UNIVERSITY OF LIEGE

Development of an embedded servomotor controller

Guillaume Lempereur

Supervised by Prof. Bernard Boigelot

Master thesis conducted for obtaining the Masters degree in Electrical Engineering by Guillaume Lempereur

Faculty of Applied Sciences
Department of Electrical Engineering and Computer Science

Academic year 2015-2016

UNIVERSITY OF LIEGE

Abstract

Faculty of Applied Sciences

Department of Electrical Engineering and Computer Science

MSc in Electrical Engineering

by Guillaume Lempereur

This project comes within the scope of the RoboCup[1] contest, an international robotics competition in which teams of robots play a soccer game. The motivation is to have a more effective control on robot joints than what can be achieved with commercially available technology. Those joints are made of servomotors, it is a rotary actuator that allows a precise control of angular position. The subject of this master thesis is to develop an electronic board that will provide advanced control mechanisms for a servomotor.

This board will receive orders from the outside via a communication bus and will control a DC motor in order to execute the order. The specificity of this work is that the orders may take many forms, from basic angular position, torque commanded motion to more specialized operations such as a sequence of command triggered by external events. The idea behind this project is to develop a board sufficiently powerful and modular to implement easily a wide set of handcrafted commands.

The servomotor Dynamixel$^{\text{TM}}$ MX-28 has been used as a basis, its electronic board has been removed in order to be replaced by a home-made one. The new board that was designed in the scope of this work improves and adds new functionalities to the previous one. The major improvements consist in a more accurate torque measurement, the support of CAN bus communication and the implementation of a more powerful microcontroller. The new functionalities consist in integrating a coupled accelerometer and gyroscope, a temperature sensor and an easy way to program the microcontroller by means of SWD or CAN bus.

The GL-28, name given to the board, has been tested and validated for some parts. Once fully tested, such a board will be embedded in servomotors that will be assembled and commanded to achieve a robot. Some people interested in this project would like to exploit such a servomotor for other purposes, opening new perspectives.

---

[1]http://www.robocup.org/

Contents

List of Figures

List of Tables

Abbreviations

AC	**A**lternative **C**urrent
ADC	**A**nalog to **D**igital **C**onverter
ADC	**B**ill to **O**f **M**aterial
CAN bus	**C**ontroller **A**rea **N**etwork bus
CAD	**C**omputer **A**ided **D**esign
DC	**D**irect **C**urrent
DMOS	**D**ouble **D**iffused **M**etal **O**xide **S**emiconductor
DMIPS	**D**hrystone **M**illion**I**nstructon **P**er **S**econd
DSP	**D**igital **S**ignal **P**rocessor
IMU	**I**nertial **M**easurement **U**nit
FIFO	**F**irst **I**n **F**irst **O**ut
GCC	**G**NU **C**ompiler **C**ollection
GPL	**G**eneral **P**ublic **L**icense
GNU	**G**NU is **N**ot **U**nix
I^{2}C	**I**nter **I**ntegrated **C**ircuit
JTAG	**J**oint **T**est **A**ction **G**roup
LED	**L**ight **E**mitting **D**iode
LQFP	**L**ow profile **Q**uad **F**lat **P**ackage
MCU	**M**icro**C**ontroller **U**nit
MIPS	**M**illion**I**nstructon **P**er **S**econd
NTC	**N**egative **T**emperature **C**oefficient
ODR	**O**utput **D**ata **R**ate
OSI	**O**pen **S**ystems **I**nterconnection
PCB	**P**rinted **C**ircuit **B**oard
PID	**P**roportional **I**ntegral **D**erivative

PWM	**P**ulse **W**idth Modulation
RAM	**R**andom **A**ccess Memory
RISC	**R**educed **I**nstruction **s**et Computing
RPM	**R**evolutions **P**er Minute
SIMD	**S**ingle **I**nstruction on **M**ultiple **D**ata
SRAM	**S**tatic **R**andom **A**ccess Memory
SMBus	**S**ystem Management **B**us
SMD	**S**urface **M**ounted **D**evice
SOIC	**S**mall **O**utline **I**ntegrated **C**ircuit
SPI	**S**erial **P**eripheral **I**nterface
SSI	**S**ynchronous **S**erial **I**nterface
SWD	**S**erial **W**ire **D**ebug
UART	**U**niversal **A**synchronous **R**eceiver **T**ransmitter

Physical Constants

$$\text{Gravity of Earth} \quad g \quad \simeq \quad 9.81\,\mathrm{m\,s^{-2}}$$

Symbols

V	voltage	V ($\mathrm{kg\,m^2\,A^{-1}\,s^{-3}}$)
I	current	A
R	resistance	Ω ($\mathrm{kg\,m^2\,A^{-2}\,s^{-3}}$)
P	power	W ($\mathrm{kg\,m^2\,s^{-3}}$)
ω	angular frequency	$\mathrm{rads^{-1}}$

Chapter 1

Introduction

This project is aimed at developing an electronic board that will be embedded into a servomotor. Beside the usual purpose of receiving instructions and controlling a motor in order to execute the order, many new and special functionalities are added to enhance the device. The motivation that drives this development is the participation to the RoboCup[1] contest. RoboCup is an annual international robotics competition in which robot teams play soccer games. Figure 1.1 depicts a participant robot that is able to move thanks to servomotors that mimic human joints. They represent a big part of its body and are the only way to move and actuate the robot. There can be no doubt that having the best possible control of the joints or at least an improved version of what can be available on the market is a real asset for the contest. It is with this perspective that this project has been carried out.

1.1   Servomotors

Considering the etymology of the word servomotor, one can provide a first simple definition. It comes from the Latin `servus` (slave) and motor, that is to say a servomotor is a motor fully controlled. A servomotor is a rotary or linear actuator that allows a precise control of angular or linear position, as well as sometimes velocity or torque.

---

[1]http://www.robocup.org/

1
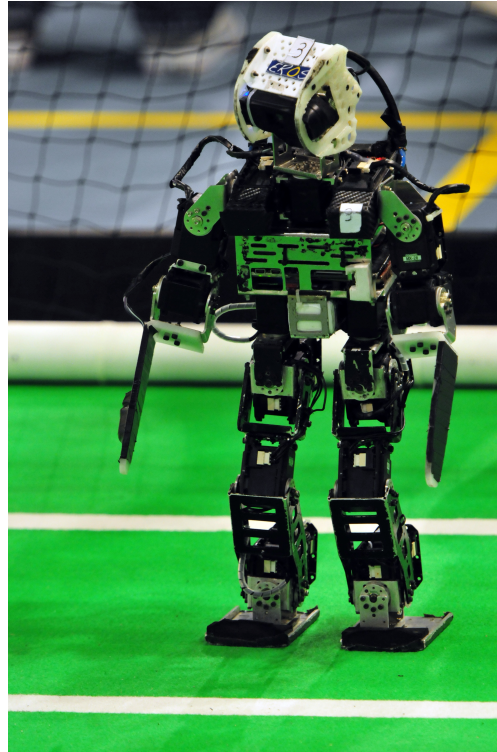
FIGURE 1.1: A Robocup humanoid made of assembled servomotors (by Ralf Roletschek) [1].

1.2 Principles of operation

A servomotor mainly consists of a suitable motor connected to a reducer that reduces the rotation speed while increasing the torque of its output shaft. The latter is coupled to a sensor for measuring the position, this information is provided to a control unit that commands the driver of the motor in order to achieve the command that was sent to the servomotor. In Figure 1.2, on can see a basic diagram illustrating the working principle. The following paragraphs explain how the class of servomotors used in this project operate.

The servomotor has a microcontroller that listens to a communication channel for any orders such as moving to a given angular position. When it finally receives the command, it executes an algorithm to reach the goal. The commands may be to move to an angular position, to rotate at a certain speed, to maintain a given torque, or may be more advanced. Protections are in place for limiting the current consumed by the motor, its temperature, the rotation speed, in order to avoid any damage to the motor or the electronic board.

The only actuator is a direct current (DC) motor commanded by a variable power supply. Such a motor may be controlled in many ways, for instance with respect to its speed or torque. One may want it to stay at a position in blocking mode, even if an external torque is exerted on it or to brake a potential applied torque. To achieve such a control, one has 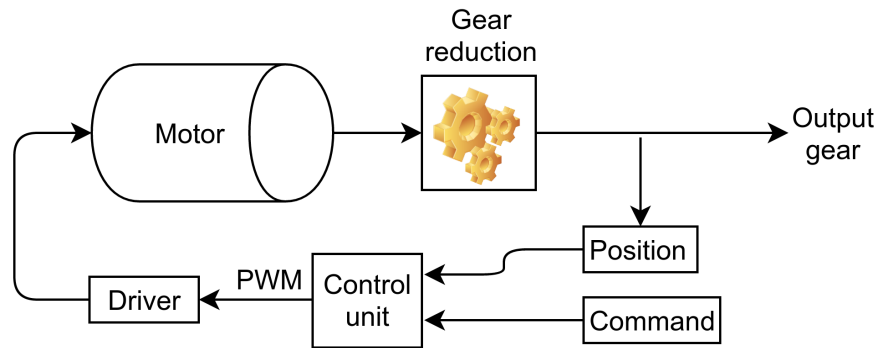to be able to deliver a current in both directions to the motor. For a torque control, the common way is to regulate the current flowing through the motor, allowing it to rotate at any achievable speed while reaching the desired torque when it is possible. In the same way, speed control is usually done thanks to a current regulation allowing the motor to provide the necessary torque to possibly reach the desired speed.

The DC motor used in this project turns at a speed of roughly 8000 RPM at nominal operation but the desired speed is usually less than a turn over a second, so the servomotor is equipped with a gear reducer. Gear reduction allows to divide the rotation speed while multiplying the torque by the same factor, called the reduction factor.

The servomotor is generally powered by a DC supply provided by a battery. There is a power electronic circuit in charge of providing adjustable current to the motor in order to execute orders. A common solution for this problem is a H-bridge, which is a circuit that enables a voltage to be applied across a load in either direction. A H-bridge is shown in Figure 1.3; it is composed of four transistors acting as switches. There are four signals for the gates that allow to open or close the switches. So there are 16 combinations for the H-bridge that allow to drive the motor in different ways. Two usual configurations are depicted in Figure 1.4, on the left is the reversing circuit where the high potential is connected to the right terminal of the load and the ground to the left terminal. On

the right is the forwarding circuit that is the just the opposite of reversing, the high potential is connected to the left terminal and the ground to the right.
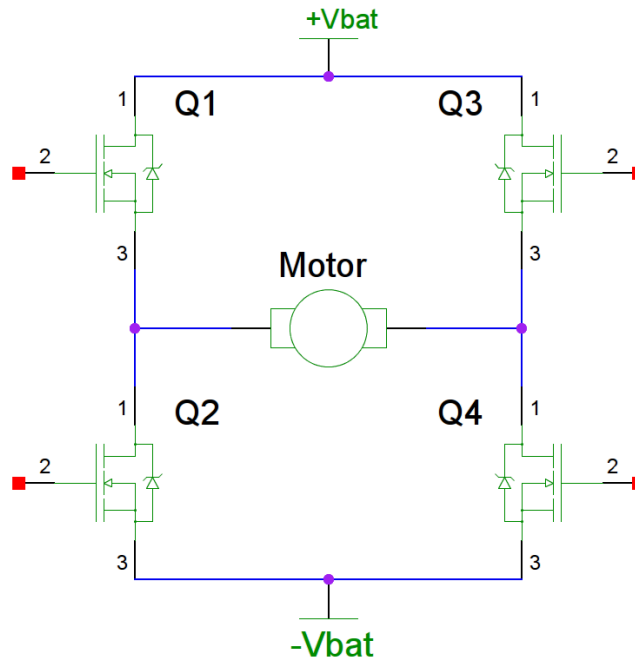
FIGURE 1.3: H-bridge.

There are 7 states that should never occur when the bridge is powered, corresponding to a short circuit when both transistors at the same branch are closed, causing the high potential to directly flow to the ground without going through the load. A less dangerous state is when both transistors connected to the same potential are closed while the two others are opened. It means that the load is short-circuited and in the case of a motor, it sets this motor in braking mode because the current generated by the driven motor flows in a very low impedance loop that dissipates by Joule effect the produced energy. Finally the last configuration is when at most one transistor is closed, which leads to a disconnected load, so in the case of a motor, this means that the motor will coast. Those three states are represented in Figure 1.5. The table 1.1 resumes every possible state for the H-bridge, the dash standing for any switch state (open or closed).

Finally, in order to regulate the power delivered to the motor, the gate signals will not be constant but driven by a PWM (pulse-width modulation). Such a signal, represented in Figure 1.6, is an alternation of ON/OFF states characterized by the ratio of the ON time over the period called duty cycle. This kind of signal is convenient to drive a H-bridge at a fraction of its maximal power because it lowers the losses in the transistors.
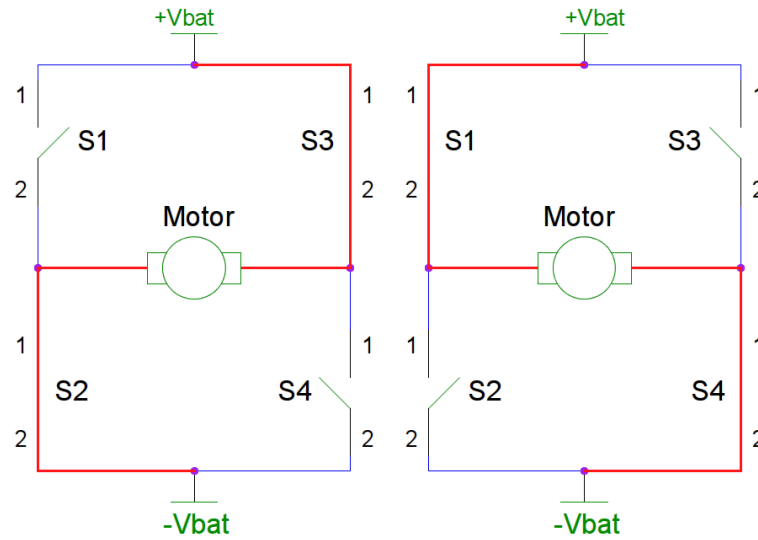
FIGURE 1.4: H-bridge in revering (left) and forwarding (right) configuration.
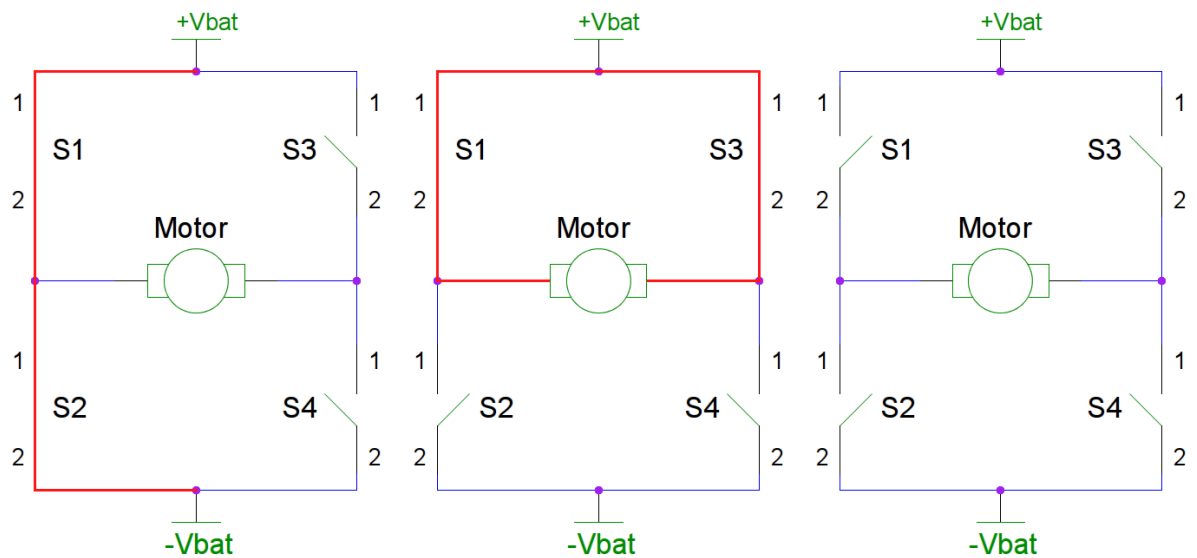
FIGURE 1.5: H-bridge in other configurations. From left to right: short circuit, braking, coasting

To achieve this regulation, one could consider four PWM signals that would alternate between the forwarding/reversing and the coasting state. In such a configuration the motor is powered only a fraction of the time and is otherwise coasting, so that the mean power provided to the motor can be regulated to achieve what is desired. This approach leads to an issue when dealing with an inductive load as a DC motor because during the forwarding phase a relatively important current flows through it and when comes the second phase that disconnects completely the motor, there is no path for the current

Switch 1	Switch 2	Switch 3	Switch 4	State
closed	open	open	closed	Forwarding
open	closed	closed	open	Reversing
closed	open	closed	open	Braking
open	closed	open	closed	Braking
closed	closed	-	-	Short circuit
-	-	closed	closed	Short circuit
-	open	open	open	Coasting
open	-	open	open	Coasting
open	open	-	open	Coasting
open	open	open	-	Coasting

TABLE 1.1: H-bridge states.

to flow back so the voltage across the transistor will increase until it will eventually damage it. A solution is to add a reversed biased diode (called a "flywheel" diode) in parallel to the transistor so that the current can flow through this diode when the motor is disconnected. This solution is used in some applications, but has the drawback that the current that flows through the diodes becomes relatively high, which leads to high losses.

To overcome this situation, there are two techniques to drive the H-bridge, the phase-magnitude and the locked anti-phase [19]. The phase-magnitude drive consists in alternating between the forwarding/reversing and the braking state. In this way, there is always a path for the current flowing through the motor that goes through the transistors. The second technique called locked anti-phase drive is an alternating of the forwarding and reversing states. The second technique is preferred because it allows to regenerate the battery when the motor is driven by an external torque. By using the phase-magnitude technique, the generated power would be dissipated by Joule effect during the breaking phase. This is why the developed electronic board will implement locked anti-phase drive.
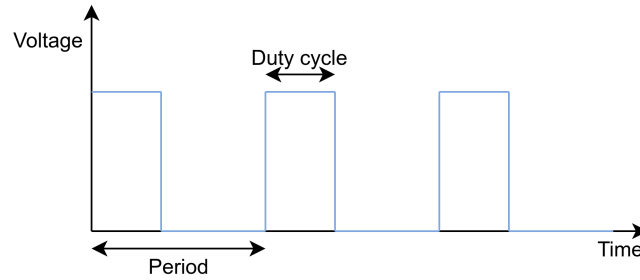
FIGURE 1.6: A PWM signal.

The angular position of the output shaft is measured using a sensor. When the command for the servomotor consists in moving to a given angular position, the microcontroller computes a duty cycle that takes into account the currently measured position. This process is executed in a loop until the servomotor eventually reaches the goal. There are many strategies that can be implemented for performing this task. A commonly used method is to use PID control (proportional integral derivative). When its parameters are fine tuned, this controller is sufficiently powerful for most purposes.

1.3   Project statement

As already explained in the introduction, the project consists in developing an electronic board that controls a servomotor. The board is designed to fit in the servomotor MX-28 from Dynamixel$^{\text{TM}}$, so that one just has to replace the MX-28 board with ours to obtain an improved servomotor. Basically the goal of the project is to build an electronic board that controls a servomotor in programmable way, in such a way that the possibilities of control are enhanced.

Almost every servomotor can be controlled in position, meaning that its purpose is to reach a certain angular position of its shaft. More advanced servomotors are able to be controlled in torque. Another possibility is to use speed control where the torque produced varies in order to reach the desired speed. For this project, one of our goals is to be able to vary the control strategy over time, e.g. the command would be a set of positions to reach at given times. Another type of control that should be achieved is a sequence of predefined controls as positions or torques that are performed according to events that trigger them. For instance, controlling the robot leg to kick the ball could

be implemented in two phases: first, a position control to reach the ball and then in a second phase torque control for regulating the force applied to the ball. The transition between these controls could be triggered by the shock due to the contact of the ball on the foot of the robot.
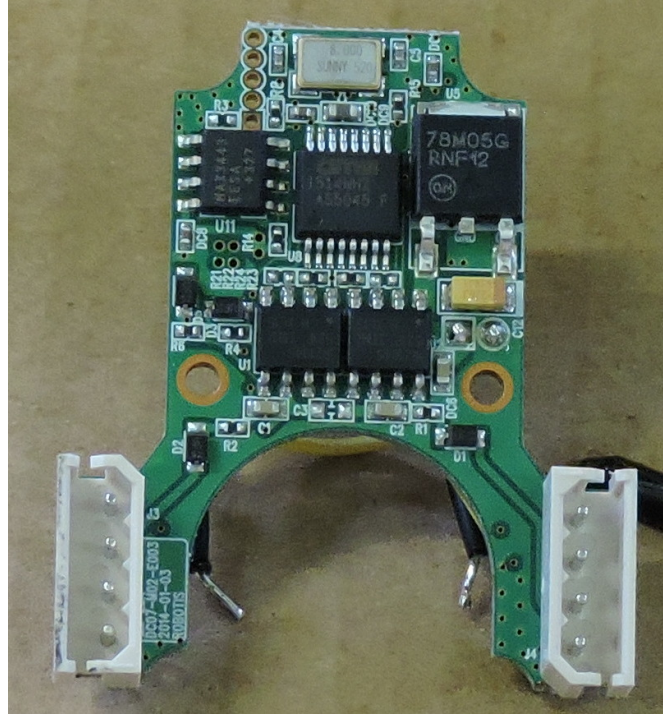
FIGURE 1.7: MX-28 electronic board.

Such improvements require a better communication infrastructure between the servomotor and its master command module. Since orders may be complex, one has to assume a reliable communication channel with a capacity large enough. Potential big messages carried at high speed means it is even more important to be immunized against noise.

In order to implement sophisticated command strategies, the microcontroller has to be powerful enough to run the algorithms. It has to perform basic arithmetic calculations such as additions of 32-bit integers, floating-point operations and so on. It should have enough RAM to run quite heavy programs, a sufficiently high clock frequency to be able to run algorithms that require fast execution. The flash memory should also be large enough to contain large programs that include, for instance, big look up table. Ideally, the microcontroller should be provided with libraries allowing common functionalities such as I^2C communication, PWM signal generation, etc.

Obviously, the microcontroller need to be interfaced to a sensor to measure the position of the external gear. The servomotor has to be able to turn over $360°$ and still be able to measure the position for any angle. The measurement has to be precise of course, and quick enough to follow the motion of the output shaft.

Information about the motion of the servomotor in space, acceleration experienced, etc are desired to be acquired in order to enhance the control and to provides useful information to the master of the servomotor. For this purpose, a gyroscope and an accelerometer are embedded in the electronic board. Their own purpose is to provide respectively the angular velocity and acceleration experienced by the servomotor, but combined it becomes an Inertial Measurement Unit (IMU) that allows to measure orientation, speed and relative position in space.

In order to control the motor according to torque, one has to know accurately the current consumed by the motor. Since the motor becomes a generator if the driven leads the driver, the current that has to be measured may be positive or negative. The supply voltage of the motor also has to be measured so as to improve control or to detect a voltage drop of the power supply.

A useful new feature is the measurement of the temperature of the motor by means of a probe glued on it; this information allows to have a better monitoring, so that the motor would be protected if an overheating is detected.

Finally, the MX-28 board has also a LED aligned with a light guide that allows to debug the servomotor in various situations. For practical and aesthetic reasons the monochrome LED is replaced by a tricolor one.

Chapter 2

Servomotor analysis

In this chapter, we analyze the internal components and the details of operation of two digital servomotors available on the market. At the end of the chapter, there is a conclusion comparing both servomotors and the reasoning that leads to selecting the servomotor that will be used as a basis for this project. The analyzed servomotors are the RX-28 and MX-28 from Dynamixel^{TM}, they are highly sophisticated and well suited for robotic applications. They have the ability to track their speed, temperature, angular position, voltage, and load. There are some configurable parameters that adjust the control algorithm, allowing to control the speed, strength and accuracy of the motor's response.

2.1   RX-28

The RX-28, shown in Figure 2.1, achieves a 3.77 N m torque when it is powered at 16 V and 2.83 N m at 12 V. The RX-28 is limited to a range of 300° for precise positioning, but there is a mode allowing to perform endless turns when the servomotor in controlled in speed.

The RX-28 embeds a 8-bit microcontroller `ATmega8` from Atmel^{TM}, based on the AVR RISC architecture. It achieves throughputs approaching 16 MIPS at a maximal clock frequency of 16 MHz, meaning that it is able to execute an instruction over one clock period, so 16 million instructions in one second. It has a 8 KB flash memory and 1 KB of SRAM.

FIGURE 2.1: RX-28 servomotor (Source: Trossen Robotics^{TM}) [2].

The angular position of the shaft is measured through a potentiometer, which is not mechanically bounded, so that it allows endless turns. One drawback is that the potentiometer has a linear resistance only for a 300° interval and is opened for the other positions. For this project, such measurement system is inadequate because of the angle limit. Nevertheless, the resolution of a position measured via this potentiometer and the ADC converter of the microcontroller is 0.29° which is sufficient for most applications.

In order to control the motor, especially for torque control, the current flowing through the motor has to be measured. The RX-28 servomotor uses a simple resistor connected in series with the motor. The voltage across this known resistor is measured via an analog input of the microcontroller, which allows to determine the current. This method is the cheapest but suffers form resolution issues as explained in Section 3.6.

At the communication level, the servomotor uses asynchronous serial communication following the standard RS-485. This device achieves a speed up to 1 Mbps and uses the `MAX485` chip as transceiver.

The RX-28 is actuated by a `RE-Max 17 214897` motor from Maxon^{TM}, shown in Figure 2.2. It is a 4 W direct current brushed motor that is 17 mm in diameter and 25 mm long. The most interesting data about this motor is reported in Table 2.1. The reduction gear has a ratio of 193:1, meaning the output gear turns 193 times slower and the output torque is 193 times higher than the output shaft of the motor.

The H-bridge driving the motor is made of the `L6201` chip, a DMOS full bridge driver. It contains everything that is needed to realize a H-bridge, including the four transistors

Nominal voltage	12 V
Nominal speed	8110 RPM
Nomimal torque	3.67 mN m
Stall torque	15.5 mN m
Starting current	1.45 A
Nominal current	350 mA

TABLE 2.1: RE-Max 17 motor specifications.

FIGURE 2.2: RE-Max 17 motor (Source: Maxon motor^{TM})[3].

and their drivers. The component is driven by three control signals, one of which being for enabling the device. The two others are for connecting their side of the bridge either to the ground or to the reference voltage. The supply voltage, the max peak current and the RMS current can go respectively up to 48 V, 2 A and 1 A. The drain to source resistance of the bridge ($R_{DS(ON)}$) is 0.3 Ω. This value characterizes to the losses, so the performance of the H-bridge.

2.2 MX-28

The second device studied is the MX-28 servomotor displayed in Figure 2.3. It belongs to the next generation of servomotors which brings more advanced control algorithms than in the RX-28 model. It develops a torque of 3.16 N m at 14.8 V, 2.55 N m at 12 V and 2.34 N m at 11.1 V, so it is a bit weaker than the RX-28. They share the same dimensions and the same communication protocol for the MX-28R model, but quite unexpectedly, we discover that its electronic design is very different from the RX-28 servomotor.

FIGURE 2.3: RX-28 servomotor (Source: Trossen Robotics$^{\text{TM}}$) [4].

Firstly, the microcontroller equipping the MX-28 servomotor is the `STM32F103` from ST$^{\text{TM}}$ that incorporates the high-performance `ARM Cortex-M3` 32-bit RISC core operating at a frequency of 72 MHz. It is a much more powerful MCU than the 8-bit Atmel$^{\text{TM}}$ embedded in the RX-28. Its throughput is up to 72 MIPS and 90 DMIPS. (The Dhrystone Million Instruction Per Second (DMIPS) is an unit that is supposed to evaluate the performance of a MCU while being architecture independent. It is based on the time taken to execute a particular benchmark program that contains integer operations.) The flash memory is also larger, up to 32 KB, as well as the SRAM memory whose size is 10 KB.

The servomotor measures the position of its output shaft using a magnetic sensor that measures the magnetic field induced by a magnet glued to the shaft. In Figure 2.4, one can see an illustration of this kind of sensor. This method has several advantages compared to a potentiometer: it can measure any angle, allows endless turn and does not require physical contact between the shaft and the measuring component. The resolution is better than the one obtained with a potentiometer, it is now 0.088°. The Austriamicrosystems AG$^{\text{TM}}$ `AS5045` chip ensure the measurement, it is a 12-bit programmable magnetic rotary encoder. It has two modes to communicate the angle: either as a PWM output signal, or by a synchronous serial communication.

The motor is driven via a H-bridge realized with four chips, it is divided into two mirrored branches containing each a driver and a dual transistor. The International Rectifier$^{\text{TM}}$ `IR2104S` is a half-bridge driver used for driving the transistors. The Fairchild$^{\text{TM}}$ `FDS6990A` is a dual N-channel MOSFET in which two transistors are placed one over the other
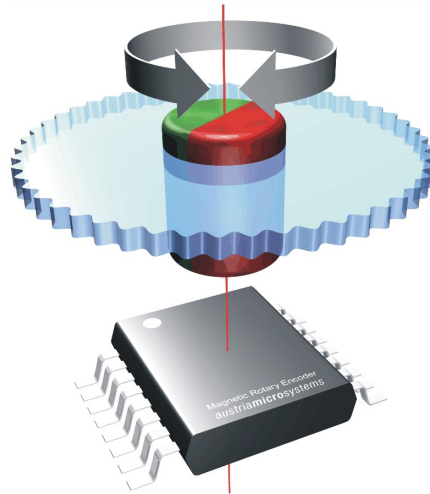
FIGURE 2.4: AS5045 magnetic rotary encoder (Picture by Vince Biancomano) [5].

in order to form half of the H-bridge. The drain to source resistance $(R_{DS\,(ON)})$ of the bridge is only $36\,\text{m}\Omega$, so one can conclude that this implementation is far better in terms of losses than the one employed in the RX-28.

The communication between MX-28 servomotors and their controller is identical to that of RX-28 components, both at the protocol and performance levels. The transceivers is this time the Maxim$^{\text{TM}}$ `MAX344EESA`. The motor is also identical to the one used in the RX-28, as well as the reduction gear. The current sensing is also achieved via a resistor and the ADC of the microcontroller.

2.3 Conclusion

Both RX-28 and MX-28 servomotors are identical in their mechanical components, besides the component used for measuring the angular position for which the RX-28 has a potentiometer and the MX-28 has a magnetic sensor. The electronic boards are completely different and more elaborated for the MX-28, leading to a more accurate and efficient operation.

The very first step of this project is the selection of the servomotor that will serve as basis to our own development. Following the analysis made in this chapter, our choice is to opt for the MX-28 because it allows precise position over a complete revolution and its resolution is the best. It is finally worth mentioning that Hubert Woczszyk, another

student working on the simulation of the robot, has determined that the servomotor MX-28 has a high enough torque and speed for our intended application [20].

Chapter 3

Electronic design

This chapter presents the electronic design of our servomotor board. It is divided in several sections corresponding each to a specific part of the electronic circuit. The sections describe the problem to be solved by this part and the reasoning that leads to the adopted solution.

3.1 Mechanical components

The project only focuses on the development of an electronic board that will be embedded in a particular servomotor, the MX-28 from Dynamixel$^{\text{TM}}$. The motor equipping the MX-28 servomotor provides enough torque since the MX-28 has been evaluated as suitable for our intended application [20], so keeping the same motor is the cheapest and most reasonable choice. The same reasoning applies on the gearbox given that the motor remains the same and the angular speed is adequate.

The electronic board is very small and has many mechanical constraints that makes the design of the board extremely difficult. Nearly all the following choices are partially or totally motivated by the space occupied by the component.

3.2 H-bridge

Considering Section 1.2, to drive the motor one has to implement a H-bridge. Among the cited techniques to drive the bridge, it is the locked anti-phase that will be used

because of its advantage over the phase-magnitude strategy. So the signals that will command the transistors on the left side of the bridge will always be in quadrature with the signals that will command the right side, with the exception of free-running mode, in which all the signals are low in order to open each transistor.

As often for this type of project, the major problem that leads to the final choice is the space that is occupied by the device. This is the reason why we can not use a chip such as L6201 that embeds a complete H-bridge, since this 13 by 10 mm component would take nearly half a side of the board. Plus, its efficiency is worst compared to what the MX-28 H-bridge can achieve, since the drain to source resistance of its MOSFETs is 10 times higher.
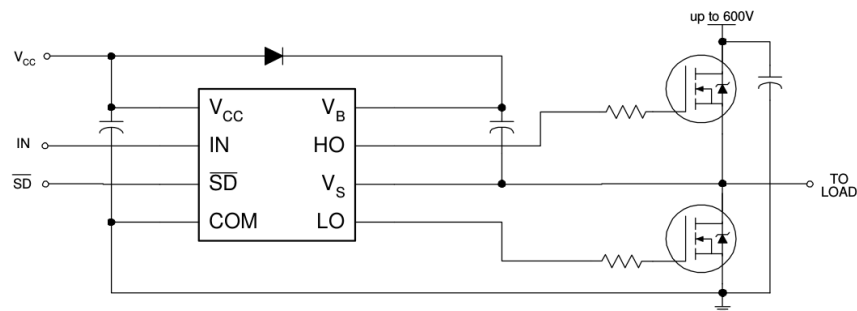

FIGURE 3.1: Half of the H-bridge (Source: International Rectifier$^{\text{TM}}$)[6].

So the chosen design is made of four chips: two dual transistors and two half bridge drivers. Even though the number of chips is higher, it allows to place them more freely on the board and the total space occupied by the H-bridge is lowered. In Figure 3.1, one can see the schematic of half of the bridge. The aim of the driver is to provide a voltage across the gate and the source of the transistors large enough to close them completely, so in other words it is used for amplifying the control signal coming from the MCU. It is difficult to polarize correctly the upper transistor because its source reference fluctuates when the bridge is in forwarding or reversing mode. The solution is to add a pump charge that accumulates some energy during the phase where the lower transistor is closed and releases that energy to power the upper transistor during the second phase. It means that to drive a H-bridge with such a driver that requires charge inflow for the pump, the PWM signal can not be close to 0 % or 100% duty cycle. A danger that has to be absolutely avoided is to close the upper and lower transistors at the same time,

because it would short circuit the power supply. The driver avoids this problem by using an internal inverter that always opens one transistor while it closes the other.

Considering the choice of the components to design the H-bridge, it seems reasonable to keep those of the MX-28. The dual transistor `FDS6990A` is well suited for battery powered applications in which low in-line power loss and fast switching are required, as it is the case in this project. For the driver, the `IR2104S` chip is also well suited for this H-bridge.

3.3 Microcontroller

Due to the popularity and some knowledge already acquired with its architecture, the microcontroller `STM32F303CC` has been used. This is a powerful chip that can work at 72 MHz and 90 DMIPS. It belongs to the same family `STM32F` as the microcontroller used by the MX-28 servomotor.

The first interesting thing about this chip is that it is based on an `ARM Cortex M4`, which contains many interesting functionalities. The `Cortex-M4` processor has been designed with a large variety of highly efficient signal processing features. The processor features extended single-cycle multiply accumulate instructions, optimized SIMD arithmetic, saturating arithmetic instructions and an optional single precision Floating Point Unit. These features mean that quite complicate and high resource consuming programs can be executed.

Finally the very determinant point that leads to this particular ST$^{\text{TM}}$'s microcontroller is that the available place on the board is very restricted. This `STM32F303CC` microcontroller is currently the most powerful one available in a 7 by 7 millimeters 48-pin package. The flash memory is 256 KB and the RAM is 48 KB. In comparison, for the same package, the best MCU from Texas Instruments$^{\text{TM}}$ is the `Piccolo` that has a frequency of 60 MHz. The same goes for Atmel$^{\text{TM}}$'s `AT32UC3` that achieves only 60 MHz.

3.4 Communication

Since the servomotor is meant to receive orders from an external controller, it requires a powerful communication system. Any devices that communicate on a channel have to speak the same language, so there is a need to implement a protocol that any device sharing the channel will have to follow.

Contrarily to usual servomotors, the communication will not be restricted to simple orders, in particular in order to implement complex control strategies that may rely on massive amounts of data and because there is an accelerometer and a gyroscope that will need to send information. These are good reasons to consider another protocol than the one used by the RX-28 and the MX-28.

3.4.1 Protocol

First of all, we have considered the Open Systems Interconnection model (OSI) to develop an early choice for the communication system. The OSI model is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to their underlying internal structure and technology.

The first layer is the physical layer, for which many technologies can be considered. A wireless communication has the obvious advantage that it does not require any cables but an antenna. Such a system is not appropriate for this project because the servomotor is not self powered via an internal battery, so there will be at least two mandatory wires for providing the power. In the wired category, one may consider optic fiber, which is undoubtedly the fastest way to communicate. For this project, there is no need to have such technology which adds unnecessarily expenses and complexity. The same argument to a lesser extent, goes for coaxial cable. The most reasonable and cheapest choice seems to be a communication based on copper-wire twisted pairs.

Once the physical channel has been selected for transmitting the data, the question then becomes to select the best suited protocol for this project. The servomotors will be assembled in order to form an arm, a leg or any other parts of the humanoid robot. This influences the choice made for the communication protocol. Indeed, a parallel

communication would require too much wires, especially when several servomotors are combined because they all have their own wires. A better solution would be to use serial communication over a bus.

Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel. One immediately understands that if the information is carried over a single wire, then the physical size of the channel will remain low. The second important component coming with the serial communication is the availability of using a bus as communication channel. A bus is defined as a communication channel shared between many devices in which some rules allow each entity to communicate correctly to its target.

Before studying the protocols, there are three important concepts to be defined in order to characterize the communications. The first is multiplexing, that allows to connect many devices on a single channel, and its reciprocal, the demultiplexing, that allows to select the device that will receive the message. The second concept is the property of a communication to be unidirectional or bidirectional. In Figure 3.2, one can see the three possibilities. Full duplex means that both parties can communicate simultaneously, while half-duplex means that they can only communicate one at a time. A simplex communication stands for a simple master sending information to a slave that can not respond at all. The third qualifier is the synchronicity, a communication is said to be synchronous if there is a clock signal that synchronizes the communication between the parties.

There exists a huge number of communication protocols that are based on the previous requirements. Since creating a new communication system is not part of this project and there are not any particularities in the messages that will be exchanged justifying to do so, only the most popular solutions have been studied.

One can enumerate the following well known communication systems and protocols:

- UART (Universal Asynchronous Receiver Transmitter)

- SPI (Serial Peripheral Interface)

- SSI (Synchronous Serial Interface)
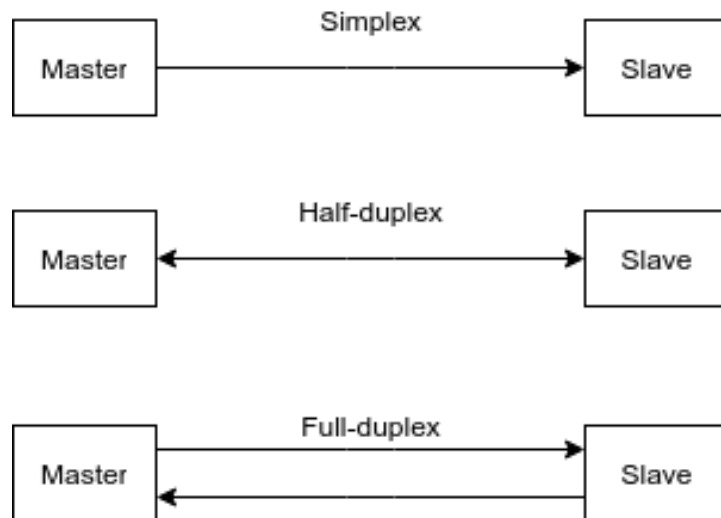
- I^{2}C (Inter-Integrated Circuit)

FIGURE 3.2: Half-duplex, full duplex and simplex communication.

- CAN bus (Controller Area Network Bus)

The SPI bus, illustrated in Figure 3.3, is used for short distance communication, primarily in embedded systems. This interface was developed by Motorola$^{\text{TM}}$ and has become a de facto standard. It works with four wires:

- SCLK: clock signal

- MOSI: Master Output $->$ Slave Input

- MISO: Slave Output $->$ Master Input

- SS: Slave select

The main advantage of SPI is the full duplex communication that allows a slave and its master to exchange information at the same time. The main drawback is that it requires a SS signal for each slave that is connected to the bus. It means that, for instance, a leg composed of 5 servomotors need 8 wires.

The SSI interface is based on the RS-422 standard, which implies that the signals are differential and carried over twisted pairs. As it can be seen in Figure 3.4, the signals are clock and data, so it implements simplex and non-multiplexed communication. The main advantage is that data can be transmitted over long distances, up to 1.2 km, and it has a high electromagnetic interference immunity. The main drawback for this project is
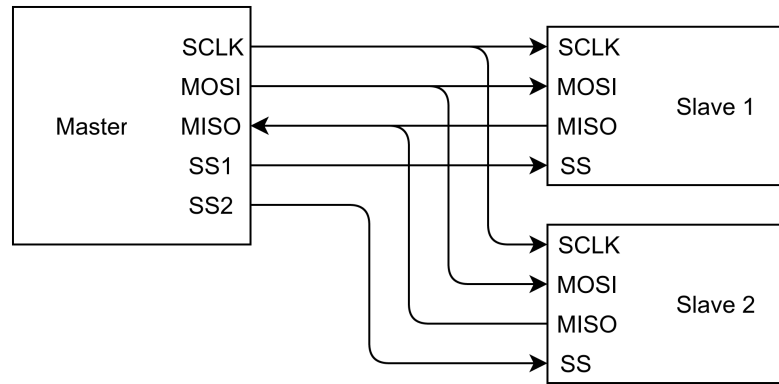
FIGURE 3.3: SPI protocol.

the simplex communication, which makes it impractical to send data such as acceleration measurements from the servomotor to the host.
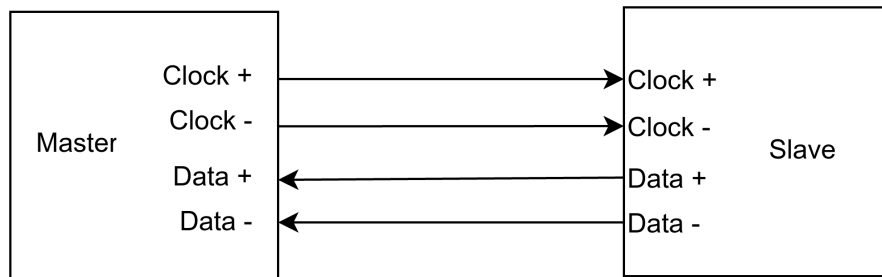

FIGURE 3.4: SSI protocol.

The UART is a very simple asynchronous communication device between a master and its slave. It require only one wire on which data is sent in a frame that usually starts with a start bit and ends with two stop bits, see Figure 3.5. Again multiplexing is not available, so it is not suited for this project.
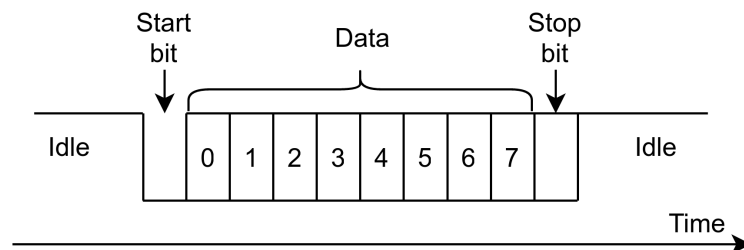

FIGURE 3.5: UART protocol.

I^2C is a multi-master, multi-slave, single-ended, serial computer bus invented by Philips$^{\text{TM}}$.

It uses two wires SDA and SCL, the first for the data and the second for the clock signal. As depicted in Figure 3.6, it uses two bidirectional open-drain lines pulled up with resistors, meaning that when no signal is sent on the line, one reads a high level. The reference design has addresses of 7 or 10 bits allowing to interconnect a large number of devices. It achieves speed of 100 kbit/s in Standard mode, 400 kbit/s for fast mode, up to 1 Mbit/s and 3.4 Mbit/s for Fast mode plus and High Speed mode. I^2C has a smart arbitration mechanism that allows several masters on the bus, which is very interesting because with only 2 wires, many devices are able to talk to each other without any problem. The biggest drawback is that the two signals are not differential which is a bad thing for noise immunity and emission, especially when the clock speed is high.
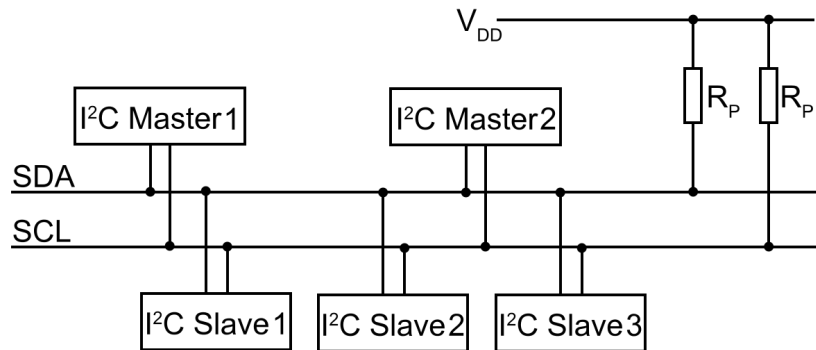

FIGURE 3.6: I^2C protocol (Source: DNL ware)[7].

Finally the last standard considered is the CAN bus which is an ISO standard computer network protocol and bus standard, designed at first for industrial networking but adopted more recently by automotive applications [21]. It uses two wires that carry a differential signal, which contains data. A transceiver is added to make the link between the microcontroller and the bus. On this device, there are two inputs, RX and TX, one for the data to be sent and the other for the data being received. It achieves a speed up to 1 Mbit/s and is especially very robust in extremely harsh environment thanks to its differential bus. All these considerations lead to the final choice, of CAN bus for our communication channel.

The transceiver `MAX3051` from Texas Instruments$^{\text{TM}}$ working in 3.3 V is the chip employed to link the MCU to the bus.

3.5 Inertial measurement unit

This is a new functionality compared to the MX-28 servomotor, so it means we have to find some space for the IMU in an electronic board that does not correspond to MX-28 components. The IMU consists of an accelerometer and a gyroscope, one of our requirement is to find a component in which both device are embedded in one chip for space reasons. The second advantage of having only one chip is that they will share the same communication channel, which simplifies the interconnection with the MCU. This is also a drawback because it is impossible to read the acceleration and angular speed at exactly the same moment. Because the data provided by the IMU is not the crucial part of the project and small delays can be tolerated, this problem is not critical at all.

FIGURE 3.7: FIS1100 inertial measurement unit (Source: Digi-Key electronics$^{\mathrm{TM}}$) [8].

There are not so many coupled accelerometer gyroscope on the market and since the technology evolves with time, we have only studied completely two chips. The `FIS1100` from Fairchild$^{\mathrm{TM}}$, represented in Figure 3.7, is a six degrees IMU embedding a triple axis accelerometer ranging from ± 2 g to ± 8 g and a triple axis gyroscope ranging from $\pm 32° \, \mathrm{s}^{-1}$ to $\pm 2,560° \, \mathrm{s}^{-1}$. The noise is $50 \mu g / \sqrt{Hz}$ for the accelerometer and $0.01°/\sqrt{Hz}$ for the gyroscope. It communicates via SPI or I^{2}C and allows to add a magnetometer to the bus to obtain a 9 degrees of freedom IMU. It has a large 1.5 kB FIFO queue that can be used as a buffer before sending the data to the MCU. The main advantage of this device is that it contains an advanced vector digital signal processor (DSP) allowing to encode high frequency motion at high sampling rates, up to 1 kHz, while having a low output data rate (ODR). This means that the IMU can exchange at low speed data with the MCU (low ODR) and still acquire accurate 3D motion data.

The `LSM6DS3H` from ST$^{\mathrm{TM}}$, shown in Figure 3.8, is the second IMU studied. It has a triple axis accelerometer ranging from ± 2 g to ± 16 and a triple axis gyroscope ranging from

FIGURE 3.8: LSM6DS3H inertial measurement unit (Source: Digi-Key electronics[TM])[9].

$\pm 125° \text{s}^{-1}$ to $\pm 2,000° \text{s}^{-1}$. The noise for the accelerometer is larger than the first device $(90 \mu g / \sqrt{Hz})$ and slightly lower for the gyroscope $(0.007° / \sqrt{Hz})$. It communicates also via SPI or I²C, can handle a magnetometer, and has a bigger FIFO of 8 kB. Its strength is its output data rate that is up to 1.6 kHz which is the best figure available on the market. It does not have an internal DSP like the `FIS1100` but this does not matter because the MCU is powerful enough to process the data and provide an accurate 3D position. The second advantage is that this component is made by the same company as the MCU, which provides libraries that exploits all the potential of the device. This is why the `LSM6DS3H` is the final choice.

3.6 Current sensing

A big improvement of our electronic board over the existing MX-28 is to have a better current sensing, leading to a better measurement of the torque developed by the servo-motor. A common way to measure a current is simply to measure the voltage across a known resistor. Indeed, according to Ohm's law (3.1), the voltage (V) is proportional to the current (I), the constant proportionality being the resistance (R).

$$ I = \frac{V}{R} \iff V = RI \tag{3.1} $$

$$ P = VR \iff P = RI^2 \tag{3.2} $$

It means that the current has to flow through the resistor, which introduces some losses. One can see according to Joule effect (3.2), that losses are also proportional to the resistance. So to limit them, one has to choose the smallest possible resistance. Such choice introduces another problem, the voltage across this resistor will be small according

to (3.1). As a conclusion, the initial task that was measuring a current is now achievable using the ADC converter of the MCU but has been turned into is a harder problem that is to measure a small voltage. There exists many techniques based on voltage sensing to measure the current, but only three different ways have been studied because the available place is very limited on our board and only those three could potentially be implemented.

Before studying these methods, one may consider another way to measure the current, relying on the Hall effect. The principle of the Hall effect is illustrated in Figure 3.9. Initially there is an electrical conductor that is subjected to a magnetic field $\vec{B}$ oriented towards the top and a constant current I that flows from the front to the back. An electrical current implies that there is a movement of many small charge carriers inside the conductors along the axis of the current flow. According to Lorentz law expressed by Equation (3.3), a moving charge in the presence of electric ($\vec{E}$) and magnetic ($\vec{B}$) fields experiences a force. It means that the current I exerts a force on the charges along the z axis but also that the magnetic field $\vec{B}$ applies a force perpendicular to itself and to the current flow, along the x axis. The paths of charges between collisions are curved so that they accumulate on one face of the material. This results in the appearance of an electric field $\vec{E_h}$ that opposes the migration of further charge, so a steady electrical potential is established and can be measured in order to estimate the value of the current I.
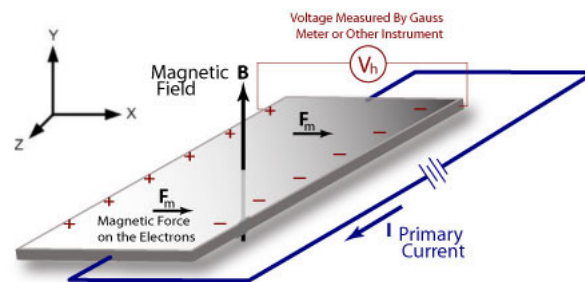

FIGURE 3.9: Hall effect principle (Source: NDT Resource Center)[10].

$$\vec{F} = q\vec{E} + q\vec{v} \times \vec{B} \tag{3.3}$$

The `ACS714` chip from Allegro$^{\text{TM}}$ is a current sensor based on the Hall effect, with an analog output that produces voltage is proportional to the measured current. A great

advantage of this solution is the very low resistance $(1.2\,\text{m}\Omega)$ that is nearly negligible. The drawback of this chip is that it is only available in a 8-pin package that is quite large compared to other solutions.
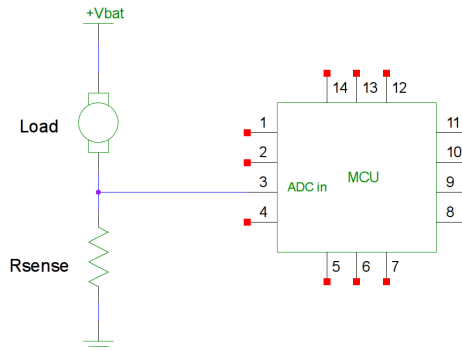

FIGURE 3.10: Current measurement through a simple resistor.

Considering now the sense resistor technique, the simplest manner, illustrated in Figure 3.10, is to connect directly the terminal of the resistor to the pin of the MCU that is configured to read an analog signal. The major drawback is that the signal is small as already discussed. For instance, for a 2 A maximal current and a 0.05 Ω resistor, the voltage will have a maximum value of 0.1 V. This is problematic because the MCU analog input voltage range is 3.3V while the signal only varies from 0 to 100 mV, which means that the resolution of the measurement will be poor. Actually, for this example, the resolution is 33 times less than what the ADC can achieve, which represents roughly a loss of 5 bits in the digital value obtained.
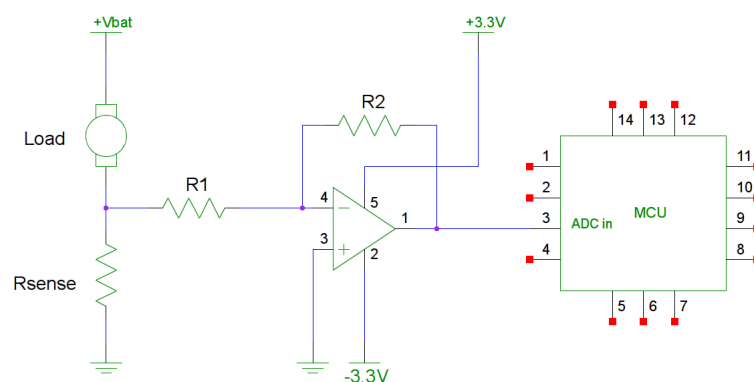

FIGURE 3.11: Current measurement through an operational amplifier.

A technique to overcome this resolution issue is to amplify the voltage across the resistor. In Figure 3.11, one can see an example of an operational amplifier (op-amp) in inverting
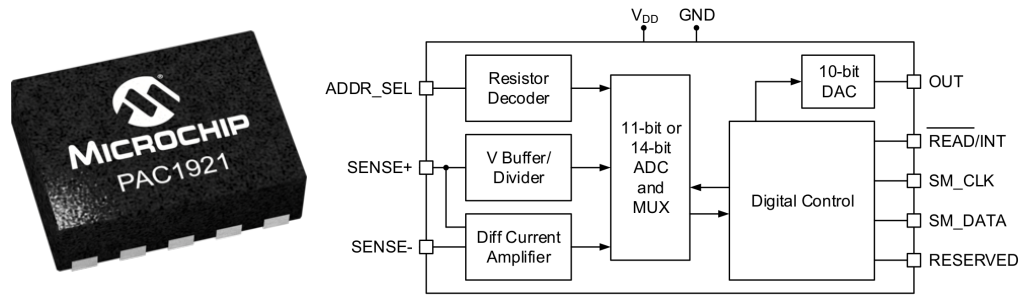
FIGURE 3.12: PAC1921 current/power monitor (Source: Microchip$^{\text{TM}}$)[11].

configuration. Op-amps are perfectly suited to amplify small signals and there are many benefits to use them. The input impedance is ideally infinite, in reality very large, which is a good thing since it affects as little as possible the input signal, although it is not really an issue because the source impedance is very low. The output resistance is low, so it minimizes the voltage drop caused by the load connected to it. Again this is not an issue because the load is a MCU input with a very large input impedance. The advantage of this solution mainly lies in the precise amplification of the signal. For the configuration of Figured 3.11, the gain [22] is given by Equation (3.4). One can see that the gain is negative, which means that the signal going to the MCU is the lowest when the current is the highest. This is not a problem for this project.

$$V_{out} = \frac{-R_2}{R_1}V_{in} \tag{3.4}$$

A third option that also uses a sense resistor is to use a dedicated chip called current/power monitor as shown in Figure 3.12. Inside this chip, there is an amplifier for measuring the current, then this signal goes through an analog to digital converter (ADC) that sends the measurement to a control unit. This unit is in charge of communicating the measurements and may compute some features as rolling average, etc. The PAC1921 from Microchip$^{\text{TM}}$ is an example of such device, it has an amplification gain of up to 128 times the input signal, an ADC with a resolution of 11 or 14 bits and communicates via I^{2}C or SMBus. A particularity of this component is that it also contains a digital to analog converter (DAC) that provides an output signal proportional to the current within the voltage range of the microcontroller. As usual, there is the issue of the space occupied by the component, which is the largest one among the three techniques enumerated.

In our case, there is a big issue that complicates the implementation of the three current measurement techniques that we have enumerated. The problem comes from the fact that the current to be measured may be negative. The load is a DC motor powered through a battery, so as long as there is no torque exerted on the motor, the current is consumed and flows from the positive terminal to the ground. If an external torque is applied to the motor, it may act as a generator, in which case the current may flow in the opposite direction and become negative according to the sensing device. In such a situation, the voltage at the resistor terminal goes below the 0 V, which is a problem since the MCU analog input can only measure a voltage from 0 V to 3.3 V. Even if the signal is amplified with an op-amp, a negative voltage is still possible.

A solution consists in adding a constant bias to the signal in order to never have a negative potential to measure. This can be done thanks to another op-amp circuit that uses the summing configuration, shown in Figure 3.13. This is roughly the same circuit as the inverter but this time there are many sources connected to the inverting input through a resistor R_x. The gain is given by Equation (3.5), one can see that the output is the sum of the input signals amplified by the ratio $\frac{-R_f}{R_x}$.

$$V_{out} = -\left(\frac{R_f}{R_1}V_1 + \frac{R_f}{R_2}V_2 + \frac{R_f}{R_3}V_3 \right) \tag{3.5}$$

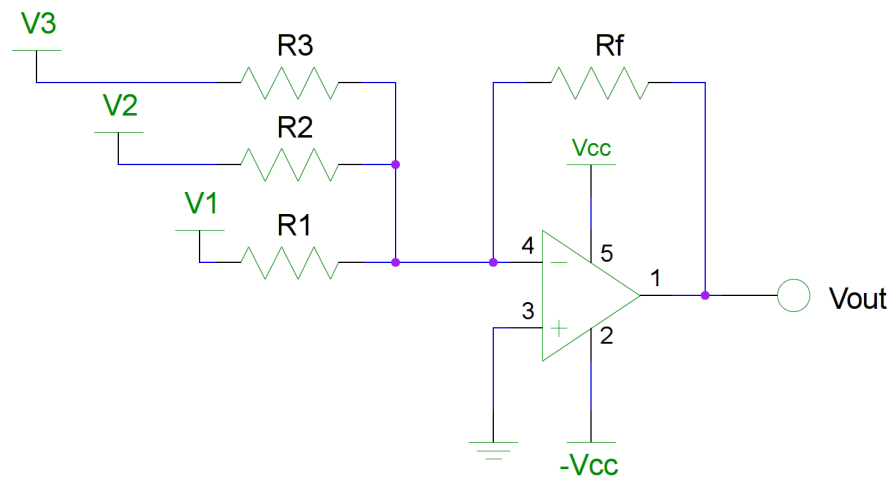FIGURE 3.13: Summing operational amplifier.

The summing amplifier is not an ideal circuit because the operational amplifier relies on a dual ± 3.3 V power supply. Since the space available on the board is very restricted, it is not practicable to have a DC-DC converter that provides ± 3.3 V, so it is necessarily

supplied with only $+3.3\,\text{V}$. The circuit can be slightly modified to correct this issue. The final circuit is shown in Figure 3.14. It includes a voltage divider made of two equal resistors connected to the positive input, thus providing a reference voltage around $1.65\,\text{V}$. There is a $100\,\text{nF}$ capacitor C_2 to stabilize the reference voltage of the op-amp and another one C_f in parallel with R_f that is used as a low-pass anti-aliasing filter. The cut off frequency chosen is about $1\,\text{kHz}$ which is more than twice below the sampling frequency at which the MCU will operate the measurement.
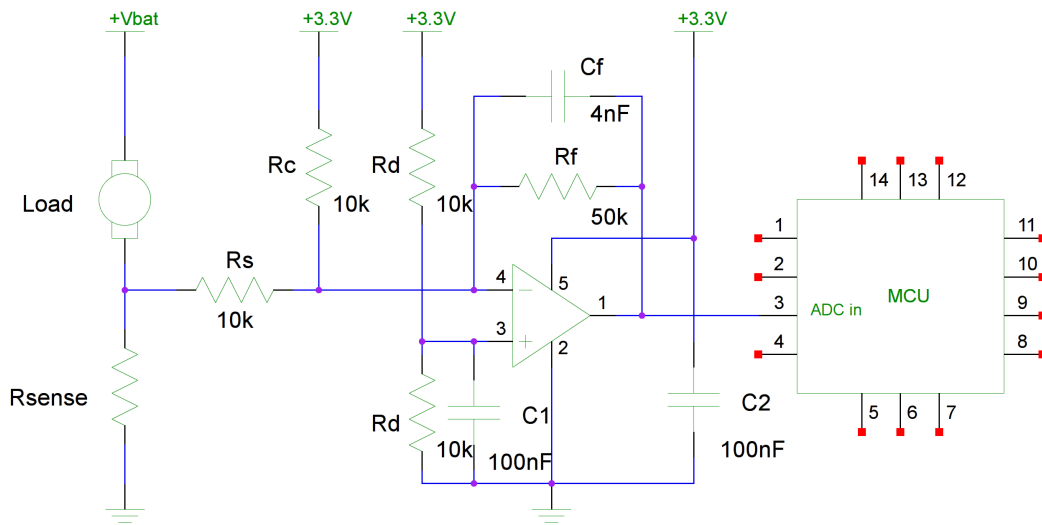

FIGURE 3.14: Implemented operational amplifier circuit.

The DC gain of this circuit is equivalent to that of the summing op-amp, but this time the input and output voltage are shifted upward by $1.65\,\text{V}$. The following development proves that result.

We assume the operational amplifier to be ideal, so that the voltages at its positive and negative inputs are considered equal ($V_+ = V_-$), and the current at the inputs is considered null ($I_+ = I_- = 0\,\text{A}$).

From these considerations, one may determine the voltage V_+ using the following equation:

$$V_+ = \frac{V_{cc}}{2} = V_- \tag{3.6}$$

By Kirchhoff's first law, one obtains that the current flowing from the sensing input (I_s) added to the current flowing through R_2 is equal to the current going through R_f, since there is no current flowing in the negative input of the op-amp. Kirchhoff's second

law can be applied on three loops, providing three more equations. Applying Equation (3.6), it leads to the following system of equations:

$$
\begin{cases}
I_c + I_s = I_f \\
V_s - R_s I_s = V_- \\
V_{cc} - R_c I_c = V_- \\
V_- - R_f I_f = V_{out}
\end{cases}
\Leftrightarrow
\begin{cases}
I_c + I_s = I_f \\
V_s - R_s I_s = \frac{V_{cc}}{2} \\
V_{cc} - R_c I_c = \frac{V_{cc}}{2} \\
\frac{V_{cc}}{2} - R_f I_f = V_{out}
\end{cases}
\tag{3.7}
$$

This yields:

$$
\begin{cases}
I_f = I_c + I_s \\
V_s - R_s I_s = \frac{V_{cc}}{2} \\
I_c = \frac{V_{cc}}{2R_c} \\
\frac{V_{cc}}{2} - R_f I_f = V_{out}
\end{cases}
\Leftrightarrow
\begin{cases}
I_f = I_s + \frac{V_{cc}}{2R_c} \\
V_s - R_s I_s = \frac{V_{cc}}{2} \\
I_c = \frac{V_{cc}}{2R_c} \\
\frac{V_{cc}}{2} - R_f I_f = V_{out}
\end{cases}
$$

$$
\Leftrightarrow
\begin{cases}
I_s = I_f - \frac{V_{cc}}{2R_c} \\
V_s - R_s \left(I_f - \frac{V_{cc}}{2R_c} \right) = \frac{V_{cc}}{2} \\
I_c = \frac{V_{cc}}{2R_c} \\
\frac{V_{cc}}{2} - R_f I_f = V_{out}
\end{cases}
\Leftrightarrow
\begin{cases}
I_s = I_f - \frac{V_{cc}}{2R_c} \\
I_f - \frac{V_{cc}}{2R_c} = \frac{\left(\frac{V_{cc}}{2} - V_s \right)}{-R_s} = \frac{V_s - V_{cc}}{R_s} \\
I_c = \frac{V_{cc}}{2R_c} \\
\frac{V_{cc}}{2} - R_f I_f = V_{out}
\end{cases}
$$

$$
\Leftrightarrow
\begin{cases}
I_s = I_f - \frac{V_{cc}}{2R_c} \\
I_f = \frac{\frac{V_s - V_{cc}}{2}}{R_s} + \frac{V_{cc}}{2R_c} \\
I_c = \frac{V_{cc}}{2R_c} \\
\frac{V_{cc}}{2} - R_f \left(\frac{\frac{V_s - V_{cc}}{2}}{R_s} + \frac{V_{cc}}{2R_c} \right) = V_{out}
\end{cases}
\Leftrightarrow
\begin{cases}
I_s = I_f - \frac{V_{cc}}{2R_c} \\
I_f = \frac{V_s - V_{cc}}{2R_s} + \frac{V_{cc}}{2R_c} \\
I_c = \frac{V_{cc}}{2R_c} \\
V_{out} = \frac{V_{cc}}{2} \left(1 - \frac{R_f}{R_c} + \frac{R_f}{R_s} \right) - \frac{R_f}{R_s} V_s
\end{cases}
$$

One can see from the last equation that in the particular case when $R_s = R_c$, one obtains Equation (3.8). The gain is exactly what was desired, an amplification by a factor $\frac{-R_f}{R_s}$ and a shift by half the supply voltage. The reference voltage of the inputs of the op-amp is around $\frac{V_{cc}}{2}$ as expected.

$$
V_{out} = \frac{V_{cc}}{2} - \frac{R_f}{R_s} V_s
\tag{3.8}
$$

The nominal current flowing into the motor is about $350\,\text{mA}$, but the starting current has a peak of $1.45\,\text{A}$. Taking a safety margin, one can estimate that the largest current that may be measured is $\pm 3\,\text{A}$. The sense resistor chosen because of its small size is `ERJ-8BWFR068V` from Panasonic$^{\text{TM}}$, with a resistance equals to $68\,\text{m}\Omega$. So following Equation (3.1), the maximal voltage across the resistor is given by (3.9), and must be between $\pm 204\,\text{mV}$ which correspond to a voltage swing (V_{swing}) of $408\,\text{mV}$.

From this value one can compute the ideal gain factor in order to achieve the best resolution of the MCU. The best case is when the signal to be measured varies between $0\,\text{V}$ and $3.3\,\text{V}$. So one obtains easily that 8.088 is the ideal gain following Equation (3.10). In fact, the op-amp may saturate or operate improperly if its output becomes too close to its supply rails. Nevertheless, there exists amplifiers qualified as rail to rail, meaning that they are able to provide an output voltage very close to its limit. As a trade off, we set the gain to 5, so the output voltage swing is $V_{swing,output} = 5\,V_{swing} = 2.04\,\text{V}$. It means that the output will vary from $630\,\text{mV}$ to $2.67\,\text{V}$, far enough from the rails to avoid any issue.

$$V_{sense} = RI = 0.068 \cdot 3 = 204\,\text{mV} \tag{3.9}$$

$$G_{ideal} = \frac{3.3}{V_{swing}} = \frac{3.3}{0.408} \simeq 8.088 \tag{3.10}$$

Once the gain is chosen, one may compute the value for each resistor:

$$\begin{cases} R_s = R_c \\ \frac{R_f}{R_s} = 5 \end{cases} \Leftrightarrow \begin{cases} R_s = R_c = 10\,\text{k}\Omega \\ R_f = 5\,R_s = 50\,\text{k}\Omega \end{cases} \tag{3.11}$$

The value of R_s is arbitrarily fixed to $10\,\text{k}\Omega$ because it is a common value met in electronic and because it is greater than R_{sense} so that the current flows mainly through the lowest resistor and the voltage at the input of the op-amp is the nearly same as at the terminal of the sensing resistor.

3.7 Angle measurement

The servomotor requires to know the position of its external gear in order to execute commands e.g. reaching a certain position. The RX-28 servomotor uses a potentiometer to measure the position but the resolution is quite bad (0.29°) and it is not able to measure every angle, but only over 300°. The MX-28 servomotor embeds a magnetic encoder allowing to measure any angle and the resolution is 0.088°, which is better than the potentiometer technique. So the technique used by the MX-28 is adopted for this project because of its advantage compared to the RX-28.

3.8 Temperature sensing

The monitoring of the motor is important if one wants to avoid the motor to overheat. A simple way to achieve this may be realized by simply gluing a temperature probe on the motor. Since the measurement does not have to be very precise, a negative temperature coefficient (NTC) resistor is well suited because it is very compact and cheap.

3.9 Power supply

The servomotor may be supplied with a voltage ranging from $12\,\mathrm{V}$ to $15\,\mathrm{V}$, but most of the components does not tolerate more than $3.3\,\mathrm{V}$. So there is a need for a voltage regulator that will supply these components with $3.3\,\mathrm{V}$.

The voltage regulator chosen is the `TL1963A-33` by Texas Instruments$^{\mathrm{TM}}$, it can supply a current of 1.5A, which is more than enough to power the components.

Chapter 4

PCB design

This chapter discusses the design of a printed circuit board implementing the electronic circuit introduced in Chapter 3. It explains how the constraints imposed by the project were dealt with.

4.1   Printed circuit board

A printed circuit board (PCB) is a board that carries all the components of a circuit and electrically connects them using conductive tracks. As illustrated in Figure 4.1 PCBs are made of copper sheets laminated onto a non-conductive substrate. These conductive layers are etched using masks and chemical processes in order to create tracks, pads and vias. The number of conductive layers may go from 1 up to 12, sometimes even more in complex applications.
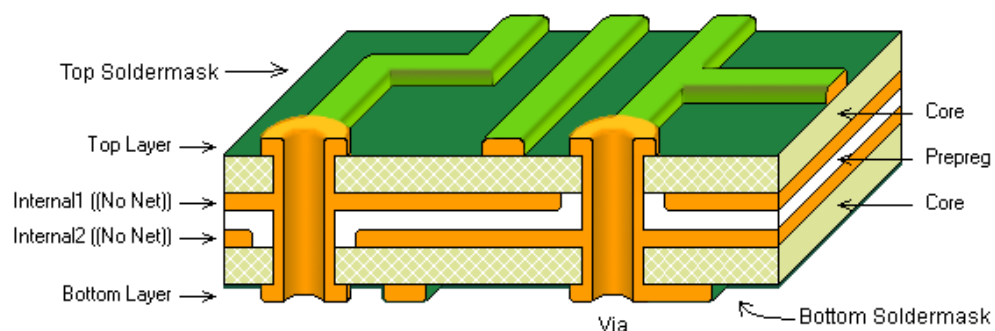
FIGURE 4.1: 4 layers PCB (Source: Engscope)[12].

The pads are small areas of copper at the surface of the board on which the components can be soldered. A board may also contain through-holes, holes surrounded by an annular pad at the board surface, allowing to solder radial-leaded components. These two types of technologies are illustrated in Figure 4.2. Actually, as it can be seen in Figure 4.3, a through hole is a type of via, of which there are three types: the buried via which is inside the board, the blinded via that is connected to only one side of the board, and the through hole. The via allows to connect tracks that belong to different layers.
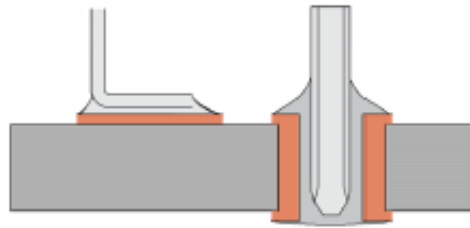
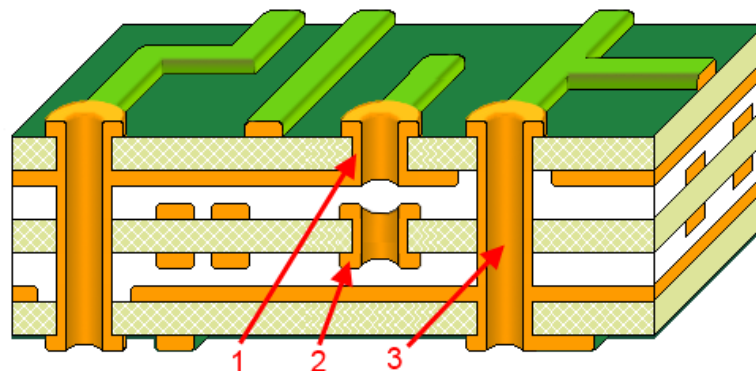FIGURE 4.2: 4 layers PCB (Source: Connector and Cable Assembly Supplier^{TM})[13].

FIGURE 4.3: Via types: 1 the blind, 2 the buried and 3 the through hole (Source: Altium^{TM})[14].

4.2 Electronic schematic

Before starting PCB design, one obviously has to build an electronic schematic, also called circuit diagram. This diagram is a graphical representation of the circuit allowing to see the components and the interconnections between them. Beside allowing one to reason about the circuit, the schematic carries others information such as the manufacturer reference of the components, their name, the footprint reference, etc. The footprint is the physical shape of the components and its pads. For instance in Figure 4.4, one can

see the footprint in gray of a through hole resistor, a surface mounted device (SMD) and a 8-pin small outline integrated circuit (SOIC). The black lines are the outline, which are silk-printed in white on the board.
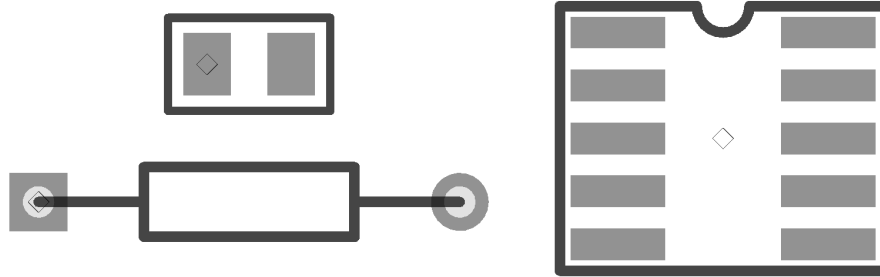

FIGURE 4.4: Different footprints.

From the schematic of the circuit, one can generate the bill of material (BOM) which is the list of all the components that are in the circuit, the netlist, which is a list of connection between terminal nets of components. The netlist has two purposes during the PCB design phase: it shows to the design which pads have to be connected by tracks and it is also used to check if there is no short circuit between nets that should not be connected together.

The `gEDA` software suite has been used for the development of our electronic board. It is a set of CAD software used for electronic design released under the GNU General Public License. `Gschem` is the schematic designer application and `Pcb` is the PCB designer. This tool suite has been used because of prior knowledge and skills acquired for other projects, and also because it is free.

During the schematic design amounts to nothing more than assembling and connecting every constitutive part in a diagram. Some last adjustment have been made such as the addition of decoupling capacitors. These capacitors are used to reduce as much as possible the high-frequency noise present in the power supply of components. The best practice rules specify that there should always be a 100 nF decoupling capacitor between the power terminals of a component. The schematic is split in two and is shown in Figures 4.5 and 4.6
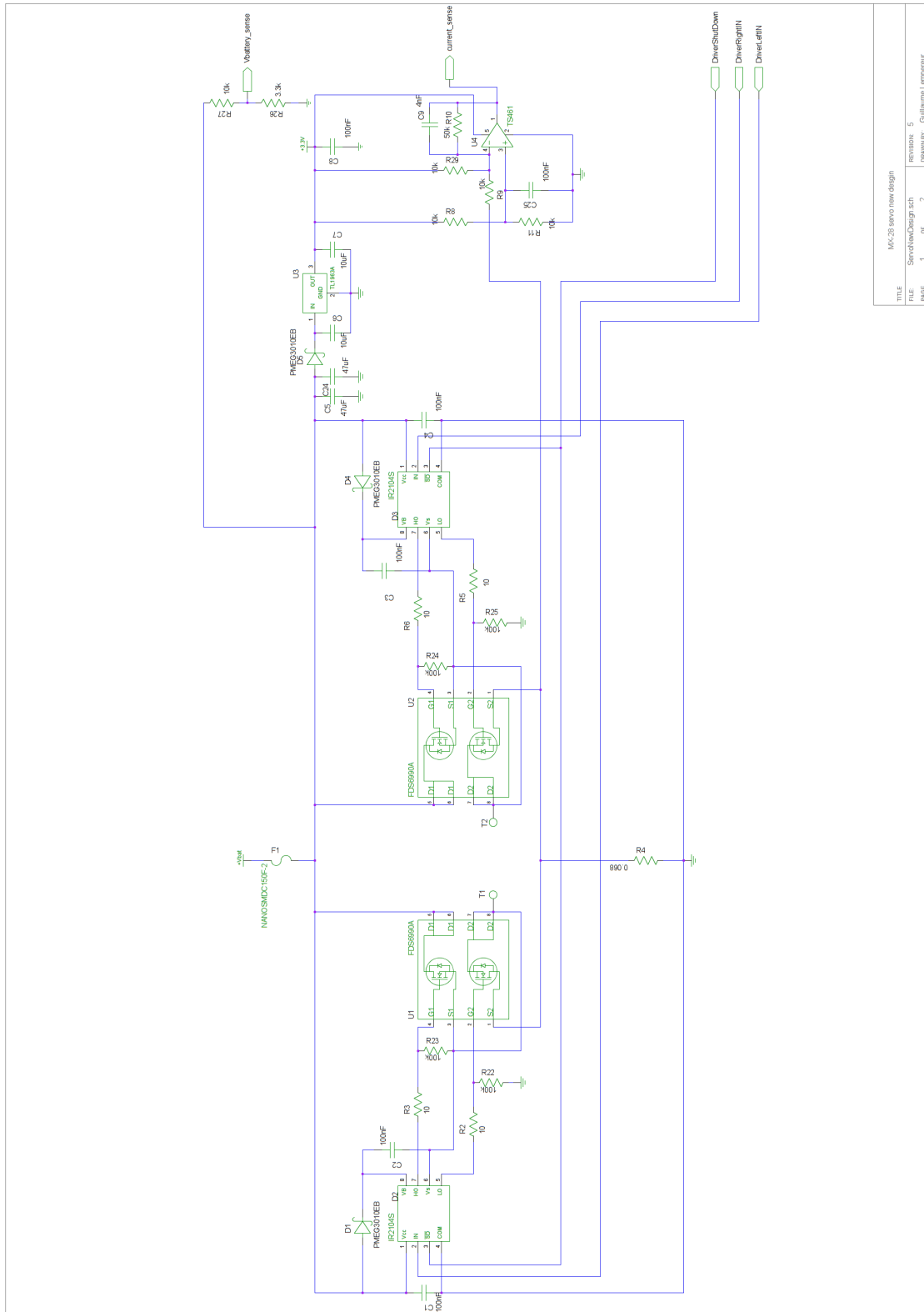
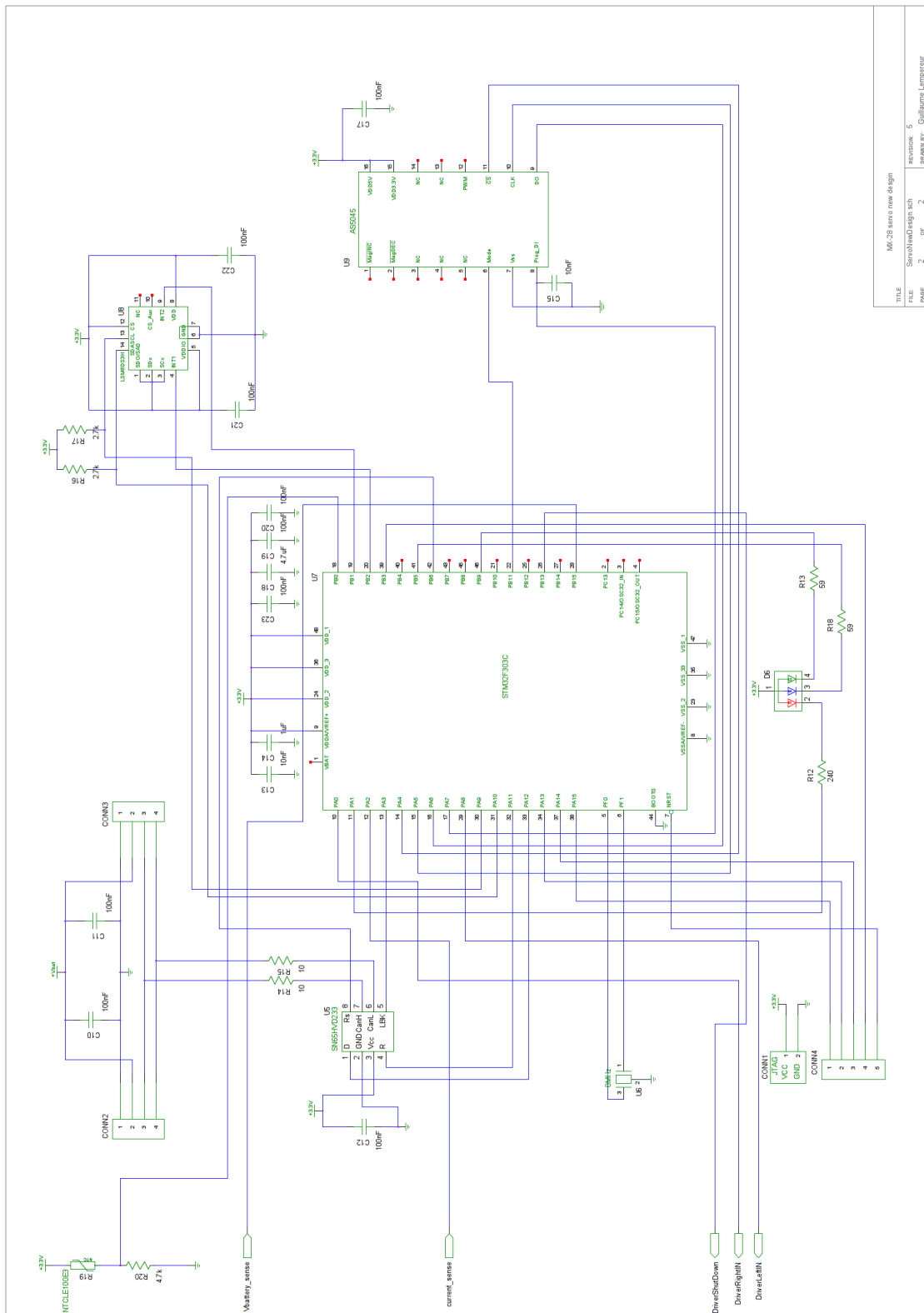FIGURE 4.5: Schematic of the developed electronic board part 1.

FIGURE 4.6: Schematic of the developed electronic board part 2.

4.3   PCB design

Once the schematic is ready, the PCB design can finally begin. The first step consists in compiling the schematic in order to generate a PCB file that contains each component footprint on the top layer. From this file, one has to produce a completed design in the form of a compilation of layers that can be used for physically building the board. The layers composing the PCB are enumerated below with the reference to the figure corresponding to that layer. The top and bottom layers, in Figures 4.7 and 4.8, are at the surface of the board where the components are soldered. There are internal layers containing additional traces (Figure 4.9) as well as ground and power plane essential to the proper operating to the MCU (Figure 4.10). Those layers will serve to make the masks necessary to the etching of the copper laminated sheets. There is a special layer called the outline, or mechanical, layer (Figure 4.11) that defines the boundaries of the board, it is used to guide the cutting machine. Finally there are the silk layers in Figures 4.12 and 4.12 that correspond to what will be printed on to the board using white ink.
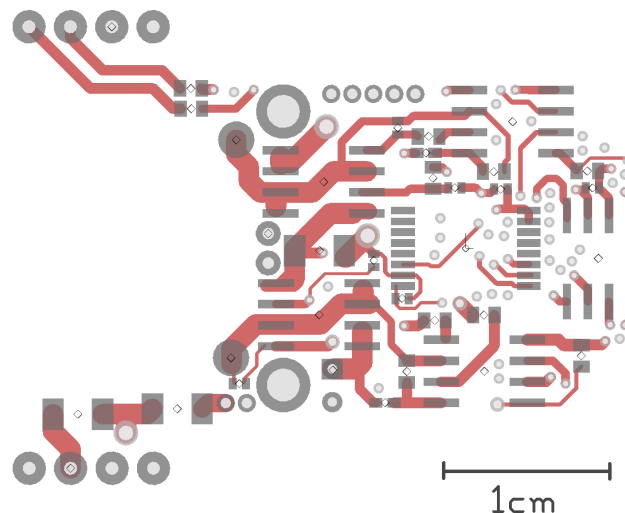

FIGURE 4.7: PCB top layer.

This electronic board has many constraints that complicate its design. From the mechanical point of view, the board has to fit in the MX-28 servomotor enclosure which implies that it has to remain relatively small. Since the space is restricted, the shape given to the board is the one that maximizes the available surface inside the device. In Figure 4.11, one can see the boundaries and notably the notch where the motor will be placed. There are two threaded holes in the servomotor frame near the notch that
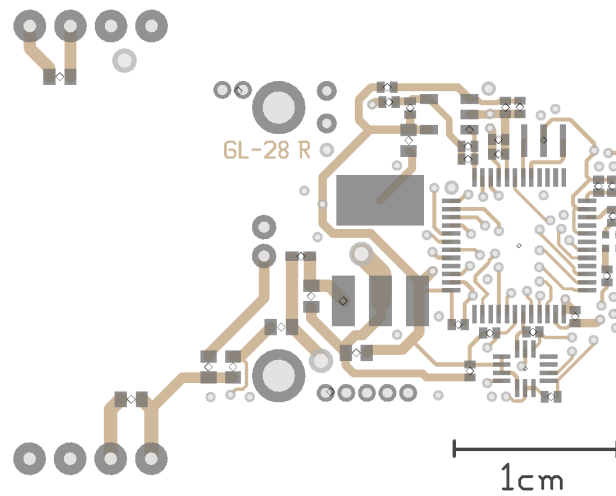
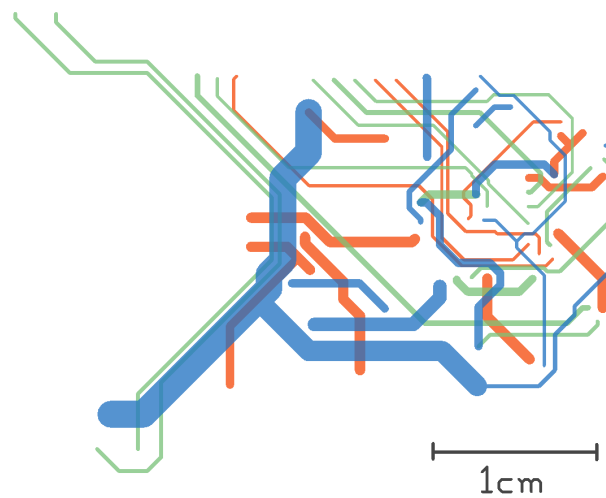FIGURE 4.8: PCB Bottom layer.

FIGURE 4.9: PCB three internal layers.

allows to screw the board to the plastic enclosure of the servomotor, so space has to be set aside for two holes of 2 mm in diameter. The magnetometer has to be precisely centered with the magnet that is fixed on the shaft of servomotor. The positioning is very important because a misalignment greater than 250 µm may cause the magnetometer to be unusable. Obviously the component has to be soldered on the side facing the shaft. There is also light guides in the servomotor that allows to reflect the light emitted by a LED fixed on the board. So the LED has to be positioned at a specific location, in the middle and at the right-most corner of the board, on the side facing the servomotor
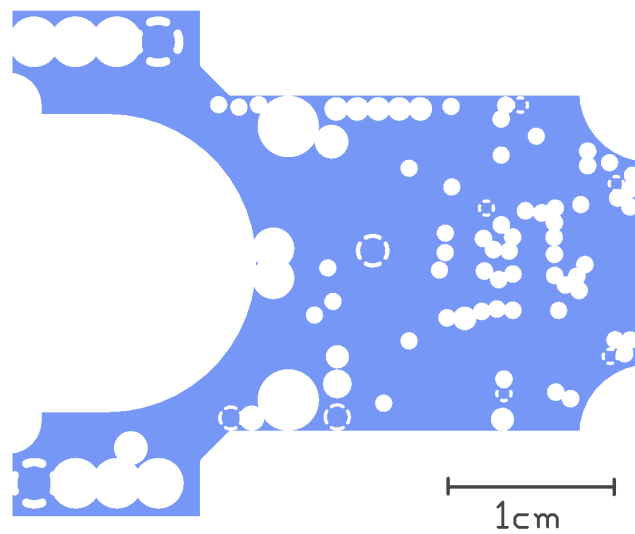
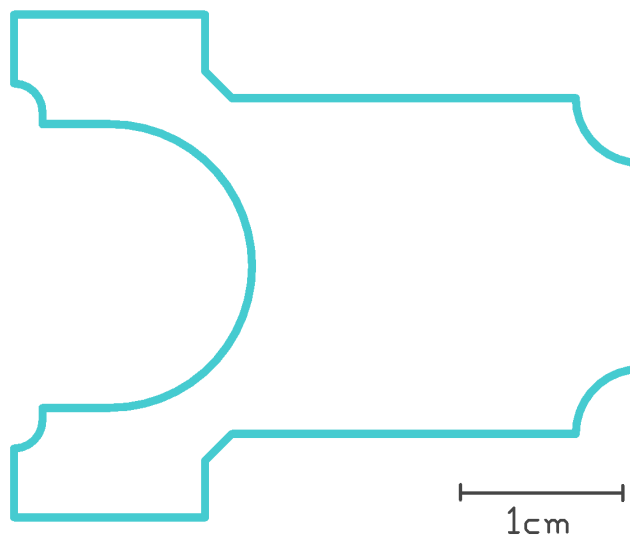FIGURE 4.10: PCB ground plane in an internal layer.

FIGURE 4.11: PCB outline layer.

cover. Last but not least, all the components have to fit within the boundaries of the board.

The arrangement of the components on the board is very important because if it is badly done, there may be some disastrous consequences. The major problem resulting from an inadequate arrangement is the appearance of strong parasitic signals at the input of some components that may lead to improper operation. A signal measurement may be distorted due to added noise caused by nearby parasitic signals. Communication may fail for the same reasons if the data channels are not properly insulated from noise.
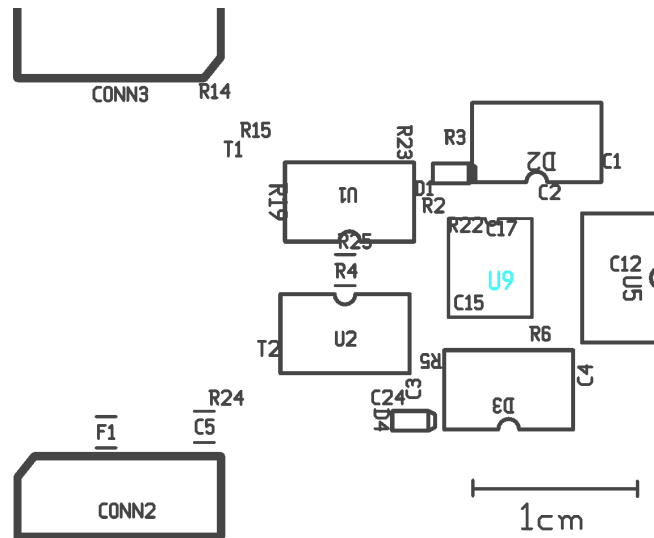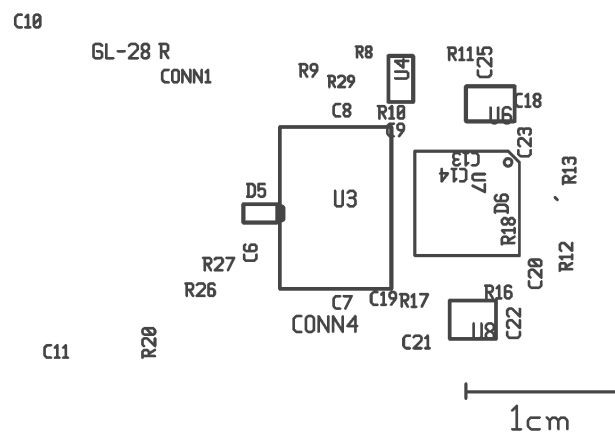
FIGURE 4.12: PCB silk top layer.

FIGURE 4.13: PCB silk bottom layer.

As depicted in Figure 4.14, there are three features that characterize parasitic effects: the source of noise, the coupling and the perturbed circuit. Parasites can be characterized in four categories as illustrated in Figure 4.15. If the coupling is achieved via a conductor material, it is a galvanic parasite, otherwise the parasitic signal is radiated towards the target. When the signal frequency is high enough so that the length of the circuit is equivalent or greater than the signal wave length, the parasite is said electromagnetic. If the circuit length is negligible compared to the signal wave length, then the parasites are said to be inductive when the current is more important than the voltage and capacitive otherwise.

There are three main approaches to overcome parasitic issues. The first is to tackle the
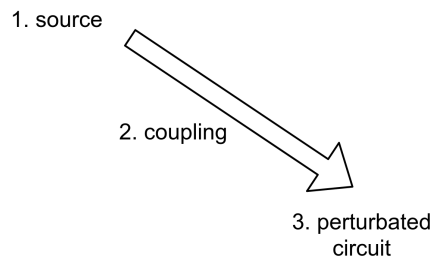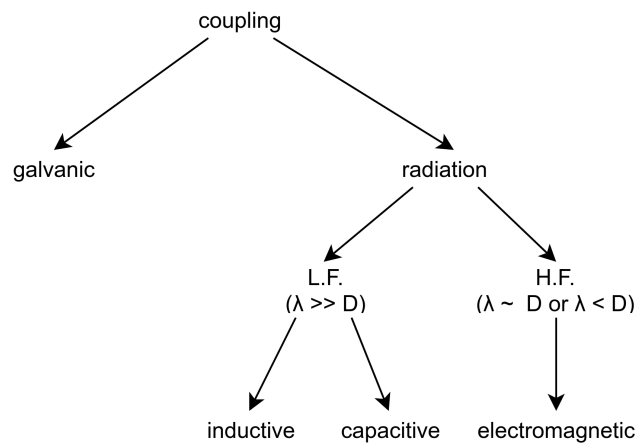
FIGURE 4.14: Parasitic effect [15].

FIGURE 4.15: Parasitic effect [15].

noise source, for instance removing every useless component that generates parasites. This seems quite obvious but it is not often feasible because almost all components present in the circuit are required. The second way is to reduce the coupling, many techniques that depends on the type of coupling allow to decrease parasites. The last way is to act on the perturbed circuit, for instance one could add a filter that reduces high-frequency noise.

Considering the coupling reduction, if the problem is galvanic, one may put a guard around the device that is sensitive to interference. For radiated parasites, there are different techniques depending on the kind of parasite. For inductive noise, the loops in the circuit have to be reduced as much as possible while for the capacitive noise, the plates surface have to be reduced and the air gap has to be increased. It means that the tracks have to be as short as possible and as far as possible from each other.

To avoid parasitic issues, the best practice rules recommend isolating and decoupling three different parts of the circuit and connecting them to the ground via a single node

as illustrated in Figure 4.16. These parts are the power which consists of components in which important currents go through, the digital part which contains chips that process low power signals at high frequencies carrying discrete information and the analog part consisting in components that process low power continuous signals that carry time varying quantities.
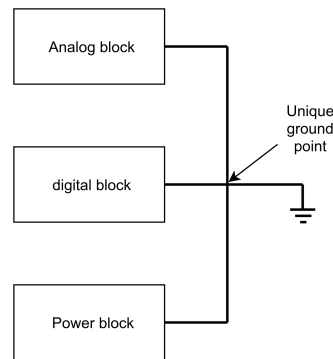

FIGURE 4.16: Analog, digital and power block separation [16].

As far as possible those rules have been applied during the design of our PCB but obviously, it is impossible with such strong space requirements to follow each rule to the letter. As it can be seen in Figures 4.17 and 4.18, the digital, analog and power parts have been fairly separated but connected to a unique ground plane because it is impossible and not relevant, to have a separate ground plane for every part. The ground plane layer has been located just below the microcontroller because it is the component that is the most sensible to parasites and there is also the op-amp on this side. The power block is as close as possible to the notch for the motor and their terminals are close to each other in order to form a small current loop. The resonator is very close to the MCU, which is very important for the stability of the clock signal. The digital components such as the IMU, the magnetometer and the transceiver are on the right part of the board and close to the microcontroller in order to limit electromagnetic emissions and to increase interference immunity.

To realize all the connections, several internal layers were required. Beside the top and bottom layer, there is a layer dedicated for the ground plane and three others for the tracks as it can be seen in Figure 4.9, so that the total number of layers is six. The cost of a manufacturing PCB mainly depends on three features, the number of layer, the pattern class and the drill class. The pattern class is defined by several features, but the more important is the minimal space between two copper areas. The drill class depends

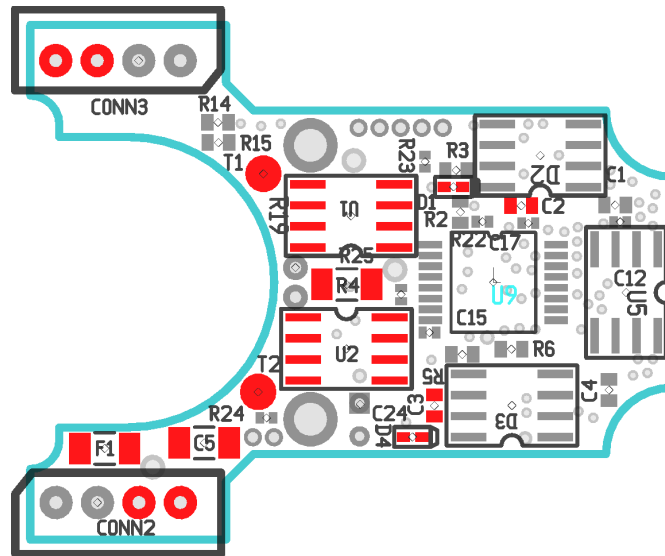FIGURE 4.17: PCB top layer divided in parts: in red the power and in gray the digital.
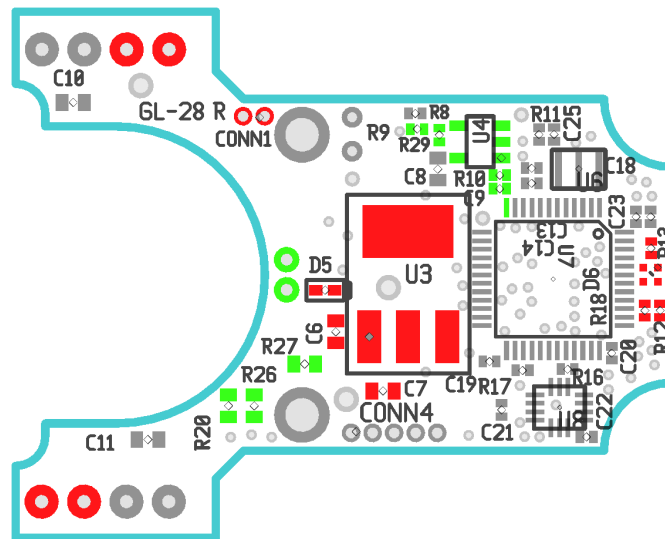

FIGURE 4.18: PCB bottom layer divided in parts: in red the power, in green the analog and in gray the digital.

on the diameter of the via. For this PCB, the Eurocircuits$^{\text{TM}}$ class is 6C, meaning the minimal space between two copper areas is $150\,\mu\text{m}$ and the minimum via diameter is $350\,\mu\text{m}$.

Chapter 5

Electronic board testing

This chapter deals with the phase that follows the PCB design which is testing the assembled board. The first part is about the soldering of the components, the techniques used and the difficulties encountered. The next part deals with the microcontroller programming that will allow to test the different parts of the electronic circuit.

5.1 Soldering

The first step after receiving the manufactured PCBs was to perform a quick quality control by using a multimeter in continuity testing mode, in order to check that every pad is connected, meaning that the internal layers are included in the PCB and that there is no defect. Then the components have been soldered to the board. Two methods have been used because of some components that require a special technique. A first technique is used for the microcontroller which has a low profile quad flat package (LQFP), shown in Figure 5.1 , for which the space between two pins is only 500 µm so using a soldering iron would be impractical. The inertial measurement unit chip has a 14 contacts land grid array (LGA) package shown in Figure 5.2, that does not have pins but small rectangular contacts. These two components are soldered using a paste that is deposited over the pads, then the chip is positioned on the paste and the board is heated to melt the paste so that when it is cooled, it forms a mechanical and an electrical connection. The second method used is simply to solder manually the components using a soldering iron.

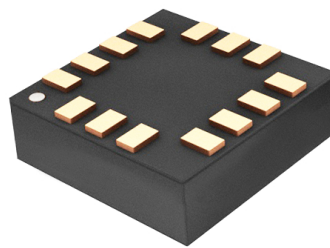FIGURE 5.1: Low profile quad flat 48-pin package used by the MCU (Source: 3D Content Central$^{\text{TM}}$)[17].

FIGURE 5.2: Land grid array 14-pin package used by the IMU (Source: Distrelec$^{\text{TM}}$)[18].

After soldering, an important quality control phase has to be done in order to check that there is no short circuit due to an unintended bonding during soldering. Once it is sure there is no short circuit, one may power the electronic board with a bench power supply which is set to the nominal voltage of the servomotor (12 V) and has a maximal current fixed to 70 mA. The current limit is a security that ensures that in the case of a short circuit, the current will be limited to not damage the board and its components.

5.2 Programming the MCU

Most of the components and parts of the circuit can be tested using small programs specially dedicated to this purpose. To achieve that, it is necessary to program the microcontroller, meaning to write the application software in the MCU memory, so that it starts to execute the test application when it is turned on. To do so, a toolchain is required, which is a set of programming tools allowing to compile and to flash a program in the memory of the MCU. The Figure 5.3 illustrates the toolchain concept, it consists of a compiler, a linker, some libraries, a debugger and programming software.
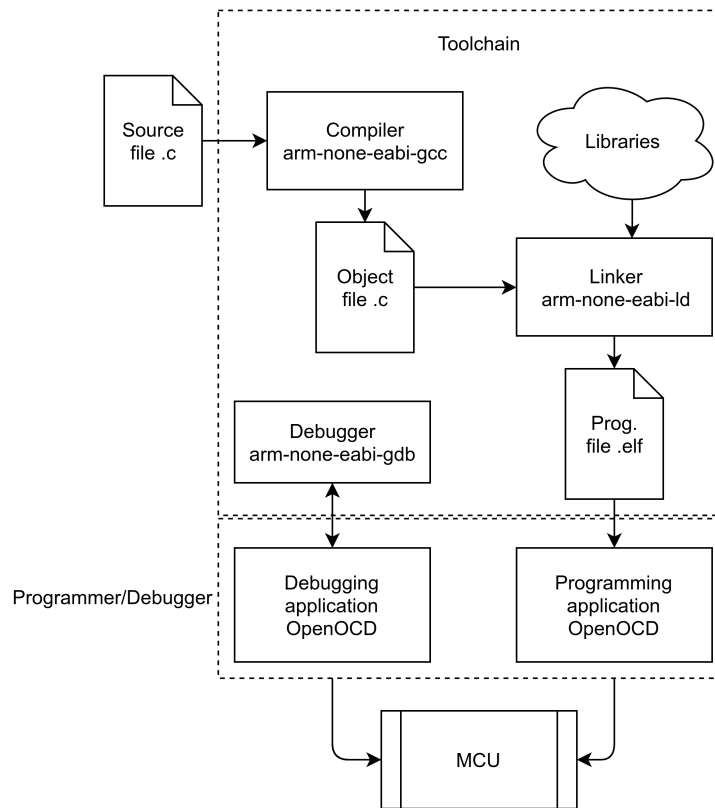
FIGURE 5.3: Toolchain diagram.

Initially, computer program is expressed as source code written in one or many pro-
gramming languages. This source code has to be compiled or interpreted in order to
be executed by a computer. The C language has been chosen to write the softwares of
this project because it is the most widespread in the embedded world, it is a compiled
low-level language (meaning it is close to assembly language and machine code), which is
an advantage when the programs are simple. The compiler is in charge of transforming
the C source code into object files that are function definitions in binary form but are
not executable by themselves. Then comes the linker that takes the object files and
the libraries containing general-purpose existing code , combines them and produces a
single executable file. The generated file is ready to be flashed in the memory of the
microcontroller thanks to a program that makes the link between the computer and
the programmer device. There are many toolchains as Keil^{TM}'s `MDK-ARM`, Atollic^{TM}'s
`TrueSTUDIO` and IAR^{TM}'s `EWARM`, but for this project, we use the OpenOCD toolchain
that has the main advantage of being free. `OpenOCD` stands for open on-chip debugger,
it comes with `arm-none-eabi-gcc` which is a compiler based on the GNU Compiler
Collection (GCC) and the debugger `arm-none-eabi-gdb` based on the GNU Debugger

(GDB). The programmer device is embedded in every evaluation board that ST$^{\text{TM}}$ sells, so the programmer that we have used is the one embedded in the `STM32F3DISCOVERY` evaluation board which uses the same microcontroller than the one employed in this project. This programmer uses the serial wire debug (SWD) protocol to program the MCU which is a simple serial communication protocol.

The very first function that was programmed in the microcontroller is a simple code that turns on the tricolor LED and periodically change its color. This test allows to check that the MCU can correctly be programmed, and that the tricolor led lights up correctly. The microcontroller was set to use its internal clock in the sake of simplicity, but our circuit contains an external resonator connected to the MCU that allows to have a better stability of the clock. So the second test was similar to the first one, but this time with the MCU configured to use the external resonator, and a period is set to 500 ms. This second test validates that the external resonator works fine and the MCU clock is correctly set to the desired frequency.

From these results, one can move on to driving the H-bridge with a small program that generates two PWM signals in quadrature with a 50 % duty cycle, allowing to drive the motor with the locked anti-phase technique. This test made it possible to detect an error made during the schematic design, the pump charge capacitor is not connected to the load terminal. As a consequence, the high-side transistor is not correctly drive and the H-bridge does not work. This mistake motivated a redesign of the PCB in order to correct this issue. Nevertheless, by modifying slightly the electronic board, the H-bridge could be tested proving that this mistake being corrected, the circuit works as expected.

The temperature sensing, the battery voltage measurement and the motor current measurement consist in an analog reading at the microcontroller level. So a simple code that configures the corresponding pin to read the voltage was used to test each of these three features and every test succeeded.

The magnetometer communicates with the microcontroller via serial communication (SPI). To enable SPI on the microcontroller, there exist some libraries that manage communication with words of 8 up to 16 bits. A problem is that the magnetometer operates with 18-bit long words, so communicating with this device requires homemade functions. The test that we performed was not adapted to the situation, but it revealed

that data is correctly exchanged between the magnetometer and the MCU, so this part works properly.

The inertial measurement unit communicates with the MCU via I^{2}C. Some libraries are also available to implement such a communication with the microcontroller but due to limited familiarity with those libraries, this part could not be tested. The same goes for the CAN bus communication which is even harder to manage.

Chapter 6

Conclusion

6.1 Achievement

The project consisted in developing an original electronic board that controls the Dynamixel^TM MX-28 servomotor in a way that enhances its control capabilities. The control usually consists in reaching a position or a torque, our board can be programmed to perform complex controls such as position or torque orders that vary over time. Another type of control that could be achieved is a sequence of predefined orders that would be performed according to some events that trigger them. The main challenge that was recurring from the electronic design to the PCB design, was to be able to embed each functionality in such a small electronic board. Hopefully, through hard work and perseverance, the puzzle has been solved and every part has been included in the electronic board.

Another improvement of our design is a better communication infrastructure between the servomotor and an external controller. The CAN bus has been used to replace the serial communication channel used in Dynamixel^TM servomotors, leading to improved reliability and throughput. In order to be able to implement sophisticated command, the microcontroller that has been embedded is the `STM32F303CC`. It improves the performance achieved by the MX-28 original MCU because it is more powerful, has a larger memory.

It was important to be able to control the servomotor at any angular position, so that it can rotate over 360° and provide at all times a precise measurement of the position

of the shaft. This is realized thanks to the magnetometer `AS5045` already used in the MX-28 servomotor.

Information about the motion of the servomotor in space, acceleration, angular speed, etc, are made available to an external controller using coupled gyroscope and accelerometer that forms an inertial measurement unit.

To perform complex control, one has to know precisely the voltage and the current supplying the motor. The voltage measurement has been simply achieved thanks to a potential divider while the current sensing is a much more involved solution. An operational amplifier has been used to amplify the signal coming from the sensing part, allowing to have a better resolution than the RX-28 and the MX-28 servomotors. Our solution has the additional advantage of being to be able to measure negative current.

The monitoring of the motor has been largely improved thanks to a temperature sensor located directly on the motor, which provides a far better solution than relying on temperature sensing inside the microcontroller as done in the RX-28 and MX-28 servomotors.

Beyond the development of this electronic board embedded in a MX-28 servomotor, this project can serve as a basis for developing electronic boards adapted to other servomotors e.g. the RX-64 which is a servomotor belonging to the same family but that develops a higher torque than the RX-28 and the MX-28.

6.2 Further development

The inertial measurement unit, the CAN bus communication and the magnetometer have not been tested in their usual mode of operation because of some difficulties to handle some software libraries. The next step would thus be to properly communicate with those components, in order to validate the whole electronic board. Once the board has succeeded every test, one could implement a controller and program it in the MCU. At this stage, the servomotor would be ready to be used in robotic applications.

It also worth mentioning that some people working in mechanical engineering department are taking a close look at this project because the control capabilities of the servomotor outperform what is achieved by current servomotors. So the future of this

project seem to be ensured not only for robotic usage but for many other application fields.

Bibliography

[1] Ralf Roletschek. Eindhoven, niederlande, robocup 2013. 2013.

[2] Trossen robotics. Dynamixel rx-28 robot actuator from robotis. 2016. URL http://www.trossenrobotics.com/dynamixel-rx-28-robot-actuator.aspx.

[3] Maxon motor^{TM}. Re-max 17 17 mm, graphite brushes, 4.5 watt, with terminals. 2016. URL http://www.maxonmotor.com/maxon/view/product/216000.

[4] trossenrobotics.com. Dynamixel mx-28t robot actuator. 2016. URL http://www.trossenrobotics.com/dynamixel-mx-28-robot-actuator.aspx.

[5] Vince Biancomano. Free development kit for rotary encoder ics cuts time-to-market. 2006. URL http://www.eetimes.com/document.asp?doc_id=1300440.

[6] International Rectifier^{TM}. Ir2104(s) and (pbf). page 1, 2004. URL http://www.infineon.com/dgdl/ir2104.pdf?fileId=5546d462533600a4015355c7c1c31671.

[7] DLN ware. I2c bus protocol. 2016. URL http://dlnware.com/dll/I2C-Bus-Protocol.

[8] Digi-Key electronics^{TM}. Fis1100 fairchild semiconductor — sensors, transducers — digikey. 2016. URL http://www.digikey.com/product-detail/en/fairchild-semiconductor/FIS1100/FIS1100CT-ND/5247136.

[9] Digi-Key electronics^{TM}. Lsm6ds3 inemo inertial module. 2015. URL http://www.digikey.com/en/product-highlight/s/stmicroelectronics/lsm6ds3.

[10] NDT Ressource Center. Measuring magnetic fields. 2016. URL https://www.nde-ed.org/EducationResources/CommunityCollege/MagParticle/Physics/Measuring.htm.

[11] Microchip[TM]. Pac1921: High-side power/current monitor with analog output. 2016. URL http://ww1.microchip.com/downloads/en/DeviceDoc/200005293C.pdf.

[12] Engscope. Pcb basics. 2010. URL http://www.engscope.com/pcb-fab-tutorial/02-pcb-basics/.

[13] Connector and Cable Assembly Supplier[TM]. Device connection integration in the smt process. 2010. URL http://www.connectorsupplier.com/device-connection-integration-in-the-smt-process/.

[14] Altium[TM]. Via — online documentation for altium products. 2013. URL https://techdocs.altium.com/display/ADRR/PCB_Obj-Via((Via))_AD.

[15] Prof. Philippe Vanderbemden. Application of electrical measurement systems course chapter 1. 2016.

[16] Prof. Philippe Vanderbemden. Application of electrical measurement systems course chapter 3. 2016.

[17] 3D Content Central[TM]. Lqfp-48. 2012. URL http://www.3dcontentcentral.fr/Download-Model.aspx?catalogid=171&id=363825.

[18] Distrelec[TM]. Lga-14. 2016. URL http://www.distrelec.ch/fr/capteur-acceleration-lga-14-st-lis2dhtr/p/30018612.

[19] Andras Tantos. H-bridges - the basics. pages 6–9, 2011.

[20] Hubert Woczszyk. Master thesis: Simulation of complex actuators. 2016.

[21] Andras Tantos. Controller area network (can) interface in embedded systems. 2015. URL http://www.eeherald.com/section/design-guide/esmod9.html.

[22] Walt Jung. Op amp applications handbook. page 10, 2016.