN_UP_GPIO_Port, Pin_Bridge_Pin) == 1){
        if(HAL_GPIO_ReadPin(Micro_IN_UP_GPIO_Port, Micro_IN_UP_Pin) == 1){
            //L'interrupt è stato causato da un rising edge
            HAL_GPIO_WritePin(Micro_IN_DOWN_GPIO_Port, Micro_IN_DOWN_Pin, 0);

        }else{
            //L'interrupt è stato causato da un falling edge
            HAL_GPIO_WritePin(Micro_IN_DOWN_GPIO_Port, Micro_IN_DOWN_Pin, 1);
        }

        //Cancello il flag dell'interrupt
        __HAL_GPIO_EXTI_CLEAR_FLAG(Pin_Bridge_Pin);

  /* USER CODE END EXTI15_10_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(Pin_Bridge_Pin);
  /* USER CODE BEGIN EXTI15_10_IRQn 1 */

  /* USER CODE END EXTI15_10_IRQn 1 */
}
```

Figure 3.22: Interrupt Routine

Three GPIOs have been set up. Two of them used as Timer and GPIO OUT for the two differential signals "Micro_IN_UP" and "Micro_IN_DOWN," respectively, while the third used as input capture "Pin_Bridge," capturing the rising and falling edges of the timer end to generate the interrupt.

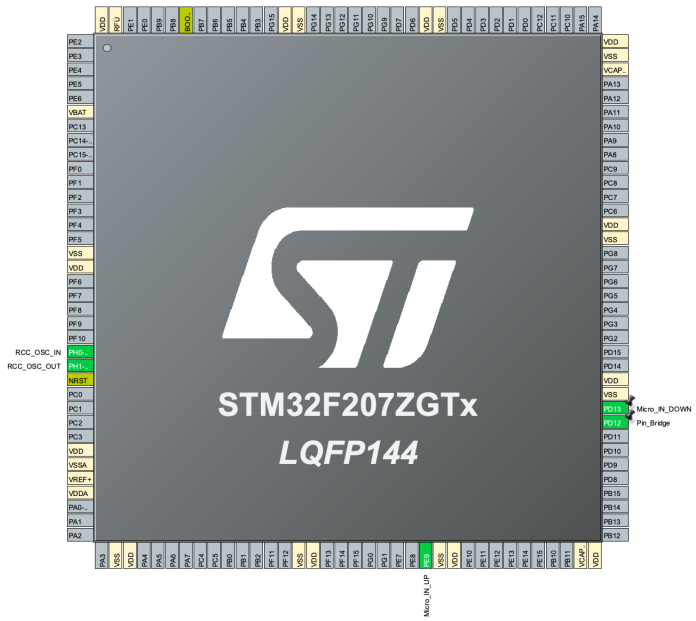


Figure 3.23: STM32 Configuration

Before the timer configuration, the internal clock has been set to 60 MHz as

well as the timer clock source. Starting from this frequency and according to the required frequency for each simulation, the prescaler of TIM1 has been adjusted in order to obtain different signal frequencies. Regarding to the Duty Cycle, different values were examined for the tests sent to the microcontroller through the CCR register.

```
int main(void)
{
  /* USER CODE BEGIN 1 */
  int TIM_CLOCK, APB_TIM_CLOCK, ARR, prescaler;

  int frequency = 10; //SCRIVERLA IN HZ


  APB_TIM_CLOCK = 60000000;
  ARR = 100;
  TIM_CLOCK = ARR * frequency;
  prescaler = (APB_TIM_CLOCK/TIM_CLOCK)-1;

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

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

  /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_TIM1_Init();
    MX_RTC_Init();

    //setto il Duty del timer principale
    TIM1->CCR1 = 50;

    //Attivo il timer principale
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);

    while (1)
     {
       /* USER CODE END WHILE */

       /* USER CODE BEGIN 3 */

     }
```

Figure 3.24: Main function

The TIM1 follows the frequency and DC set by the user. The same is then connected externally to the 'Pin_Bridge' which captures the edges of the signal itself and triggers an interrupt in which it activates the GPIO channel 'Micro_IN_Down'.

Chapter 4

Simulations

In the last part of the project we dealt with testing the prototype board under all the required specifications, verifying the correct functioning and comparing the different measurements with each other, up to defining the maximum operating limits. The first part will show the simulations obtained through the Infineon Tool Spice while the real ones will be shown in the last part.

4.1 Spice-Infineon Simulations

Just to have a comparison with the spice simulations, three simulations are shown below in Low side, High Side and Push Pull configuration with a different Duty Cycle and two different sample frequencies of 10Hz respectively to test the low frequencies and 100KHz for the high frequencies. All simulations refer to the setup circuit in figure 4.1.

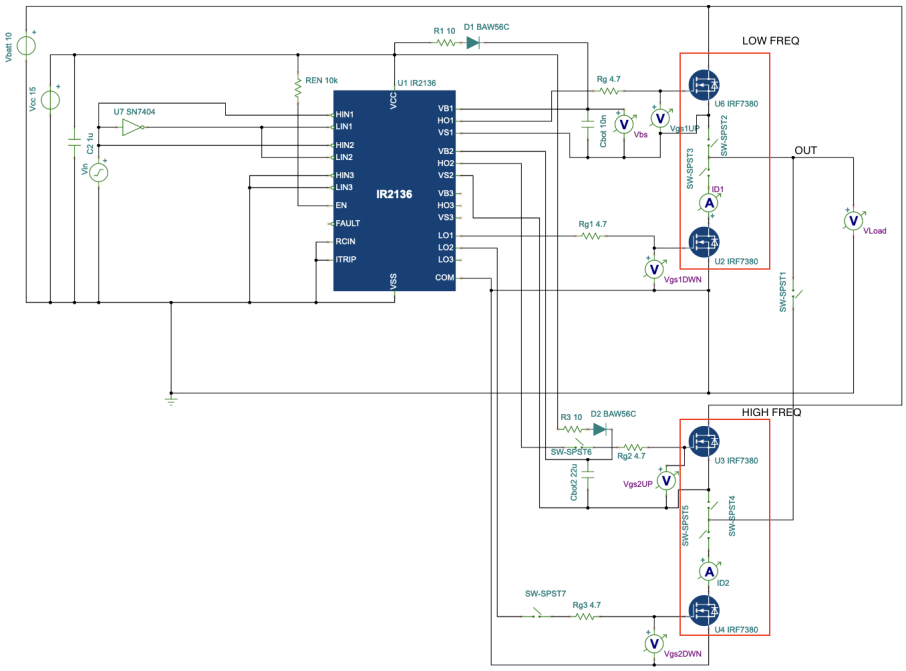


Figure 4.1: Setup Circuit for Push Pull configuration

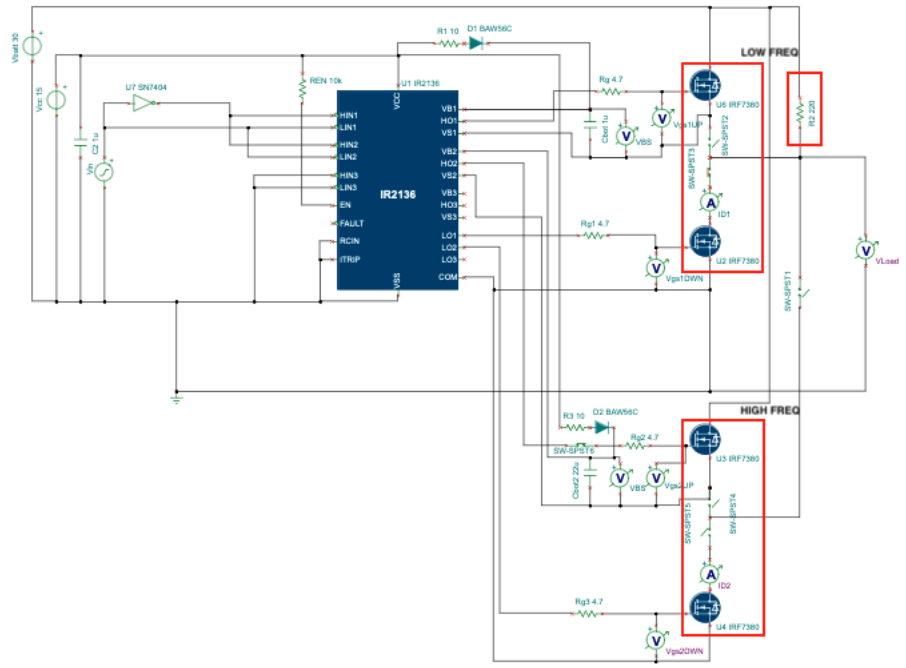


Figure 4.2: Setup Circuit for High Side configuration

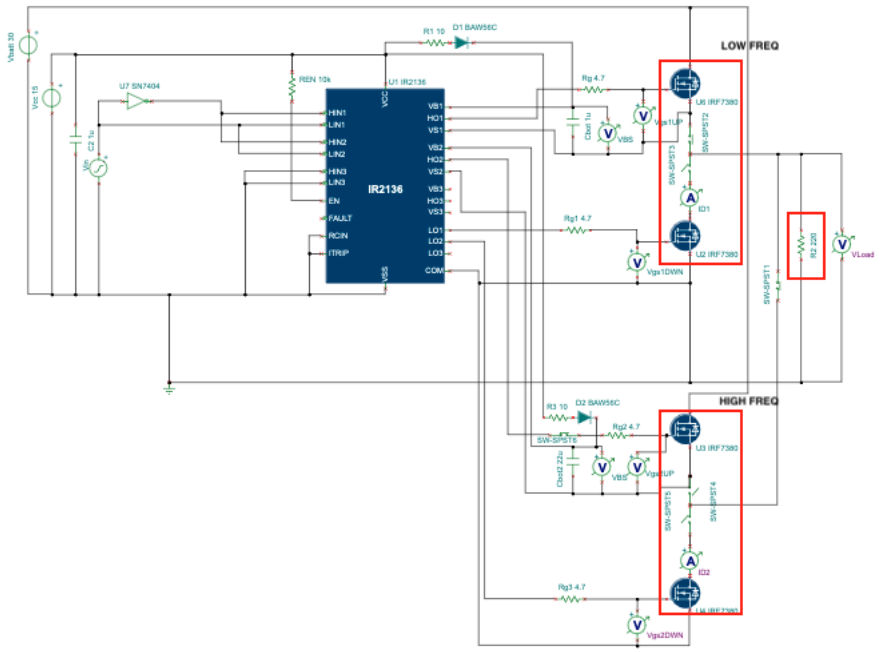
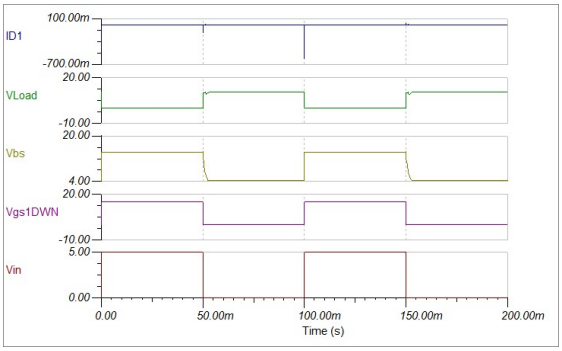
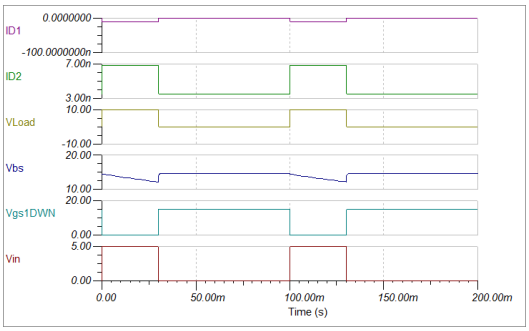


Figure 4.3: Setup Circuit for Low Side configuration

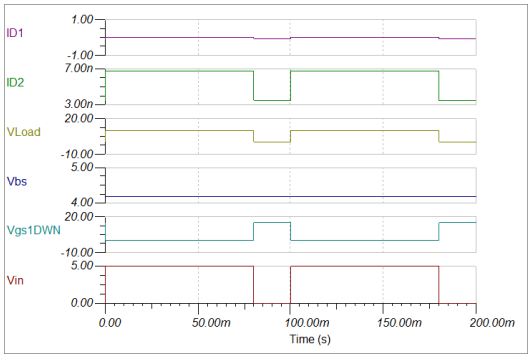
As mentioned in chapter 3 section 3.3.1, when the High Side and Low Side configuration is simulated, in order to have a correct view of the output waveform, a 220Ω [19] resistor is placed at the output of the configuration. The only difference between the two is that in the high side case the resistance is set to GND, vice versa to battery voltage.



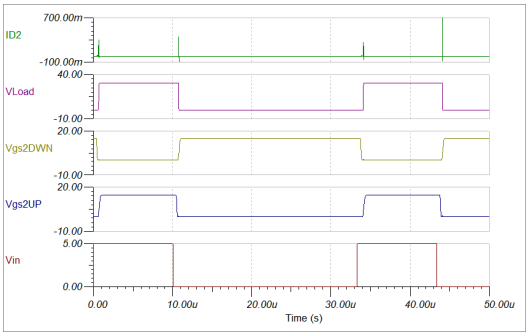
(a) Vbatt = 10V, f = 10Hz, DC 50%, Push Pull



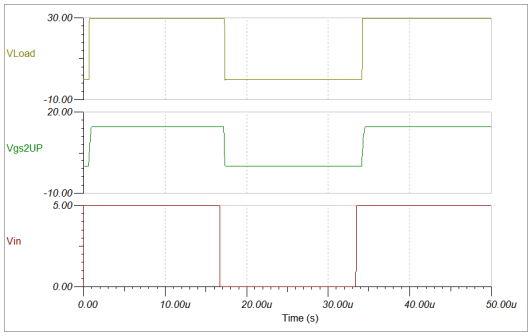
(b) Vbatt = 10V, f = 10Hz, DC 30%, High Side



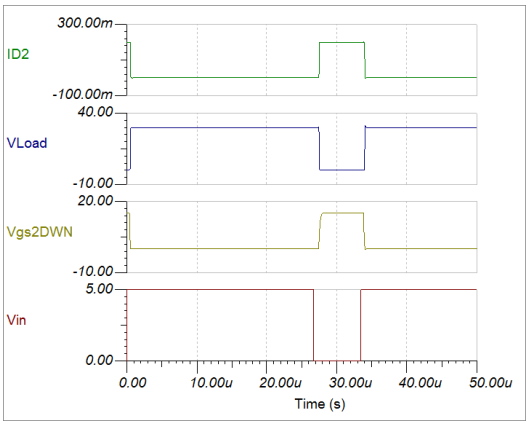
(c) Vbatt = 10V, f = 10Hz, DC 80%, Low Side



(d) Vbatt = 30V, f = 30KHz, DC 30%, Push pull



(e) Vbatt = 30V, f = 30KHz, DC 50%, High Side

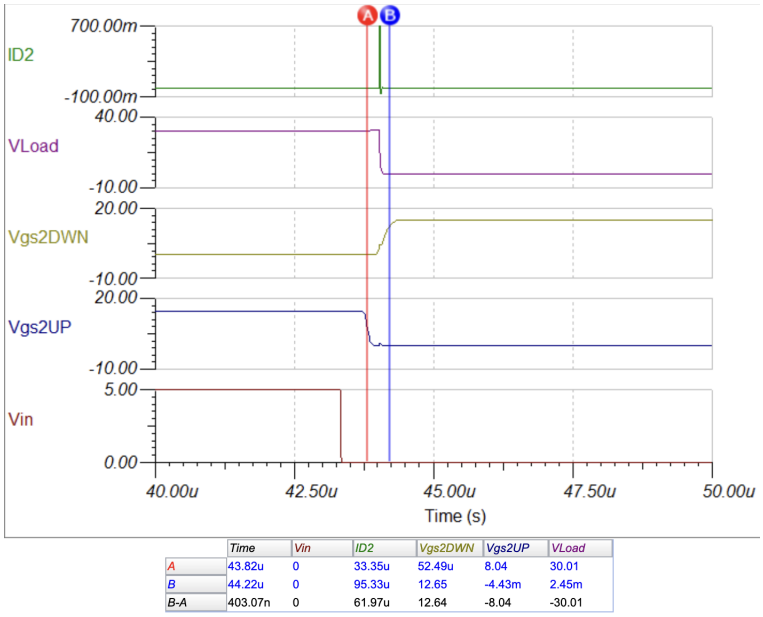


(f) Vbatt = 30V, f = 30KHz, DC 80%, Low Side

Figure 4.4: Spice Simulations

In order to also test the response of the circuit in the voltage range 0-30V, it was also simulated at the maximum voltage 30V figure 4.4 d, e and f. Of course, in

order to be able to simulate with an higher voltage, in the case of the low side and high side a power resistance with an higher resistance value of 220Ω [19] was used so as not to damage the load itself. Particular attention has been paid to the drain current in the case of push-pull configuration. This current has been one of the points of discussion on the choice of the proper pair of mosfets. This current is less than 700mA as shown 4.4d. This is due to the instant of switching of the high side and low side mosfets although the activation of one and the deactivation of the other does not occur simultaneously as in the ideal case but in a time equal to about 400ns which, although limited, is not negligible (figure 4.5).



	Time	Vin	ID2	Vgs2DWN	Vgs2UP	VLoad
A	43.82u	0	33.35u	52.49u	8.04	30.01
B	44.22u	0	95.33u	12.65	-4.43m	2.45m
B-A	403.07n	0	61.97u	12.64	-8.04	-30.01

Figure 4.5: Switching time

About the Duty Cycle, different tests were performed at the same frequency by adjusting the Duty Cycle between the minimum and maximum value but no particular limits were observed, except for the frequency. For the latter, the upper limit found through the spice simulation for each channel was different than expected. Channel 1 (low frequencies) follows the design constraints imposed by the Cbot1 capacitance chosen through the Matlab script. In terms of channel 2, the maximum frequency range found was 30KHz. Once this frequency is exceeded, the channel begins to distort the signal and the duty cycle is degraded. The figure below 4.6 shows a measurement at a frequency higher than 30KHz where the problem is highlighted.

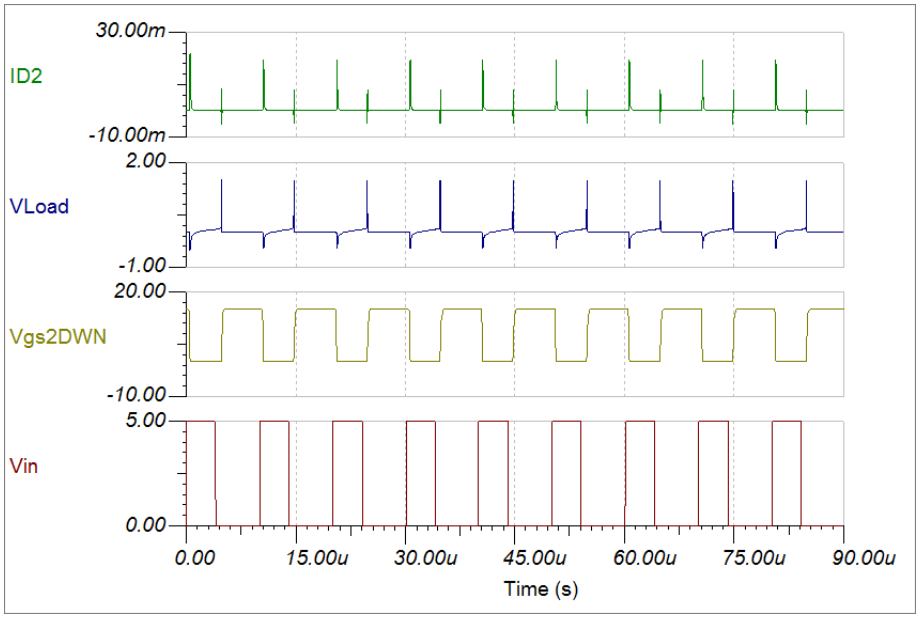


Figure 4.6: Vbatt = 30V, f = 100KHz, DC 40%, Push Pull

The unusual behavior is due to the bootstrap capacitance which is not suitable for that frequency. As expected, this value was chosen so as to cover the maximum possible range and changing capacitance would have been harmful for the lower frequencies.

4.2 Real tests

The simulations were carried out by powering the board at 15V, the driver's power supply voltage, and supplying it with different battery voltages (Vbat) ranging from 0 to 30V according to requirements. Although the board has been tested from 0 to 30V, the tests in this section will only be demonstrated at two different reference voltages. For high voltages the reference was 30V while for low ones 15V. Regarding the frequencies, since the maximum limit found was 20KHz (unlike spice simulations), in the first part we will analyze only the cases 10Hz to test the low frequencies and 20KHz for the high ones. In the last part two cases will be shown in which an attempt has been made to go beyond the 20KHz frequency. For the same reasons previously mentioned also in these solutions a 220Ω 5W resistor [19] was used to test the high side and pull down configurations.
The instrumentation used to do this is as follows:

- Oscilloscope Hantek DSO5102P [20]
- Two Power supply LAVOLTA BPS-305 [21]

- Nucleo board STM32F207ZGT, to generate a PWM square wave [16]
- Current probe Pico TA018, to measure the switching current [22]
- Voltage probe Pico TA375

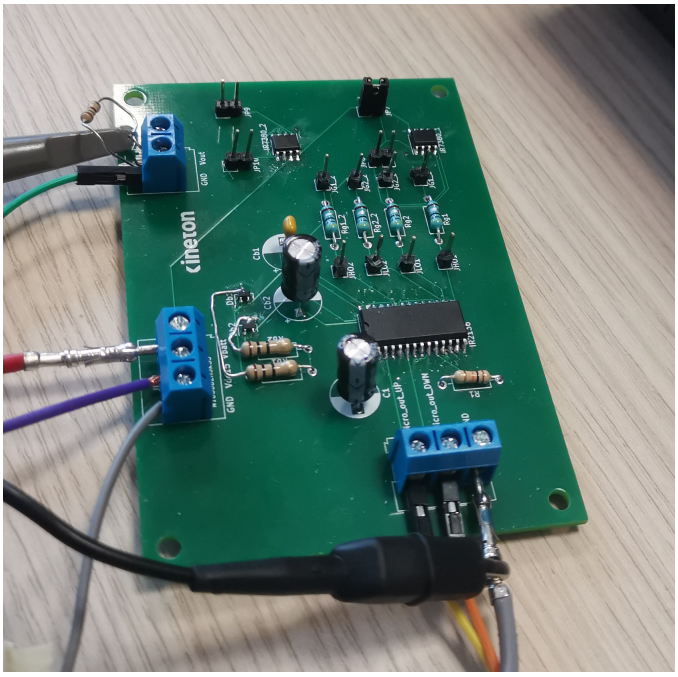


Figure 4.7: Real board

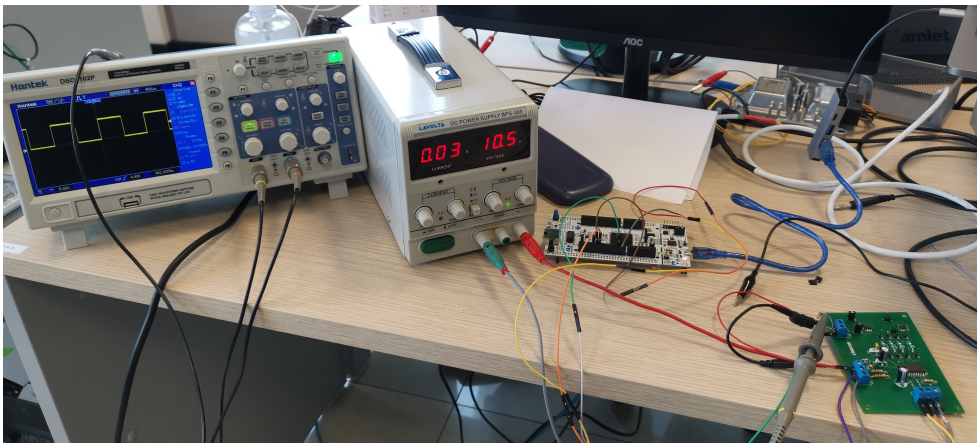


Figure 4.8: Setup Circuit

All tests were carried out using the two channels of the Hantek oscilloscope. The voltage probe was connected to CH1 (yellow trace) while the current probe was connected to CH2 (blue trace). About the switching current, will be reported only the critical case of simulation such as those with high voltage battery (30V) and push pull configurations in which no load resistor was used.

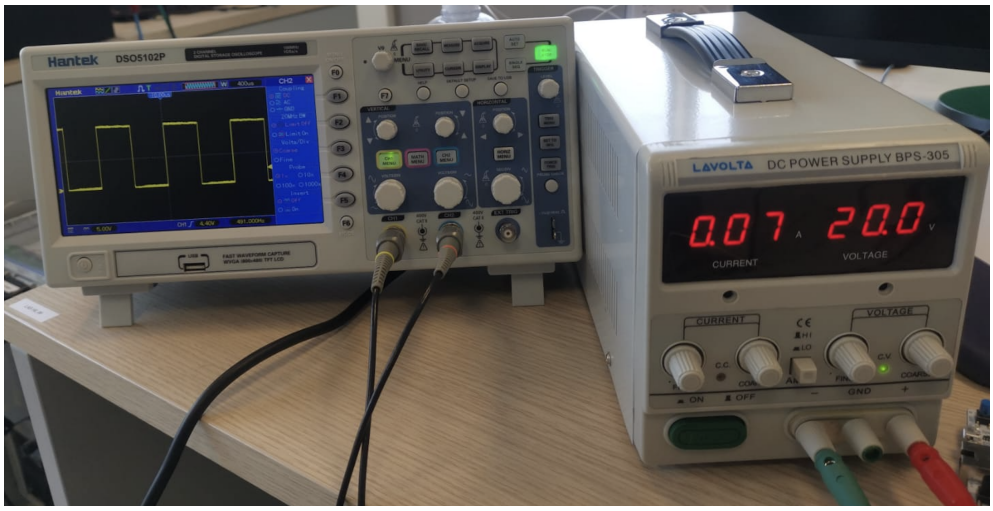
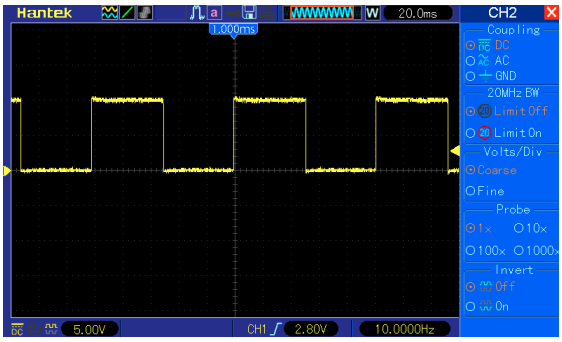


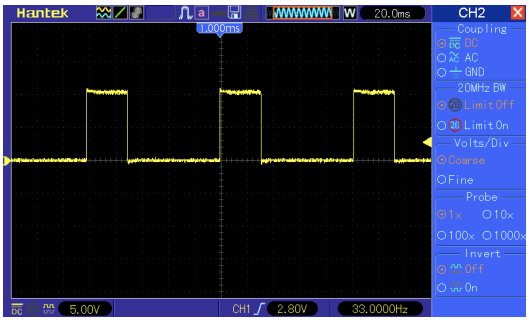
Figure 4.9: Real test

4.2.1 15V test Low Frequencies

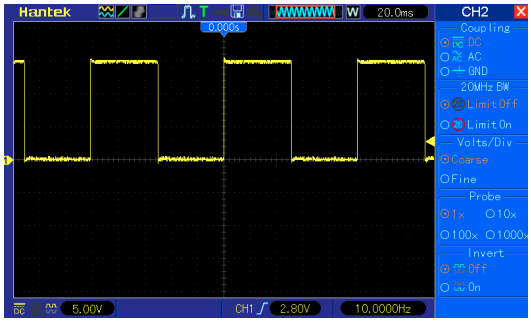
The first channel of the board was tested at three different configurations with a reference frequency of 10Hz and Vbat 15V. Changing the duty cycle the output response was evaluated and shown in figure 4.10.



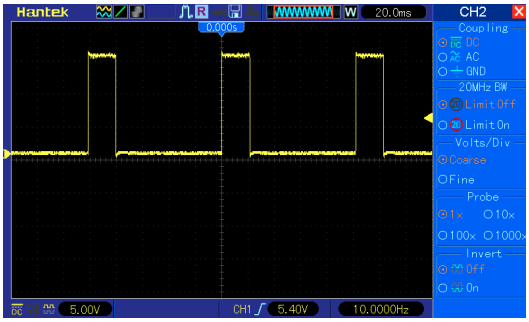
(a) Duty Cycle 50%, Low Side



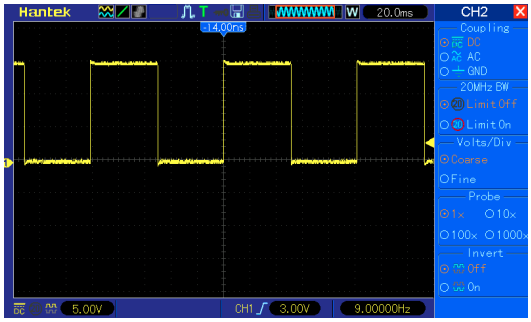
(b) Duty Cycle 30%, Low Side



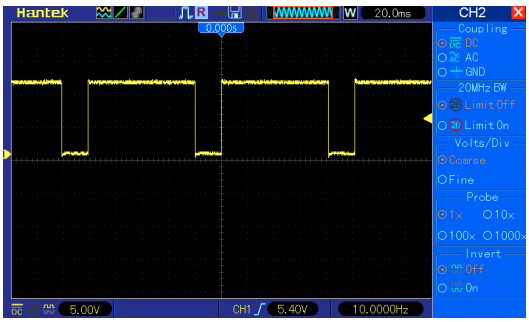
(c) Duty Cycle 50%, High Side



(d) Duty Cycle 20%, High Side



(e) Duty Cycle 50%, Push Pull



(f) Duty Cycle 80%, Push Pull

Figure 4.10: Output signals at Vbat = 15V, f = 10Hz

As a first observation, the output waveform is not disturbed and the duty cycle is aligned with the input one. For these tests changing the configuration does not affect the output signal. Regarding to the propagation delay, the amount of time

required by the signal from the input until the output, can be observed in figure 4.11

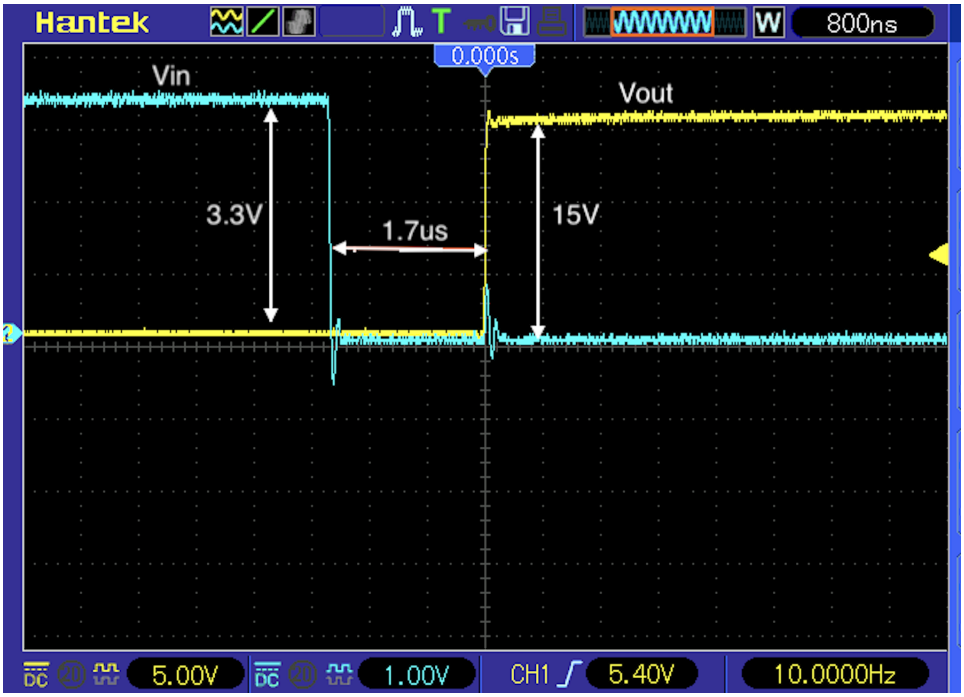


Figure 4.11: Propagation Delay

Although the driver outputs are inverted with respect to their input signals, this means that an input rising edge corresponds to an output falling edge and vice versa, oscilloscope channel 2 (blue trace) represents one of the two driver input signals and channel 1 (yellow trace) its respective output. The observed propagation delay is about 1.7us that compared with the switching frequency 10Hz so a period of 100ms and duty cycle 50%, it can be neglected.
This time depends by the architecture of the device, in particular by the internal delay of the driver. Changing duty cycle, the delay remains constant.
In the figure 4.12, through the current probe connected to the channel 2 (blue trace), the short-circuit current was measured in the push-pull configuration, the most critical one because both mosfets (UP and DOWN) are involved.

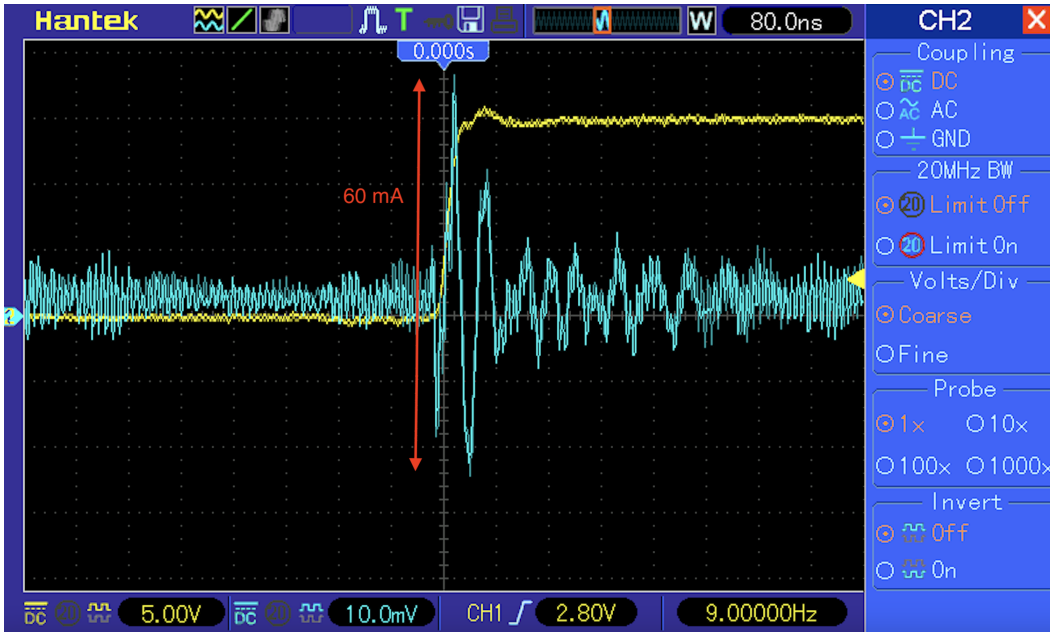


Figure 4.12: Switching current at 10Hz Push Pull configuration

With this measurement we wanted to verify that the switch current was lower than the maximum drain current of the chosen mosfets. Even if the probe is quite disturbed, the maximum current peak reached at the instant of switch is 60mA. This result confirms that the device is working properly.
The same current was measured again at a greater frequency maintaining the same configuration and battery voltage Vbat. The result is shown in figure 4.13.

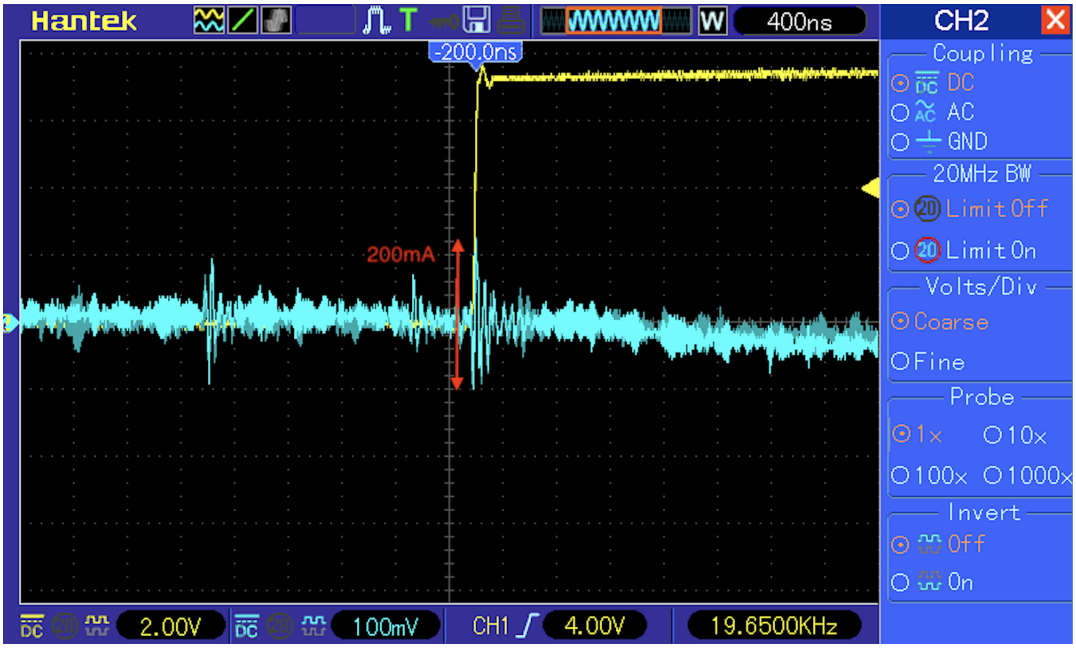
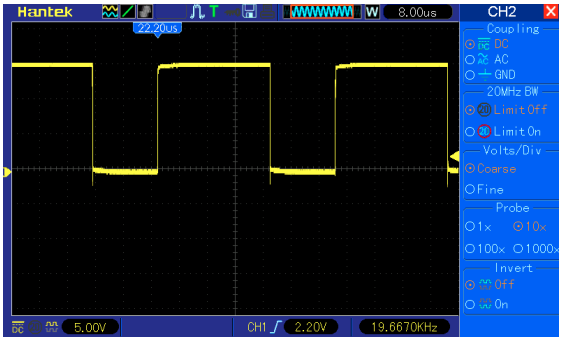


Figure 4.13: Switching current at 20KHz Push Pull configuration

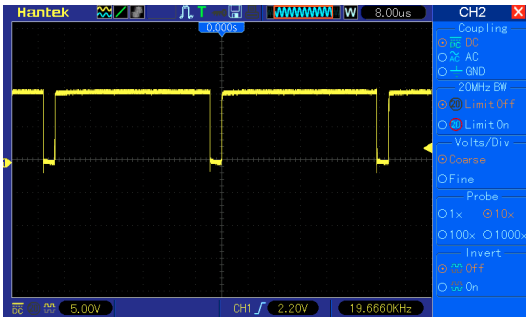
In this last case, the peak current is about 200mA so higher than the previous one. This important result shows as the switching current depends by the frequency. However, the frequency is limited since the current cannot exceed the maximum drain current established by the mosfet manufacturer. Because the latter is around 3.6A, this example also works properly.

4.2.2 15V test High Frequencies

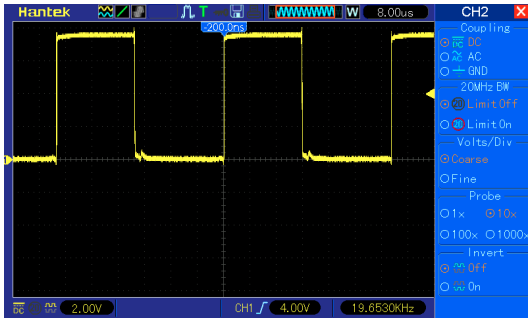
Compared to the previous section, this time keeping the battery voltage at 15V as before, we want to test the effects of the output signal as the switching frequency increases. Unlike the previous tests, this time channel 2 of the driver is used, which is the one for the high frequencies. The reference frequency was 20KHz.



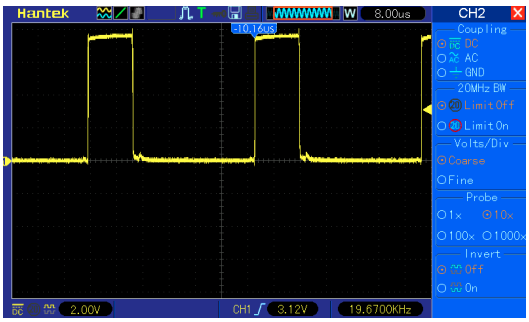
(a) Duty Cycle 60%, Low Side



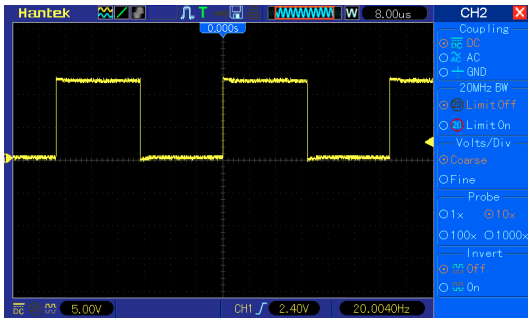
(b) Duty Cycle 90%, Low Side



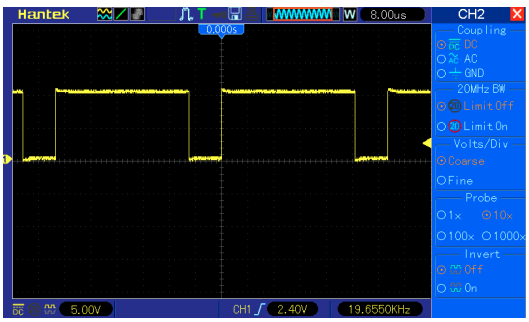
(c) Duty Cycle 50%, High Side



(d) Duty Cycle 30%, High Side



(e) Duty Cycle 50%, Push Pull



(f) Duty Cycle 80%, Push Pull

Figure 4.14: Output signals at f = 20KHz

In the signal transitions of the all three configurations, the output waveform begins to be affected by overshoot. While this phenomenon at 15V is minimal, it can

be accepted for our purposes. What instead begins to be relevant is the duty cycle. In fact, at this frequency, it starts to not be as accurate as we expect. Compared to the duty cycle of the input signal, the one arriving from the microcontroller, a slightly different value is observed at the output, figure 4.15.

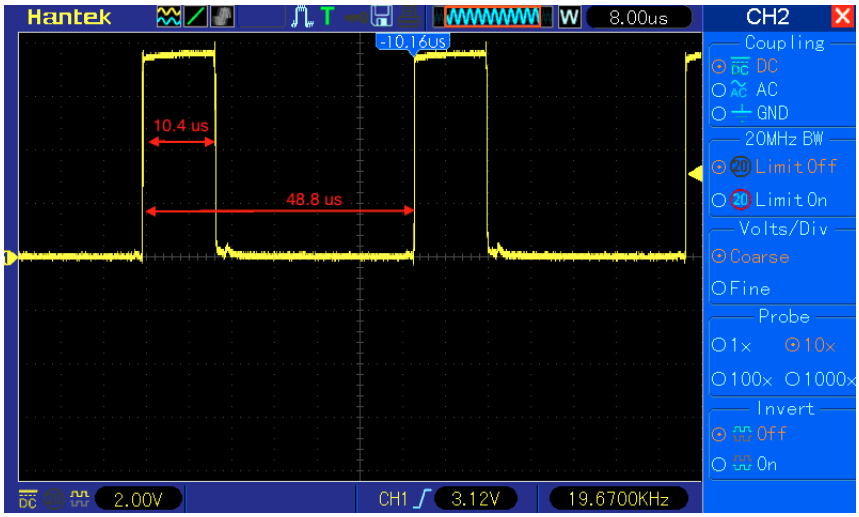


Figure 4.15: High Side configuration output signal

Defining the Duty Cycle as $DC = \frac{T_{ON}}{T}$, is evident that it is not as completely accurate as the input one (30%), so increasing the frequency the DC information is lost with an error of 8-9%. Also increasing the duty to higher value than 30%, it was demonstrated that the error remains about the same. For each different input Duty Cycle it swings in the range (5-9)%. This means that it does not strongly affected by DC value itself but only with the frequency value. Changing the configuration to Low Side, it continues to be different from the one set at the input. Figure 4.16

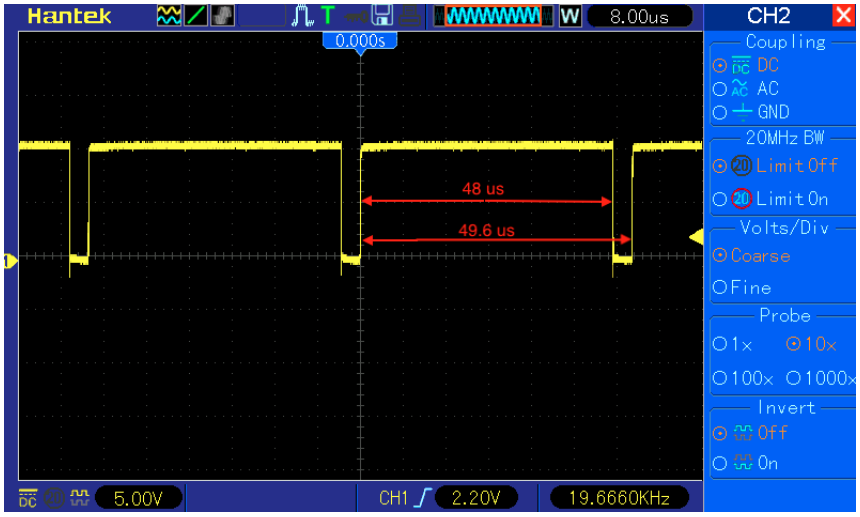
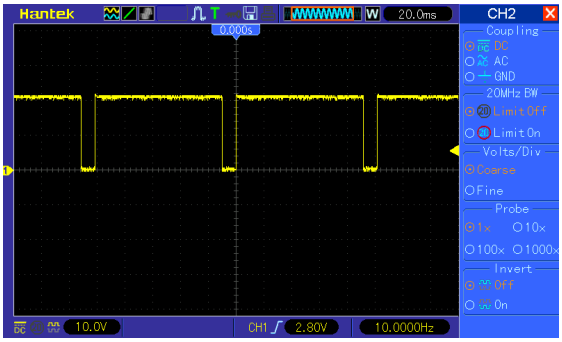


Figure 4.16: Low Side configuration output signal

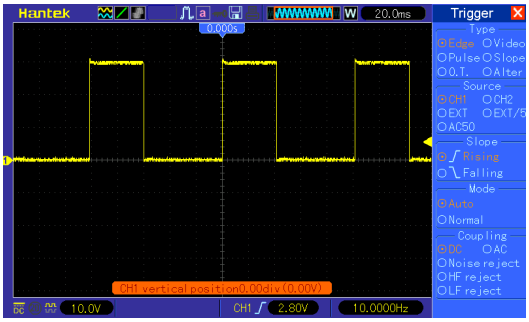
In this last case the expected duty cycle was 90% like the input one but a DC of about 96% is obtained at the output. So, another error of about 6-7% occurs. A different behaviour occurs for the Push Pull configuration where the duty cycle of the output signal follows the input one. An example can be seen in the figure 4.17f.

4.2.3 30V test Low Frequencies

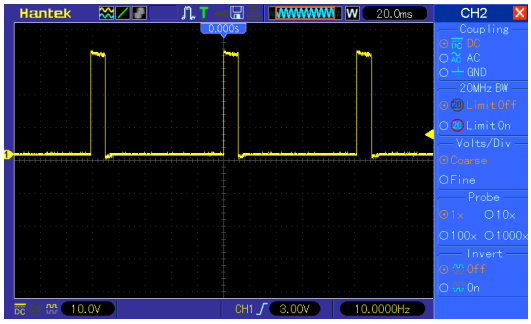
In the last part of the simulation we tried to understand what were the limits of the board in terms of voltage, frequency and duty cycle. Although the choice of the bootstrap capacitor for both driver channels has been made in a range of possible values such as to cover the largest frequency range, real limits are expected from these simulations which are not appreciable in the Spice simulation.
Furthermore, considering the tests with a battery voltage higher than the previous tests, it was verified through the current probe that the switch currents were in the maximum range supported by the pair of MOSFETs. In the first part, the reference frequency for the low frequencies will always be 10Hz.



(a) Duty Cycle 90%, Low Side



(b) Duty Cycle 40%, High Side



(c) Duty Cycle 10%, Push Pull

Figure 4.17: Simulations at Vbat = 30V, f = 10Hz

In figure 4.17 three different configuration were tested at three different Duty Cycle. The aim of these, was to verify that by changing the configuration, the output waveform also changed without any disturbance. Apart from a slight overshoot in the transitions the output waveform is not disturbed and the maximum voltage reached as expected is 30V. The duty cycle seems to respect the input setting except for a small error. This is evident in the push pull configuration, figure 4.18

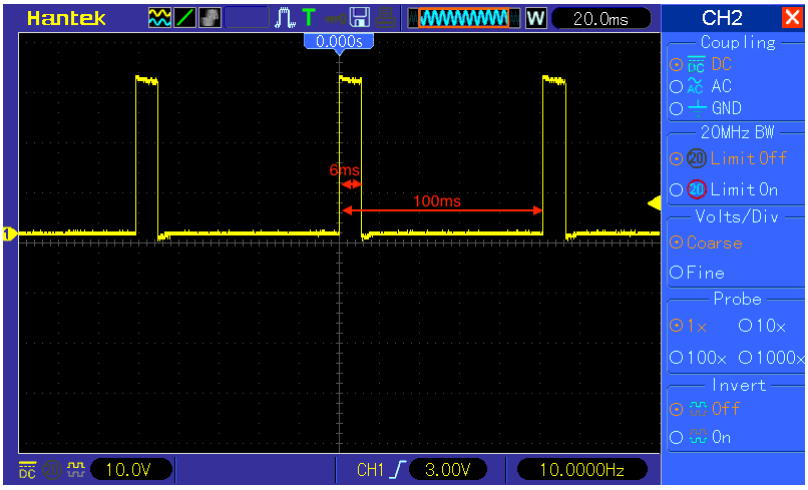


Figure 4.18: Push Pull configuration output signal

Setting an input Duty Cycle of 10%, the obtained output is about 6%. Again, it is affected by an error of 4%. Taking into account the current at this frequency we get about 540 mA peak to peak as in figure 4.19 that is acceptable.

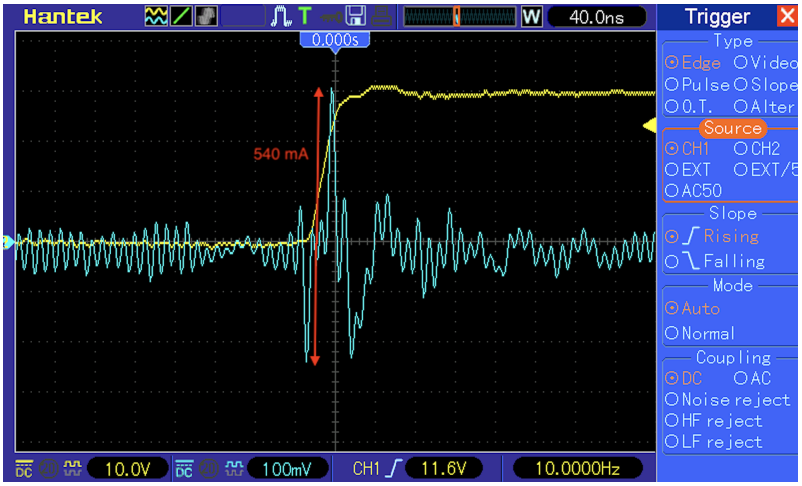
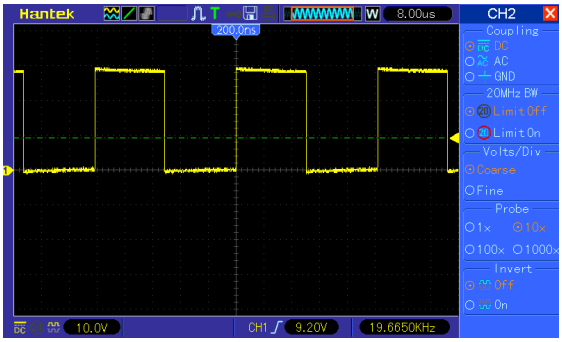


Figure 4.19: Switching Current, Push Pull, 10Hz

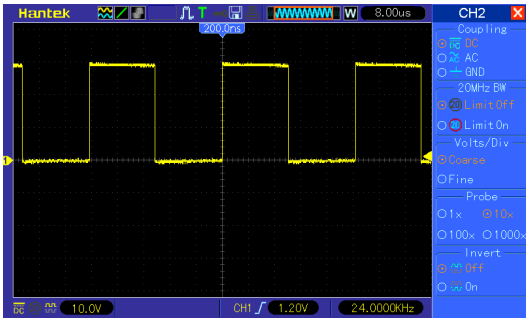
4.2.4 30V test High Frequencies

Until now, simulations have focused on the maximum voltage, therefore switch current, and the correct duty cycle in relation to the input one. However, an attempt has also been made to determine the maximum operating frequency limit, with any anomalies being evaluated as this grows. In the next tests, we will attempt

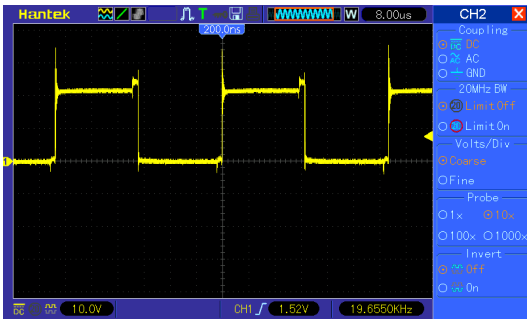
to understand how, starting from the maximum battery voltage of 30V, the circuit behaves as the frequency increases. As usual, at the reference frequency of 20KHz.



(a) Duty Cycle 90%, Low Side



(b) Duty Cycle 40%, High Side



(c) Duty Cycle 10%, Push Pull

Figure 4.20: Simulations at Vbat = 30V, f = 20KHz

Again, in order to evaluate the behavior on the output waveforms, the simulations in figure 4.20 have been reported, where random duty cycles have been chosen for each configuration. The first obvious observation comes from the Push Pull configuration. In the latter, in fact, the problem of overshoot is highlighted and leads the output in the instants of switch to considerable voltage peaks that could damage the device for a long time. Not considering the switching interval, the amplitude of the output signal is also reduced at about 20V instead of 30V as the input battery voltage. About the overshoot, being the pair of MOSFETs oversized compared to the maximum voltage 30V, in fact the maximum V_{DS} for each of the two is 80V, the device still manages to function correctly.
One of the reasons these voltage spikes are observed, which is a problem as mentioned in chapter 3 section 3.3.4, is due to the negative voltage observed at the VHS node of the driver. An explanation for this can be given by looking at figure 4.21 below.

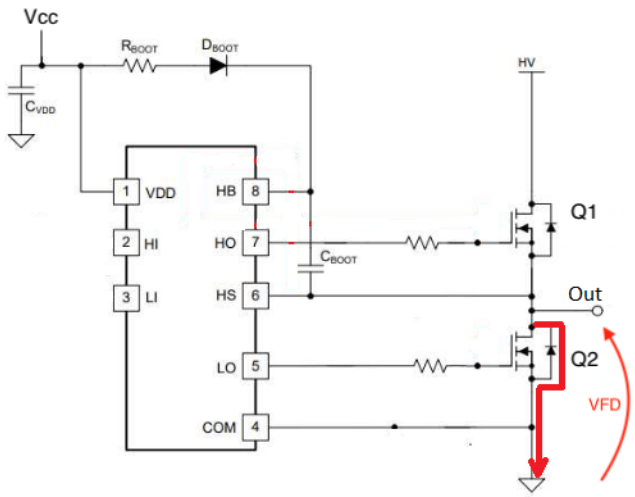


Figure 4.21: freewheeling diode current

In the instant that the gate High Side (Q1) closes and Low Side (Q2) does not yet open, there is a small current in the freewheeling diode of Q2, which causes an undershoot voltage on pin High Side (HS) of the driver, therefore on the source of the high mos. In the figure 4.22 is highlighted the time in which the MOS Q1 switches off and Vs has an undershoot.

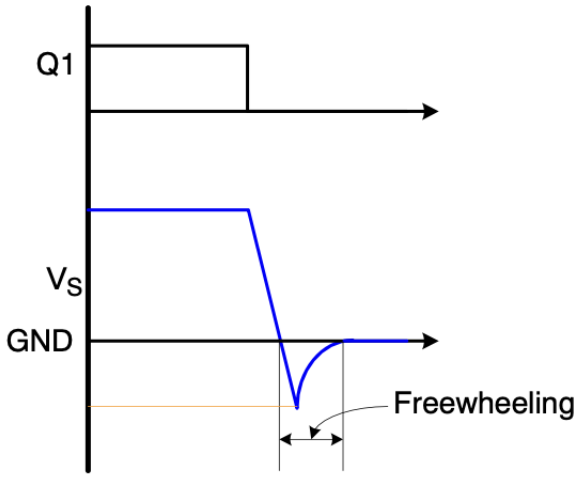


Figure 4.22: Vs (HS) waveform

However, this voltage can cause an overvoltage on the bootstrap capacitor. The latter, is charged across the diode by Vcc which in this case is 15V but since Vcc is referred to ground, the maximum voltage that the capacitor can see is the sum of

Vcc and the amplitude of the negative voltage. This is VFD = |VHS - GND|. So, at the end the maximum voltage that can drop on the bootstrap capacitor is:

$$VCboot = Vcc + VFD \tag{4.1}$$

Also in this case the maximum switch current was evaluated in the push-pull configuration, figure 4.23

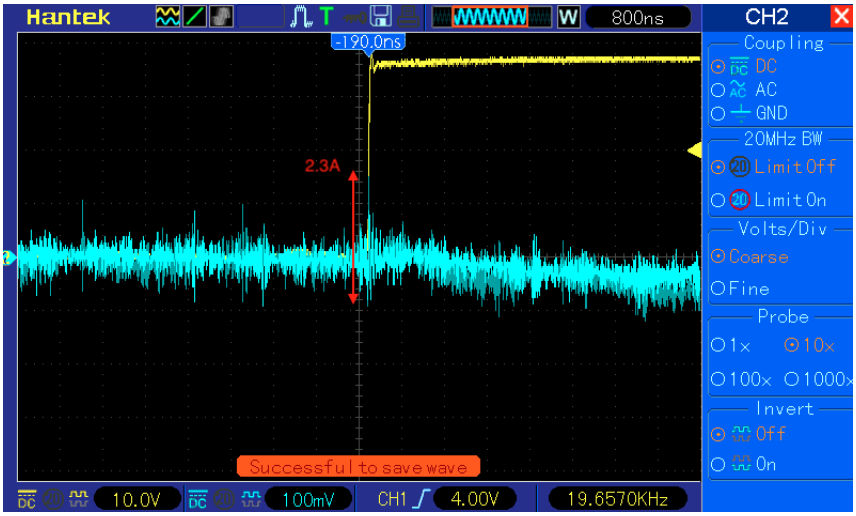


Figure 4.23: Switching Current, Push Pull, 20KHz

Even if the current probe is affected by noise, a higher current peak is evident in the transition than was measured. The 2.3A value, although it is included in the maximum current range of the MOSFETs pair, is a very high current value which could still lead to damage to the device in the long time.
Other higher frequency simulations have been performed and reported below.

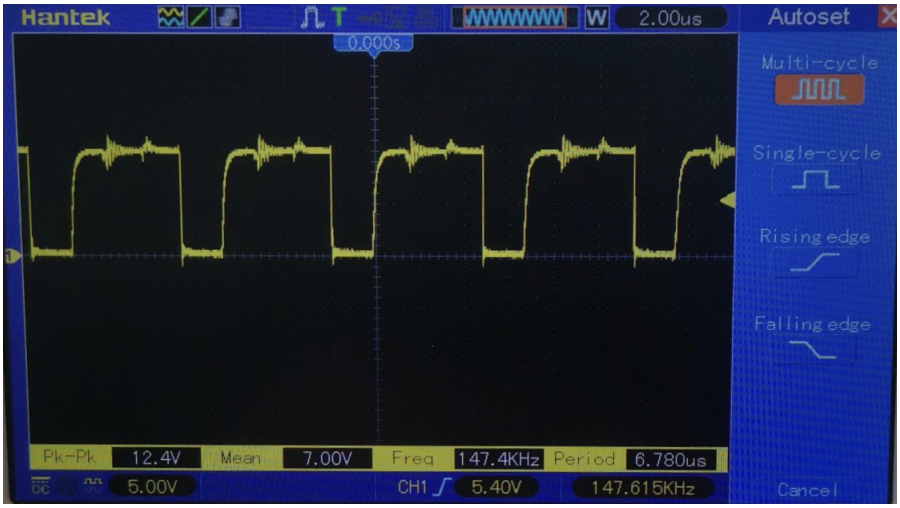


Figure 4.24: Output waveform, High Side, 150KHz, Duty Cycle 50%

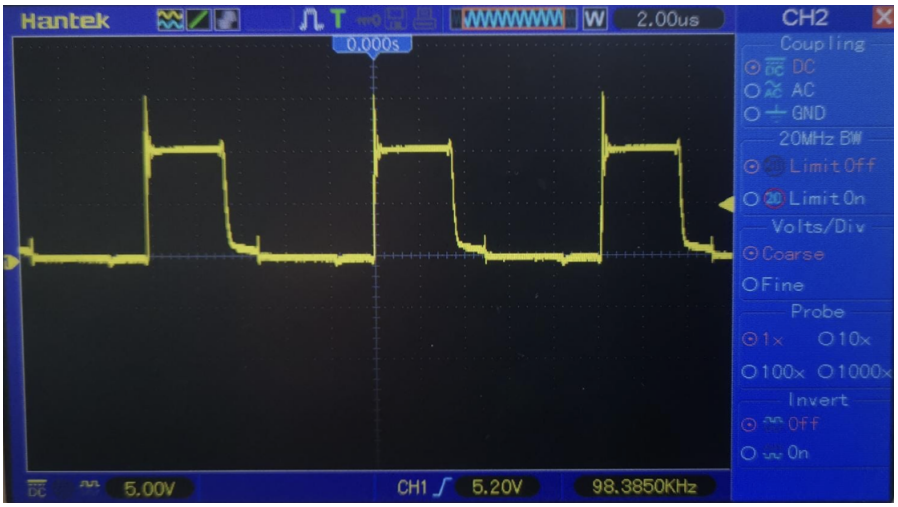


Figure 4.25: Output waveform, Push Pull, 100KHz, Duty Cycle 50%

Naturally, as can be observed in both cases, at higher frequencies, not only is the waveform disturbed but also the duty cycle no longer follows the input one. This represents a limitation for the device. Regarding the possible causes of the disturbed output waveforms there may be an inductance problem with the probe used in the measurement, parasitic effects due to the layout. Instead, the limitation of the duty cycle is essentially due to the limitation of the bootstrapping architecture.

Chapter 5

Conclusions

The developed thesis work was born from the need to have an expansion board for digital output channels used in HIL simulators. Each simulator generally has a hardware configuration customized to the reference project. This means that at the end of the same it is not said it is possible to adapt it to a new one since the number of channels requested may not be available from the same board. In this regard, the expansion board has the purpose of increasing the functionality of an already existing board but with a reduced number of channels.
Following the technical specifications required by a typical digital output channel of the simulator, the following design constraints came out:

- Operating frequency 0-100KHz
- Duty Cycle 0-100%
- Output voltage 0-30V
- High Side, Low Side, Push Pull configuration

The first phase of the thesis was mainly characterized by the research on the possible solutions to the case, where three possible approaches have been identified:

- Separate configuration architecture
- H-Bridge architecture
- Gate driver with Bootstrapping architecture

The first two did not lead to good results after a circuit analysis in the first case and an evaluation of the operation through the relative datasheet in the second. The solution with the bootstrap capacitor proved to be the best even if with limitations. In particular, although the demand working frequency was in a wide range, it is

not possible to find a single capacitor that satisfies this condition.
One solution was to work with two different capacitors, this took to use a multi-channel driver and divide the frequency into two ranges (0- 5KHz) and (5KHz - 100KHz). The choice of the capacitor for each of the two ranges was made after an analytical analysis for a given operating frequency in the respective range. In order to exploit it as better as possible, therefore in the widest possible range of frequency values, different capacitance values were calculated for several frequency points within the two ranges and an intermediate capacitance value was taken between them. This choice then led to consequences, such as limitations.
Through a considerable number of real simulations the following conclusions has been reached. For low frequencies (0-5KHz) the device complies with the project specifications in all three configurations. As the frequency increases, in particular after the 20KHz, the limit frequency of the device, the same begins to distort the output waveform in all three configurations and the duty cycle begins to no longer be coherent with that of the input signal. This represents the maximum operating limit of the device.

5.1 Future Improvements

A possible future challenge could be to exceed the 20KHz frequency for the Bootstrapping architecture, keeping under control the maximum switch currents in the three required configurations. The latter could be reduced as the transition time of each MOS High Side and Low Side driving signal decreases, thus minimizing the rise and fall times. Limitations on frequency and duty cycle may be overcome by finding more accurate capacitances or adding more channels with different frequency ranges. Furthermore, the switches used to switch between configurations can be replaced by switches controlled by microcontroller.
Although the IR2136 driver offers the possibility of having a protection circuit for the outputs which is able to disable them according to the maximum current read on the ITRIP pin of the driver itself, a possible improvement of the board may be to design an output protection stage which allows do not exceed the maximum current that can be supplied by the channel.
Finally, regarding the Separate configuration architecture, a possible future challenge could be to find on the market devices capable of maximizing the switch time by balancing the high side path with the low side in the best possible way.

Chapter 6

Bibliography

[1] ControlAreaNetwork08.pdf

[2] Introduction to HIL Simulation.pdf

[3] SCALEXIOHardwareInstallationConfiguration.pdf

[4] SCALEXIOSystemOverview.pdf

[5] https://www.dspace.com/en/pub/home.cfm

[6] https://fscdn.rohm.com/en/products/databook/datasheet/discrete/transistor/mosfet/sh8m41-e.pdf

[7] https://www.onsemi.com/products/discrete-power-modules/mosfets/nvmfd6h840nl#technical-documentation

[8] https://www.mouser.com/pdfDocs/ANDLV8548MC-D.pdf

[9] https://it.rs-online.com/web/p/condensatori-ceramici-a-strato-singolo/2427534

[10] https://it.rs-online.com/web/p/condensatori-elettrolitici/1730277?gb=s

[11] https://www.infineon.com/cms/en/product/power/gate-driver-ics/ir2136s/#!documents

[12] https://it.rs-online.com/web/p/resistenze-a-foro-passante/7078782?

gb=s

[13] https://www.digikey.it/en/products/detail/infineon-technologies/IRF7380TRPBF/1928393

[14] https://docs.kicad.org/4.0/it/getting_started_in_kicad/getting_started_in_kicad.pdf

[15] https://freerouting.org/

[16] https://www.mouser.it/ProductDetail/STMicroelectronics/NUCLEO-F207ZG?qs=mKNKSX85ZJdRAguq1PNPkA%3D%3D

[17] https://www.st.com/en/development-tools/stm32cubeide.html

[18] https://www.ti.com/lit/ds/symlink/tms570lc4357.pdf?ts=1635107576594&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTMS570LC4357.

[19] https://it.rs-online.com/web/p/resistenze-a-foro-passante/6831049?gb=s

[20] https://www.hantek.it/oscilloscopi-da-tavolo/118-hantek-dso5102p-oscilloscop
html.

[21] https://it.rs-online.com/web/p/alimentatori-da-banco/1233578?cm_mmc=IT-PLA-DS3A-_-google-_-CSS_IT_IT_Strumenti_di_misura_Whoop+(2)-_-(IT:Whoop!)+Alimentatori+da+banco-_-1233578&matchtype=&aud-827186183686:pla-300610945169&gclid=CjwKCAjw-7OlBhB8EiwAnoOEk5NQPG4yNu-cROtLHRvBIG5Ptgv3egqnXFzKEdOUeChoCPskQAvD_BwE&gclsrc=aw.ds

[22] https://www.picotech.com/download/manuals/ta018-operators-manual-web.pdf.

[23] https://it.rs-online.com/web/p/diodi-schottky-e-rettificatori/4367868