escola superior de tecnologia e gestão
instituto politécnico de leiria

DEVELOPING COLLABORATIVE APPLICATIONS USING THE KUKA IIWA

Master degree in Electrical and Electronic Engineering

MAILDO LOPES PINTO

Leiria, September of 2021

escola superior de tecnologia e gestão
instituto politécnico de leiria

DEVELOPING COLLABORATIVE APPLICATIONS USING THE KUKA IIWA

Master degree in Electrical and Electronic Engineering

MAILDO LOPES PINTO

Project Report under the supervision of Professor Doctor Hugo Filipe Costelha de Castro (hugo.costelha@ipleiria.pt) and Professor Doctor Carlos Fernando Couceiro de Sousa Neves (carlos.neves@ipleiria.pt).

Leiria, September of 2021

ORIGINALITY AND COPYRIGHT

This project report is original, made only for this purpose, and all authors whose studies and publications were used to complete it are duly acknowledged.

RESUMO

Desenvolver aplicações para robôs colaborativos, como o KUKA iiwa, é uma tarefa desafiadora. Esse tipo de robôs geralmente têm uma arquitetura difícil de interagir e sua linguagem de programação não é muito amigável. No Laboratório de Robótica Avançada e Fábricas Inteligentes do Instituto Politécnico de Leiria, temos um robô industrial, modelo KUKA iiwa 7 800. O projeto descrito neste relatório foi constrído de forma a facilitar o desenvolvimento de aplicações robóticas na indústria, com base na ideia de interagir com o robô KUKA por meio de um Sistema Adaptador da Flange Multifuncional, para realizar uma tarefa de pegar e posicionar e uma tarefa de seguir caminho com reação de força. Para isso, foi desenvolvido um dispositivo composto por dois módulos, que são controlados por microcontroladores ATmega, e podem se comunicar por fio ou sem fio. O robô e o dispositivo estão ligados à unidade de controle KUKA, por meio de portas digitais de um módulo Beckhoff. Os testes realizados e a análise dos resultados mostraram que o sistema funcionou conforme pretendido.

Palavras-chave— KUKA iiwa, Dispositivo de Guia Manual, Pegar e Colocar, Feedback de Força, Aplicações de Manipulação

ABSTRACT

Developing applications for collaborative robots, like the KUKA iiwa, is a challenging task. These types of robots generally have a difficult architecture to interact with, and their programming language is not very user-friendly. In the Advanced Robotics and Smart Factories Laboratory of the Polytechnic Institute of Leiria, we have an industrial robot, model KUKA iiwa 7 800. The project described in this report was built in order to facilitate the development of robotic applications in the industry, based on the idea of interacting with the KUKA robot using a Multi-purpose Flange Adapter System, to perform a pick and place task and a Path Following with Force Feedback task. To do so, a device was developed consisting of two modules, each controlled by an ATmega microcontroller, and can communicate with each other by wire or wirelessly. The robot and device are linked with the KUKA control unit, using digital ports from a Beckhoff module. The conducted tests and the analysis of the results showed that the system worked as intended.

Keywords— KUKA iiwa, Handguiding Device, Pick Up and Place, Force Feedback, Manipulation Application

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

Collaborative robotics is becoming increasingly popular both in research and industrial settings. In the collaborative paradigm, robots do not replace humans, instead, they help operators in accomplishing a common objective in a shared workspace. This idea has found many applications, such as cooperative work in assembling small parts, welding [23], picking and manipulating objects [24] and training in surgery [25].

1.1 MOTIVATION

The School of Technology and Management (ESTG) of the Polytechnic of Leiria has been working industrial robotics for many years, and has been increasing its research and development in collaborative robots. This work contributes to that effort, by developing and implementing additional collaborative applications using the KUKA iiwa 7 R800 available at the Advanced Robotic and Smart Factories laboratory of the ESTG, Polytechnic of Leiria.

The KUKA iiwa 7 R800 robot is a very versatile and complete collaborative robot, allowing the development of numerous applications that require precision, repeatability and force/torque sensitivity. However, programming this robot is not always an easy task. In addition to the user needing a certain level of knowledge in the java programming language used by the KUKA iiwa software, in the applications the robot follows three-dimensional points that contain the position of the flange and its orientation in the workspace. These points can be difficult to specify when doing online programming, particularly if a virtual model of the workspace does not exist or no simulation software is used, where the user, through the robot console joystick, has to move and position the robot's tool manually in order to then identify such points in space.

Guiding the robot using the joystick is not an easy task, positioning the KUKA robot in manual mode presents some problems. For example, if the operator is using axis jogging mode in the jogging options on the robot console joystick, also known as smartPad, he will only be able to perform one action at a time in each robot joint, and this can be a slow and tedious task. If the operator changes the jogging mode to the world coordinates translation/rotation motion on the robot's smartPad,

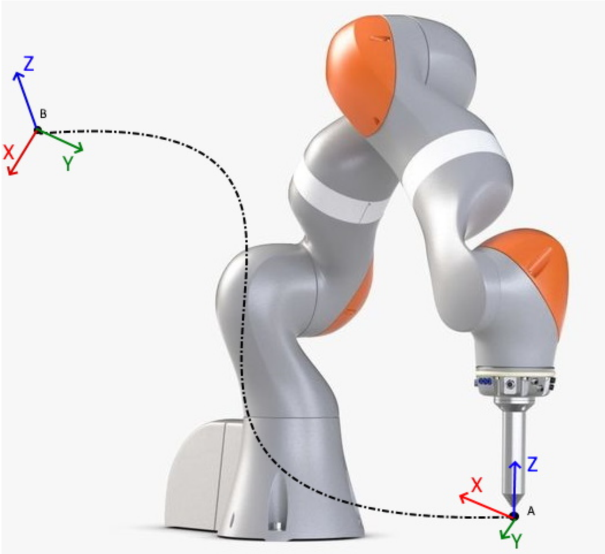


Figure 1: Example of a difficult move in manual control.

it will now be possible to move the robot through the Cartesian x, y and z axes which, depending on motion being applied, it can still be a slow task. These are the two ways to operate the KUKA robot, or any other industrial robot, in manual mode. Now imagine having to move the robot flange from a point A in front of the robot, to a point B behind the robot, as shown in Figure 1, in this example, regardless of which mode you are using, you will need to go through a series of target points until it reaches the destination.

It would be much easier if there was another way to drive the robot. The KUKA robot has a mode where the robot can be moved just by touching it and manually guiding its flange, without having to use the joystick. To use this robot mode, a device, usually coupled to a flange, is required, which allows the user to interact and choose when to activate or not this mode of operation. With this device, the robot can be guided easily by the operator, since now, the operator just takes it to the desired point, and the robot itself is in charge of making the necessary movements to follow the operator's movement.

1.2 OBJECTIVES

The aim of this project is to provide a simple, cost-effective solution for a hand-guided device for the KUKA robot and, with this solution, explore the robot's capabilities. This report will focus on the development of this device and how it can be used to control the robot's manual guidance mode.

1.3 DOCUMENT STRUCTURE

This report was divided in 5 chapters: Chapter 1 presents the motivation, objectives and framework of this work, Chapter 2, contains an analysis of the state of the art regarding the technologies used in this work, Chapter 3, shows the Collaborative Applications System Development, Chapter 4 is where the experimental tests are presented, along with the respective results, and, finally, Chapter 5 presents the conclusions and future work.

2

STATE OF THE ART AND SUPPORTING TECHNOLOGIES

This chapter presents a comprehensive state of the art on the supporting technologies used in this report, namely the ones related with the use of the KUKA LBR iiwa collaborative robot. Concepts such as collaborative and sensitive robots, as well as some well-known brand robots, bluetooth devices for wireless communication and Arduino-based solutions will be presented.

2.1 COLLABORATIVE ROBOTS

When it comes to industrial robots, it is important to analyze three main technical standards:

ISO 10218-1: 2011- Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots, this standard specifies the requirements and guidelines for the inherent safe design, protective measures and information for use of industrial robots. It describes the basic risks associated with robots and provides requirements to eliminate or reduce them [17].

ISO 10218-2: 2011 - Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration, this part of the standard specifies safety requirements for the integration of industrial robots and industrial robot systems as, defined in the ISO 10218-1 [18].

ISO/TS 15066: 2016 - Robots and robotic devices - Collaborative robots, here it is specified the safety requirements for collaborative industrial robots systems and their work environment, and supplements the requirements and guidance on industrial collaborative robot operation [19].

Collaboration is the action of working with someone to produce or create something, or achieve shared goals [10]. A collaborative robot, also known as a cobot, is a robot that has been specifically developed to work with a human operator to perform a specific task, to achieve a common goal in a collaborative workspace, such as assembling a circuit board into a factory. or even help a doctor with a surgical procedure in a hospital. According to the ISO 10218 and the ISO/TS 15066, a collaborative application can use 1 or more of the following techniques (Figure 2):

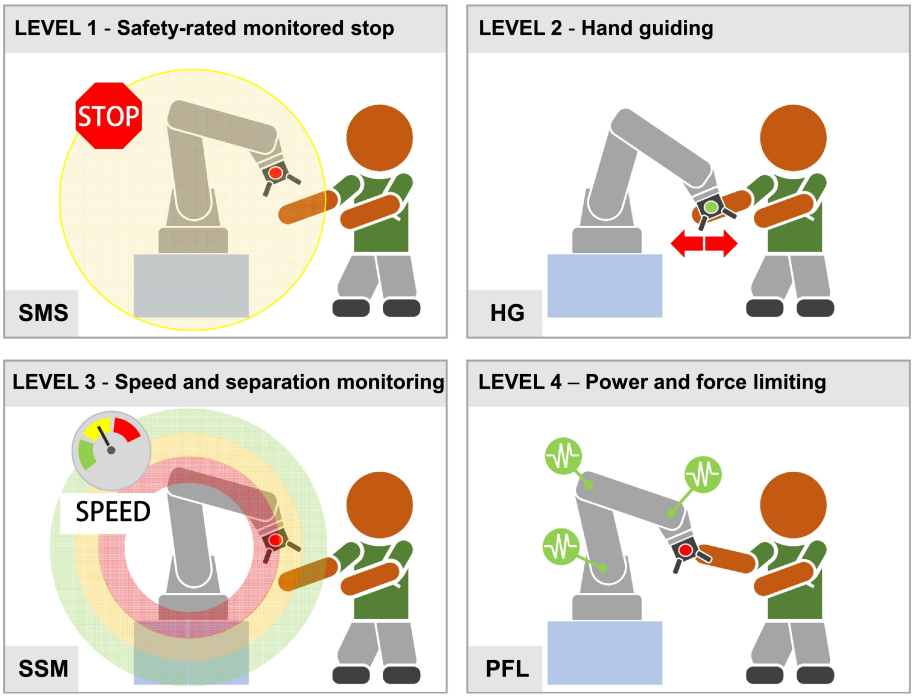


Figure 2: The four collaborative operative modes identified by robot safety standards 10218-1/2:2011[39].

Safety monitored stop is perhaps the simplest model of operation. In this mode, the interference of the human operator with the robot is minimal. This type of collaboration uses a traditional industrial robot in conjunction with safety devices, such as a laser sensor, which detects the employee's entry into the designated robot zone. If an employee is detected entering the robot work zone, the robot stops, and the employee can perform any necessary work operations, and then order the robot to resume its main task at the push of a button. This type of collaboration is often used when a large industrial robot is needed due to high loads, but a secondary operation must be performed by an operator, for example, a task where a robot takes a heavy piece of metal from one place, and places it in another, but a human operator needs to go into that environment and do something to the part, at which time the robot will stop, the operator will do the operation, and then inform the robot that it can return to its normal task [10].

Hand guiding is when an operator can guide the collaborative robot arm by hand to program the robot to perform new tasks. This allows for quick reprogramming to minimize downtime, as well as reducing the need for robot operators to have specialized programming knowledge. It should be noted that, if the robot is not a force limited robot, proper safety guarding and logic should still be in place for regular operations. The tools used by the robots also need to be taken into consideration in this mode of operation,to avoid risks for the operator.

Speed and separation monitoring is the type of collaborative operation that also uses traditional industrial robots, but it is more suited for environments where employees will be frequently interacting with the robot. In this type of installation, the area around the robot is constantly monitored by a safety-rated monitoring sensor, which can detect employee proximity to the robot. If the employee enters the warning zone, the robot slows to a safe speed and, if the operator enters the stop zone, the robot pauses until the employee has left the zone. Once the employee leaves the zone, the robot automatically resumes operation. This mode of operation is better than safety monitored stop in instances where frequent employee interaction, because the robot will automatically determine the safe running speed based on the location of the operator [10].

Power and force limiting is this type of operation which allows work between humans and robots without additional safety and process interruption devices. In this case, robots are designed with collaboration in mind, meaning they do not have sharp corners, exposed motors or tightening points, in order to increase the degree of safety of the operators who will interact with the robot. These robots have sensitive force monitoring devices and generally have a padded ”skin” to dissipate the force in the event of a collision. These robots work alongside humans and stop instantly if any collision is detected. This means that no vision systems, laser scanners or fences are needed when the robot is set up correctly. Often, the extra cost of limited strength robots is offset by the cost savings of not having to purchase and program a network of security scanners. Currently, these force-limiting robots are limited to applications smaller than 35 kg [10].

2.2 SENSITIVE ROBOTS

According to the Oxford Dictionary, something sensitive is something capable of detecting or responding quickly to small changes, signs or external influences [22]. Sensitive robots are capable of detecting the environment around them by means of sensors, such as thermal sensors, torque and distance sensors. Sensitive robots are robots with a Stigmergic sensitivity level (SSL), which is expressed by a real number in the unit interval of 0 to 1 [28]. This means that robots with a lower value of SSL have a greater capacity to explore environments and greater independence. Next section shows the main characteristics of the robots that lie within this category.

The IRB 14000 YuMi robot, (Fig 3) is an ultra compact and lightweight design robot, with two arms with seven joints each, allowing for collision-free access to objects in a constrained working environment. It weighs a total of 38 kg and has 0.02 mm of repeatability, while the payload of each arm is 0.5 kg within a reach of 559 mm, each arm has a multi-tooled gripper that can be used in different applications. YuMi enables safe assembly operations in collaboration with human workers with sensors to see and recognize objects and interact with the environment. YuMi’s

Figure 3: IRB14000 YuMi - ABB[40].

arms are like a human arm in terms of mobility and flexibility of movement, and its magnesium skeleton is covered in a floating plastic casing and wrapped in a soft padding. The YuMi's software was built upon existing ABB controllers with new interactive teaching capabilities, and to ensure safety for both the human worker and the robot and its sensors will detect the force in the event of a collision, and will stop the robot in milliseconds [40].

Figure 4: CR-7iA - Fanuc[9].

The Fanuc CR-7iA (Figure4) is small and can work side by side with a human. It takes care of light (up to 7 kg) but tedious and repetitive tasks that include different types of material handling, which would otherwise consume an immense amount of operator time. These tasks can range from small parts assembly, to highly repetitive tasks, such as picking and placing items from one place to another. Its long reach of 717 mm makes it the ideal candidate for machine tending and palletizing applications. This robot can work side by side with humans or collaborate with them without the need for external safety devices. It has collision sensors that stop the robot after colliding with a fixed object or human operator, so it is not necessary to use safety fences to delimit your work space, thus, in addition to saving space, it also reduces the costs of manufacturing. Depending on what you want to do, the CR series can be equipped with a FANUC R force sensor that measures force and torque, allowing the robot to perform collaborative assembly, fitting and weighing operations, for example [9].

Figure 5: duAro2 - Kawasaki[12].

Like a human being, the Kawasaki duAro2 (Figure 5) has two arms and is able to work side by side with humans in the same environment, thanks to the various functions that ensure safety and to the use of soft materials on the surface of the arm. In case of collision with the worker, the collision detection function will make the duAro stop. Its arms have a vertical stroke of 55 cm, and can lift a maximum payload of 3 kg on each arm. In addition to the controller integrated into the arms, this robot can be mounted with separate arms and controller, thus allowing a free layout on the production line. The advantages of double-arm robots are that they allow handling of various types and sizes of workpieces at the same time, unlike single-arm robots. In addition, the coaxial construction allows the robot arms to approach and work from the rear. Each arm can perform different tasks simultaneously, considerably reducing cycle time. With one arm, the robot can tighten the screws of a plate with a screwdriver and with the other, hold a workpiece [12].

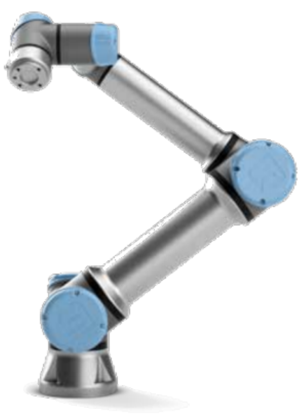


Figure 6: Ur5e - Univaersal Robots[13].

The Universal Robots, UR5e (Figure 6) is the medium-sized member of the UR e-Series family, wich has 3 members, the UR3e, UR5e and UR10e. Each cobot has a different reach and payload, but they share the same precision, accuracy and dependability. The UR5e is ideal for automating low-weight processing tasks with its 5 kg payload and 850 mm reach radius, is able to perform a 360 degree rotation on all wrist joints, and an infinite rotation on the end joint, and has a weight of 20.6kg. It has a built-in force and torque sensor with improved precision [13].

Figure 7: Gen3 - Kinova[20].

The KINOVA R Gen3 is an ultralight weight robot, (Figure 7) has an optional vision module, integrated with 2D and 3D sensors and, has torque sensors in all 7 joints. With a weight of 8.2 kg this robot can reach up to 902mm with 4kg payload, its body is composed of carbon fiber and aluminum [20]. This robot has two actuator sizes, one small and one large, with each actuator having a torque, current and temperature sensor on each motor phase [20].

Figure 8: Panda - Franka Emika[26].

The Panda (Figure 8) is designed to act like a human arm with high sensitivity in all its 7 joints. Its arm is inspired by the agility of the human arm and is capable of lifting a 3kg payload and has a reach of 850mm. Its hand, which is a very sensitive tweezer, has a continuous grip force of 70N, with a stroke of 80mm. To "teach" an action to Panda, someone can guide the robotic arm manually through a series of actions, while pressing buttons on the arm, that commit each step to memory. In just a few minutes, the arm learns the task and can start repeating it, without the need for complicated programming. The user can also choose a series of pre designed apps to launch the arm on other jobs. It has a modular shape, that makes it easy to swap in parts for other tasks, the arm is highly sensitive, making it aware of the presence of other people or objects [26].

Figure 9: TM12 - Techman[38].

Designed to work with both humans and machines the TM12 (Figure 9) offers a 12kg load capacity, 1300mm range and has 6 joints. This robot complies with ISO 10218-1 and ISO / TS 15066 robot-human cooperation safety requirements for collaborative robots, enabling the robot to be programmed with speed and strength limits [38].

Figure 10: LBR iiwa - KUKA [21].

The KUKA LBR iiwa version has two models, the LBR iiwa 7 R800, and the LBR iiwa 14 R820, both of which are lightweight collaborative robots made of aluminum, and have 7 joints. They have torque sensors that contribute to the operator's safety, allowing him to work with humans. The torque sensors allow the robot to respond to external forces and provide secure protection against collisions. In the event of an unexpected contact, the LBR iiwa reduces its speed, thus limiting its kinetic energy to a level that prevents injury.

The KUKA R LBR iiwa can handle up to 7 kg payload, and only weighs about 22.3 kg, with a reach of 800mm, while the KUKA LBR iiwa 14 can handle up to 14 kg payload, weighs about 29.9 kg with a reach of 820mm. Both models can work with a repeatybility of 0.1mm. The best characteristics of these robots are their actuators, that allow them to reach a maximum torque of 176Nm at joints 1 and 2, 110Nm at joints 3, 4 and 5, and 40Nm at joints 6 and 7 in the model R800, while in the other model, the R820, is possible to reach 320Nm at joints 1 and 2, 176Nm at joints 3 and 4, 110Nm at joint 5 and 40Nm at joints 6 and 7, making these robots a good choice for applications that require great sensitivity such as surgical procedures [21]. Another interesting study was the use of robot skin, as a potential enabler, is able to drive the development of cobots to deal with secure collaboration, immersive teleoperation, affective interaction. In addition, the robot's skin characteristics of satisfying inherent safety issues, sensory feedback, natural interaction and energy autonomy were analyzed.

2.3 COLLABORATIVE AND FORCE SENSING APPLICATIONS

Research with robotic manipulators using force sensing and impedance control gained wider adoption with the comercial introduction of the sensitive robots described above. One of these more recent works proposes the use of a deep a Q-learning algorithm in a KUKA iiwa robot, which makes use of visual perspectives and force detection, to learn how to assemble a plastic fixture in a low-voltage device. An image model corresponding to the assembly state is used by a reward system which, with visual feedback, analyzes whether the assembly operation is complete. Force sensors are used to detect the softness or stiffness of objects, and the contact state of the assembly in the assembly operation [34]. Another interesting study was the use of robot skin as a potential enabler, to drive the development of cobots to deal with secure collaboration, immersive teleoperation and affective interaction. In addition, the robot's skin characteristics of satisfying inherent safety issues, sensory feedback, natural interaction and energy autonomy were analyzed [27]. Recently, flexible manufacturing processes have become an important requirement in factories that need to produce customized products. The customized product goes through the various manufacturing processes, where optimizing the time needed to transport these objects becomes as important as the layout of the workstations. To improve the overall capability of the system, an integration of two robot systems to work in collaboration with a human worker has been proposed. An autonomous drone can be used to travel in 3D space, allowing access to hard-to-reach areas, while an autonomous mobile robot can be used to efficiently traverse flat terrain and transport heavier objects [31]. Most robots have little ability to collaborate and rely on expensive sensors for force information. To solve this problem, a human-robot collaborative assembly scheme, without the use of additional force/torque sensors was proposed. The coupled load force and the human-robot interaction force affects the balance and stability of torque-controlled robots. To solve this problem, a method of identification and load compensation has been proposed. Considering load and friction compensation, an impedance control algorithm was used to adjust the robot according to the demands of the task, helping the robot to complete an assembly task and also making the robot complacent when encountering disturbances or collisions [35].

2.4 ARDUINO BASED SOLUTIONS

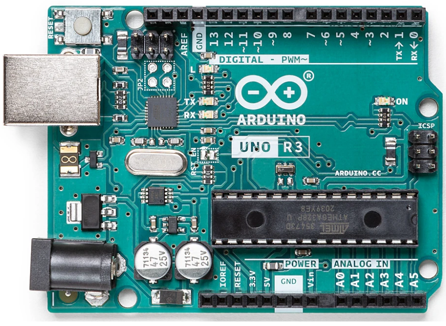


Figure 11: Arduino Uno Rev3 [3].

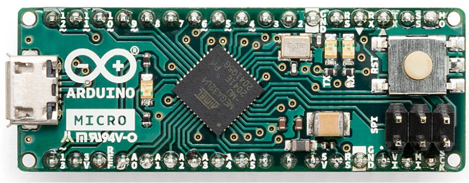


Figure 12: Arduino Micro [2].

Arduino is an open-source electronics platform based on easy-to-use hardware and software [1].

A robotic arm is useless without the needed tools and peripherals.The right tools and peripheral greatly extend the possible use cases, or the ease with which the applications are developed. The robotic arm being used in this project is in an academic environment, thus being applied in many different applications. As such, it is both important to make it more userfriendly, and to allow more flexibility in terms of Inputs/Outputs (I/Os), communication and other sensor or actuators connections,particularly on its flange. Of the possible approaches to increase this flexibility and connectivity on the robot flange is to develop a specific electronic-based adapter that fits the flange, using a programmable microcontroller as the core processing unit. Although many possibilities exist, the Arduino family of microcontrollers allows for an easier development and faster implementation, due to its open-source nature, large community and documentation assets.

2.5 ROBOTIQ 2F-85 GRIPPER

The 2F-85 Adaptive Gripper (Figure 13) is a gripper suitable for collaborative robots. This gripper has an opening of 85mm between fingers, has a force control that varies between 20N to 235 N, a speed that can vary from 20 to 150mm/s, and can support up to 5kg [30].

Figure 13: Robotiq Adaptive Gripper 2F-85.

2.6 SUNRISE SYSTEM

KUKA Sunrise workbench is an integrated development environment for programming the KUKA's range of collaborative robots. Sunrise is used to configure and manage the KUKA iiwa robots. It also provides tools for coding applications to run on the robots, written in the Java programming language. Another software package, WorkVisual, is used for the bus configuration and bus mapping. The Robotiq 2F-85 gripper and the Beckhoff modules used in this project are configured with Workvisual. For more details how this process is made, see Appendix C.

3

COLLABORATIVE APPLICATIONS SYSTEM DEVELOPMENT

This chapter presents the components chosen for the development of the hand guiding device mentioned on chapter 1 for the KUKA robot. That includes the characteristics of the idealized system, such as its resolution and the purpose for which it was developed. The device can be used not just for handguiding operations, but was also designed to be usable in many different applications. This includes developing a device that is simple but not limited to the applications of this project and that can be easily adapted to match future specifications.

3.1 MULTI-PURPOSE FLANGE ADAPTER SYSTEM

As has been stated before, one needs to develop a hand guided device to use the handguiding mode on KUKA robot. First of all, it is important to understand how this mode works on the robot and how to activate it. In the KUKA robot there are external enabling devices that can be connected via the safety interface X11 (Figure 14) of the robot controller, and can be used to activate the manual guidance of the robot. When the device is active, the robot may only be moved at reduced velocity [15].

Figure 14: X11 interface on Kuka Sunrise Cabinet.

The intended system must be simple and easy to use, in such a way that it can be operated without requiring high technical expertise. The bare robot system is composed of the KUKA robot and its control unit, with the idea being to develop a device that complements this system. Figure 15 shows an overview for the integration of the device being developed with the current system. This system, is composed by the KUKA robot (1) which communicates with the KUKA Sunrise Cabinet robot controller (2), with the addition of the Multi-purpose Flange Adapter System, composed by two modules, the Flange Module (3), that is coupled to the robot's flange, and the Controller Module (4), that will receive the signals sent by the first module, and will make the connection with the robot controller using the Beckhoff input and output module (5) and the Safety cable. The Adaptive Gripper (6) is connected to the Robotiq controller (7), which is connected to the KUKA Sunrise Cabinet robot controller. The Flange and Controller modules form the device that guides the robot, which, in addition to putting the robot in manual guidance mode, will also be responsible to inform the robot controller when it is necessary to do some operation with the gripper.

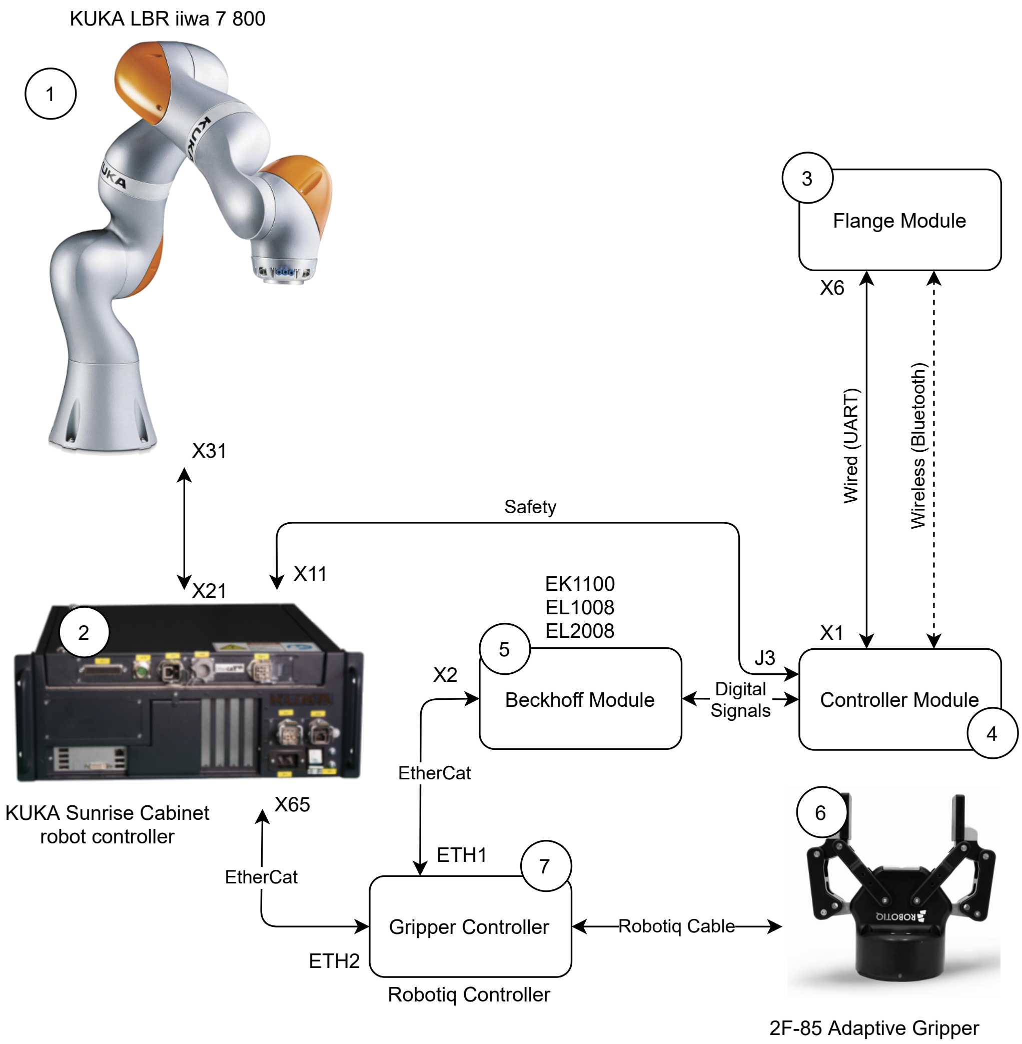


Figure 15: Overview of the handguiding system.

To develop the flange adapter device for the robot, one needs to decide what will be needed and what are the possible difficulties associated with the project. As already mentioned, this device is divided into two modules, one that will be coupled to the robot's flange, and the other which will be close to the robot controller. Each module will be described separately in this chapter, where the specifications and functions of each module will be presented.

3.1.1 *Flange Module Specifications*

Taking into account that the user needs to interact with the device, it will be necessary for the user to use some input on the device, which could be a button to activate the manual guide mode, and another button to activate the gripper, for instance. It would be good for the user if the device provides a visual feedback to indicate some operating status, and it should have a controller to manager all the features, such as communicating with the other part of the system, which is responsible for activating the robot mode. As such, the device must have the following specifications:

- 2 input ports
- 1 output port
- 1 microcontroller
- 1 wired and wireless communication

The input module will be coupled to the robot's flange, so the dimensions of the flange must be considered for the layout of the circuit to be developed, to make sure it is within the dimensions of the flange, allowing the device to be more discreet when coupled to the robot.

The flange of the KUKA robot, show in Figure 16, has a diameter of 63mm, which corresponds to an area of $3{,}117.24 mm^2$. However, the flange is not totally free from obstacles, as shown in Figure 16, it has 8 holes equally spaced, forming a 45-degree angle between two holes and the center. Of the 8 holes, the hole highlighted in the figure is a guide and has 6mm in diameter, while the other 7 have an internal M6 thread. Therefore, the components required for the project must be compact enough to fit within the working area (see Figure 17), which has a maximum diameter of 104 mm.

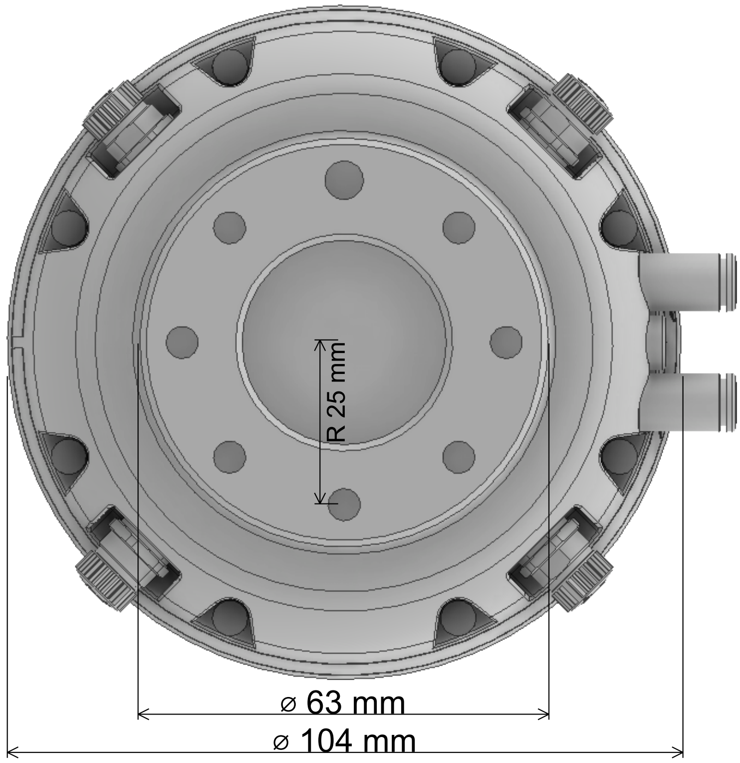


Figure 16: Kuka flange layout.

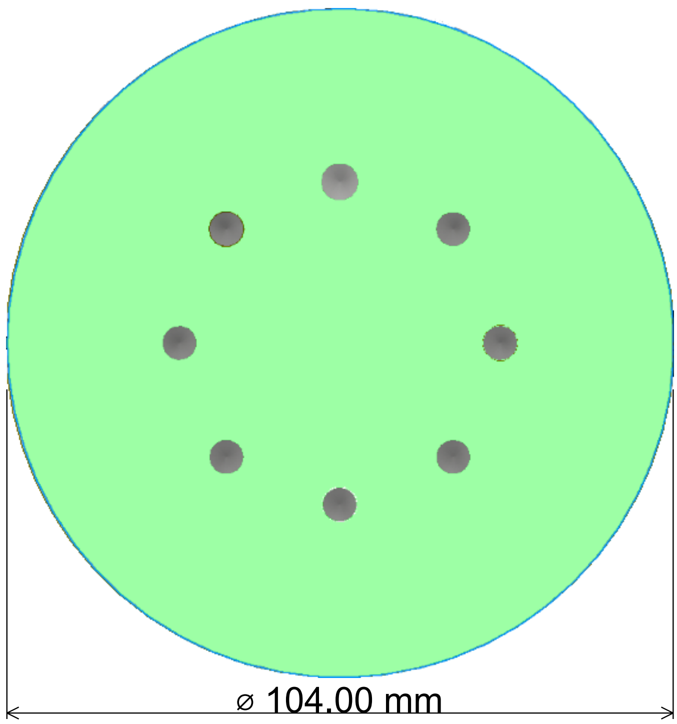


Figure 17: Kuka flange work area for the PCB development.

3.1.1.1 *Microcontroller*

Arduino's biggest advantage over other microcontroller development platforms is its ease of use, which allows developers with little experience to create simple and complex projects in a relatively quick time. There is a huge community of users who are available to help and offer information to newcomers and a good place to find answers is the arduino forum. Arduino has different models with different specifications and even sizes. In such a way, the developer can choose the model that best suits his specifications without having to give up the integrated development environment (IDE) that serves for each different model. The working area (Figure 17) implies the use of a microcontroller such as the ATmega32U4 which have only 9mm, which facilitates its use on the board. In addition, the ATmega32U4 does not need usb/serial converter circuitry, which further reduces the minimum amount of components that need to be added to the board for it to work [4].

Another important feature of this microcontroller is that it has 20 input and output pins, 7 of which can be used as PWM, 12 as analog inputs, that is, it meets the device specifications and it still leaves a significant amount of resources free, such as inputs and outputs, available for future use.

3.1.1.2 *Visual Feedback*

One of the functions of a manual guidance device is to be able to indicate the device status visually. In this device, leds will be used to indicate the operating status of the device. In this case the SK9822 (Figure 18) led strip was chosen, which are leds that can be individually addressed using an easy-to-control SPI interface, allowing full control over the color of each RGB led. This led strip has two wires for the power and two wires for control, so, the ATmega32U4 just needs to use two outputs to control this strip, whereas one pin is used for data and the other is for the clock (Figure 19).

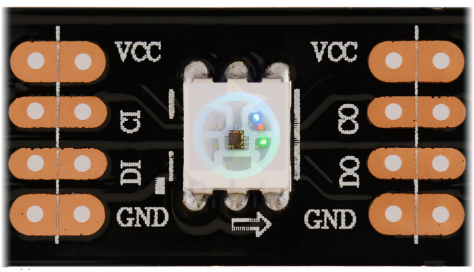


Figure 18: Single LED SK9822.

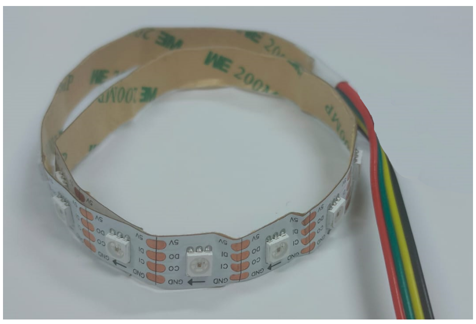


Figure 19: LED Strip.

3.1.1.3 *Communication*

The ATmega32U4 has two serial communication ports, one that is used exclusively by the USB (Serial) port and the other that is connected to pins 0 (RX) and 1 (TX) (Serial1) [33]. As such, it is possible to program the Arduino exclusively with the Serial port and the communication with another device can be done through Serial1, so a bluetooth module, or a cable connection, can be used to communicate with the devices.

3.1.2 *Flange Module Development*

This module was initially based on the circuit diagram publicly available for the Arduino micro, but with some changes. The robot controller works with voltages in the 24V range, therefore a 24V power supply is required. The Arduino Micro's reference circuit has a voltage selector, which controls where the power comes from when using an external power supply and the USB port power supply, whereas this circuit is essential to program the Arduino while it is powered by the external source.

The NCP117 voltage regulators of the original circuit can receive up to 20V at the input [32] , so it had to be replaced by the LM317 which support up to 37V [16]. The typical application example described in the LM317 datasheet was used.

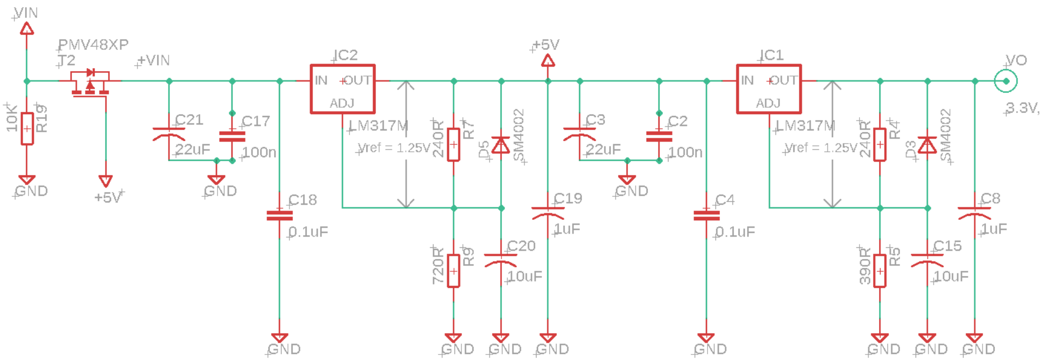


Figure 20: Power Circuit of the Flange Module.

The two mosfets in Figure 21 are used for the case when the board is powered by the USB port, so that the 5V of the port does not go through the IC2 regulator, but only through IC1. If the board is powered by both the USB port and the external source, the mosfet T1 stops conducting, causing only the external voltage to flow, passing through the IC2 regulator as shown in Figure 20. These transistors do not support voltages in the 24V range so, for this board, it was necessary to use a lower voltage to protect the mosfets, in this case, a voltage regulator module was used, the LM2596 module, which was adjusted to regulate the input voltage to 12V.

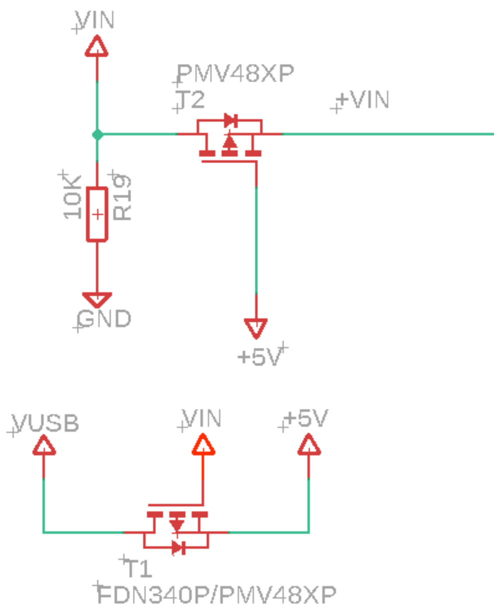


Figure 21: 5V Selector.

As 5V is needed for the Arduino power supply, and 3.3V to power the bluetooth module, the LM317 circuits were designed for these voltages, where through the following equation it is possible to calculate the value of resistance R2 for 5V and 3.3V.

$$Vo = V_{ref}\left(1 + \frac{R2}{R1}\right) + I_{adj}R2$$

Since I_{adj} is, typically, around 50μA and negligible in most applications, the second term of the equation can be neglected. By substituting the fixed values in the equation, it is possible to find the value of R2 for an output voltage of 5V, which corresponds to 720**Ω**, and for 3.3V where the value is 390**Ω**.

The HM-10 bluetooth module was used for module communication because, as can seen in Appendix A, this module was the easiest to use compared to the RN4678 and BM71, the others that were tested. The HM-10 works with a 3.3V and, on the TX and RX pins, since the ATmega32U4 communicates using 5V voltages, it was necessary to add a bi-directional logic level converter circuit based on the BSS138, as shown in Figure 22, to be able to successfully connect between the ATmega and the HM-10.

This circuit converts the voltages from 5V from the Arduino to 3.3V from the HM-10 and vice versa, so that both components can communicate with different voltage levels. Analyzing the reference circuits of the HM-10 datasheet [5], the following circuit was created, which represents the minimum configuration for the device to work in normal operation (see Figure 23).

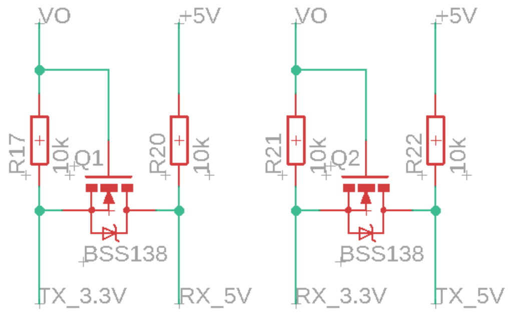


Figure 22: Bi-directional Logic Level Converter.

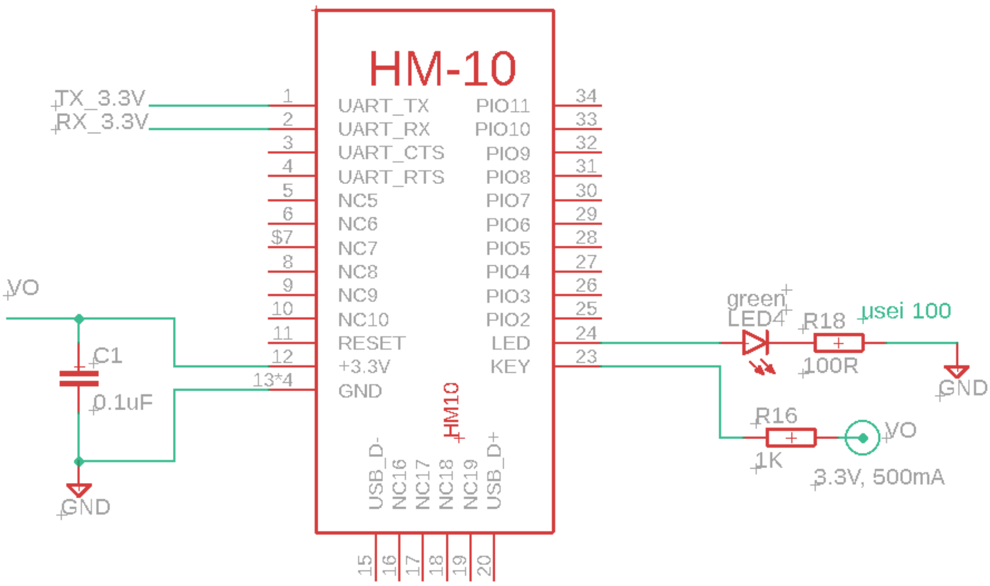


Figure 23: HM-10 circuit for normal operation.

Taking into account the workspace previously defined (see Figure 17), the board shown in Figure 25 was developed, with a diameter of 101.6mm, value within the proposed limit. On this board the ATmega was positioned slightly below the center, and the HM10 was positioned at the right of the board, with the antenna facing away from the board, as shown in Figure 25 and Figure 26.

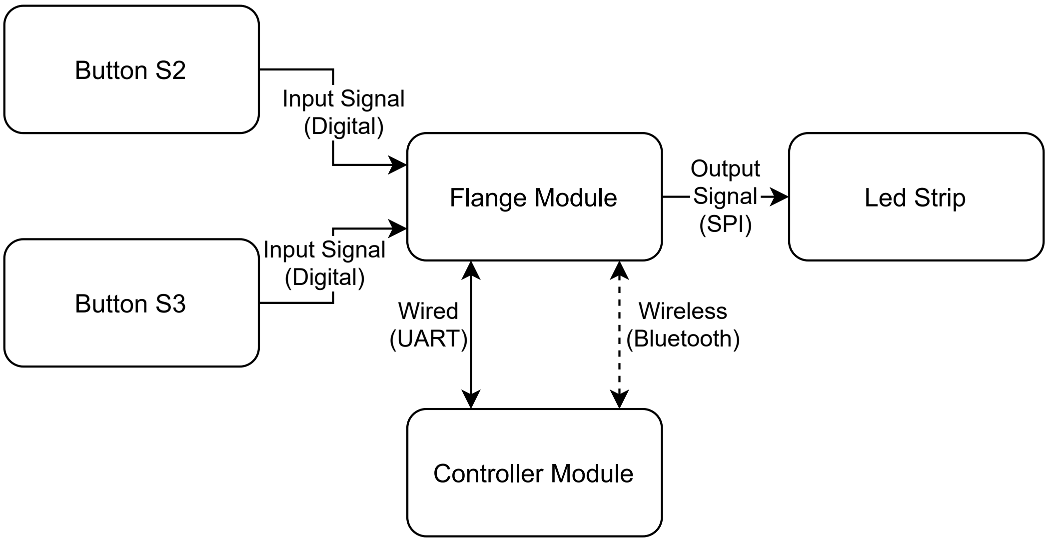


Figure 24: Flange module communications' diagram.

The diagram in Figure 24 shows how signal changes are made in the flange module. The module receives two input signals coming from the S2 and S3 buttons, which will control the robot handguiding mode and the gripper control respectively. In addition, an output of the module must control the LED strip to indicate possible status of operation. And you can see that there is communication, wired or wireless, between the flange module and the control modules.

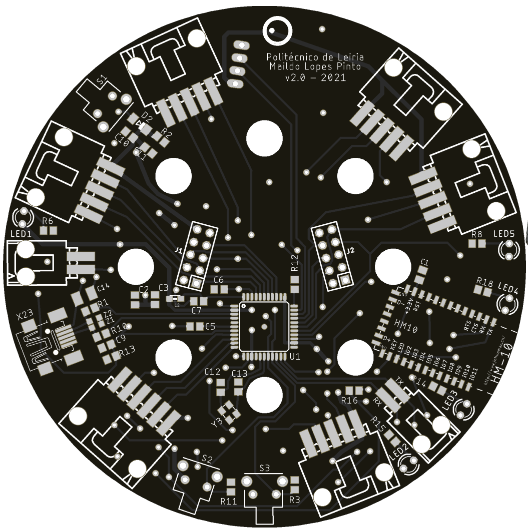


Figure 25: Layout flange module.

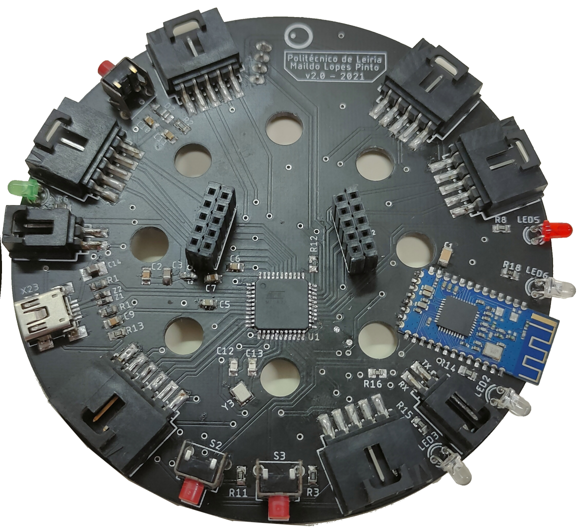


Figure 26: Flange module PCB.

This board has 6 molex 70634-POS5 connectors to provide lateral inputs and outputs of the ATmega that, in future works, can be easily accessed for various functions. Three buttons were made available on the side, one of them for resetting the ATmega and the other two to activate the handguiding mode, and to activate the F-85 gripper. Two molex 70634-POS2 connectors positioned on the side were also used, one for the board's external power supply, and the other to provide the TX and RX pins for possible wired communication with another device. For more informations about the module schematics and layouts, see appendix E.

The software development is done using the Arduino IDE. At this stage, you have to be careful with the ATmega specifications acquired because, for the microcontroller to be programmable, it needs to have a bootloader burned, if not, it is necessary to burn the bootloader performing the procedure described in Appendix B.

As in the hardware development, the software also has its specifications:

- The connection between the Flange module and the Controller module must be established whenever possible and, in the event of a loss of signal, the Flange module should try to reconnect to the Controller module;
- Each time the buttons are pressed to execute a function, the corresponding output must be activated.

The flowchart of the software is shown in Figure 27. Note that, when the module starts, the first thing to be done is to check the connection status with the control module, and, if there is no connection, the Flange module stays in a loop trying to connect to the module, doing the following sequence: search for devices, then check if the controller module is in the list of available devices; if it is, it connects to the module, if not, it repeats the cycle. If the connection has been established, every time a button is pressed, a command is sent to the controller module with the information of which button is pressed or released.

3.1.3 *Flange Module 3D Case*

To complete the development of the flange module, a 3D case was designed to accommodate the developed board and to be able to fix it on the flange of the KUKA iiwa robot. Figure 28 shows the case that was developed with a cylindrical shape and no sharp corners. It is 37.2 mm high and has a diameter of 104.5 mm, with its lower part being a replica of the robot flange fitting, thus the Flange module can be used between any device that was originally fitted to the robot's flange, such as the 2F-85 gripper, which can now be attached to the flange module.

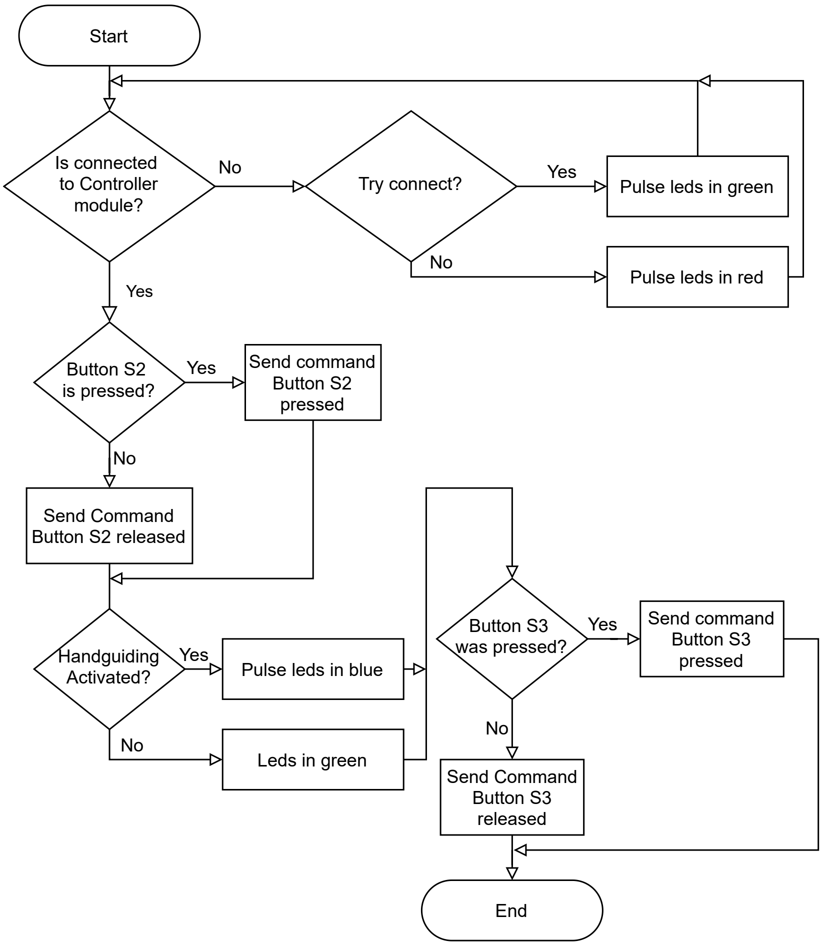


Figure 27: Diagram of the software on the Flange Module.

Figure 28: 3D Case of the Flange Module.

3.1.4 *Controller Module Specifications*

The Controller module no longer has the same restrictions as the first, the Flange module has a maximum area to be built (Figure 17), while the second module does not have a specific place to be placed, so it can have any shape and size.

3.1.5 *Controller Module Development*

Without the space limitations, it is not necessary to develop a board with an ATmega separately, here it is possible to use the Arduino micro module and just develop a slot for it on the board itself, which simplifies the board development. In this module, another HM-10 bluetooth module was also used to establish a wireless communication with the Flange module. As space here is not as restricted as in the Flange module, a complete HM-10 module will be used without the bidirectional logic level converter circuit, as the HM-10 Module already has this built in. Three TBL009-254-10GY-2GY connectors were used to make the Arduino's inputs and outputs available for future work with the board.

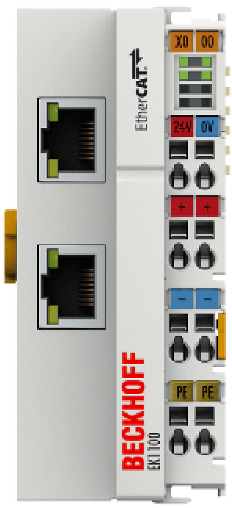


Figure 29: EK1100 [11].

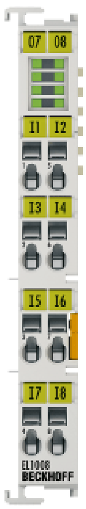


Figure 30: EL1008 [36].

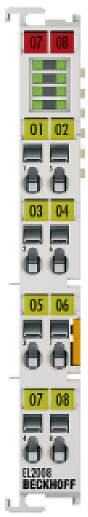


Figure 31: EL2008 [37].

The communication between the Controller module and the robot Controller was made by the EK1100 EtherCAT Coupler, shown in Figure 29, plus an EtherCAT Terminal module, 8-channel digital input EL1008 (see Figure 30) and another EtherCAT Terminal module, 8-channel digital output EL2008 (see Figure 31). These components operate on 24V, so the controller module must send 24V to an input on the EL1008 for it to work. And for safety, the board voltages were separated from the Beckhooff module voltages through an optocoupler, the TPL521 that supports voltage up to 55V. To enable the manual guidance mode, it is necessary to use a Safe Input of the Cabinet Interface Board, Small Robot (CIB_SR) on the KUKA Sunrise Cabinet robot controller. The inputs of the CIB_SR are of dual-channel design with external testing, and the test outputs A and B are fed with the supply voltage of the CIB_SR. To enable the handguiding mode, the signal test output A must be connected with an input A and a test output B with an input B. So, to make these connections, the TPL521 optocoupler was also used. These signals must not be significantly delayed, so the Controller module must activate both signal at the same time.

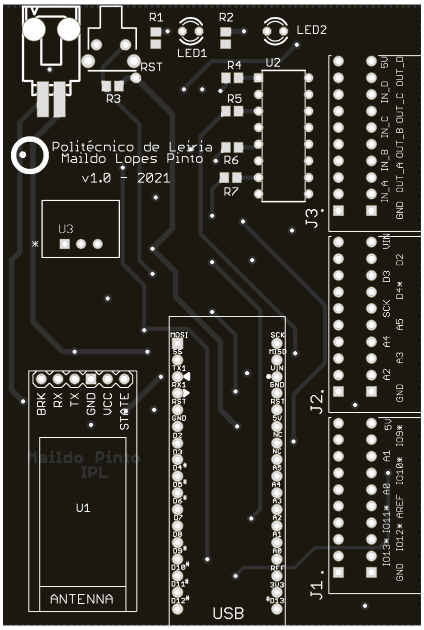


Figure 32: Layout controller module.

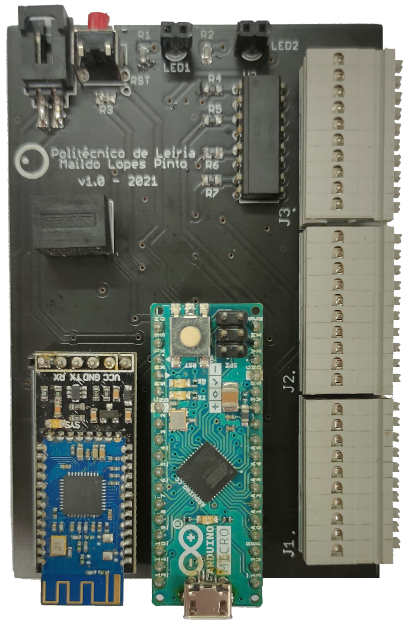


Figure 33: Controller module PCB.

Figure 32 and Figure 33 shows the arrangement of the components on the board. This board has a size of 95mm x 64mm. The HM-10 was positioned at the bottom near the edge of the board, to improve the quality of the Bluetooth signal. Beside it, two 17-pin female connectors were positioned to receive a Arduino micro module. At the, top 2 LED were positioned, one to indicate that the board is powered with 5V, and the other indicates the Bluetooth status, in other words, if it is connected or not to another device. For more informations about the module schematics and layouts, see appendix E

The controller board software was developed based on the following specifications:

- If a connection has been established, received commands must be processed and the corresponding actions must be taken;
- For security reasons, if the connection was lost, the output that activates the robot's handguiding mode is turned off.

The structure of the software is shown in Figure 34. When the program starts it enters a loop that is continuously checking if any connection has been established. If no connection is made, the outputs that activate the robot's handguiding mode are

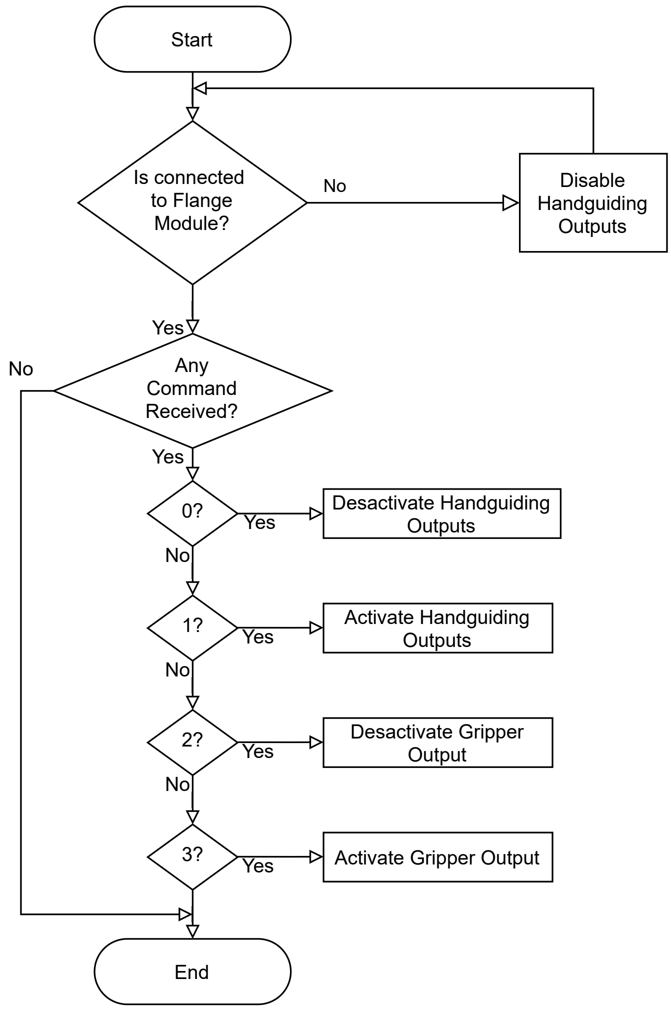


Figure 34: Diagram of the Software on Controller Module.

disabled. Only after the connection with the flange module is established, does the program verifies the received commands and, if a function is requested, the output for that function is activated.

3.1.6 *Controller Module 3D Case*

As was done for the previous module, a 3D printed case was also developed for the Controller module. The main difference is that this module will be used in an electrical panel, so it had to be developed with this in mind. The case (Figure 35) developed is 102.0 mm high and 70.0 mm wide, where two holes were left in the cover to couple the two LED. On the side there was a tear to connect the wires to the connectors and, a din rail connector was designed on the outside, in order to fix the case on the din rail of the electrical cabinet.

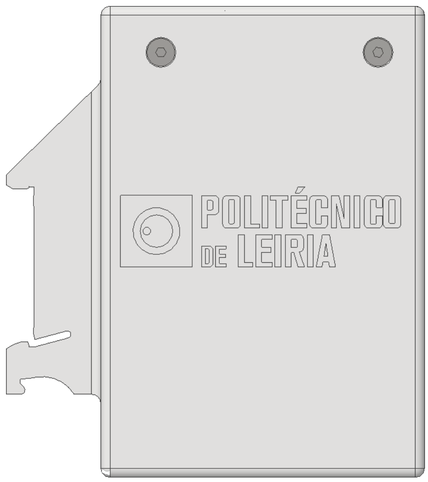


Figure 35: 3D Case of the Controller Module.

3.1.7 *Flange Module Assembly*

This section presents the precautions taken when mounting the flange module on the robot. The module power supply is 12V and is supplied by the voltage regulator located inside the electrical panel, described in the following section, this power reaches the flange module by cable WX651 mentioned in the electrical diagram in Appendix D, this cable is connected to connector X651 on the base of the KUKA robot that leads to connector X13 on the robot flange, therefore, the module is mounted so that the module's power input, connector X1, is aligned with flange connector X13, allowing for a shorter connection between these connectors. Once aligned, the module can be fixed to the flange together with the clamp connection base, using 4 screws M6x30mm, as shown in Figure 36.

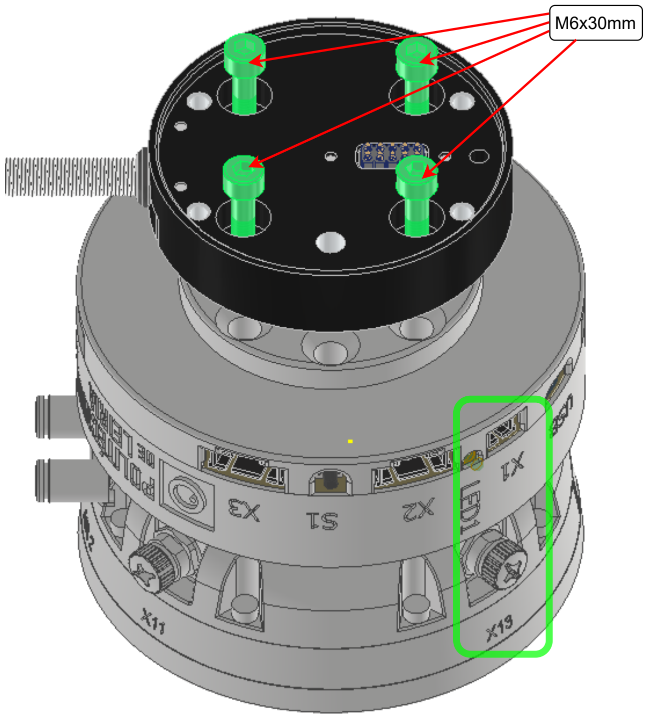


Figure 36: Assembling suggestion.

After the module is fixed to the flange together with the base of the gripper, now just fix the gripper to its base with 4 screws M5x25mm and the system is assembled correctly and ready to be used.

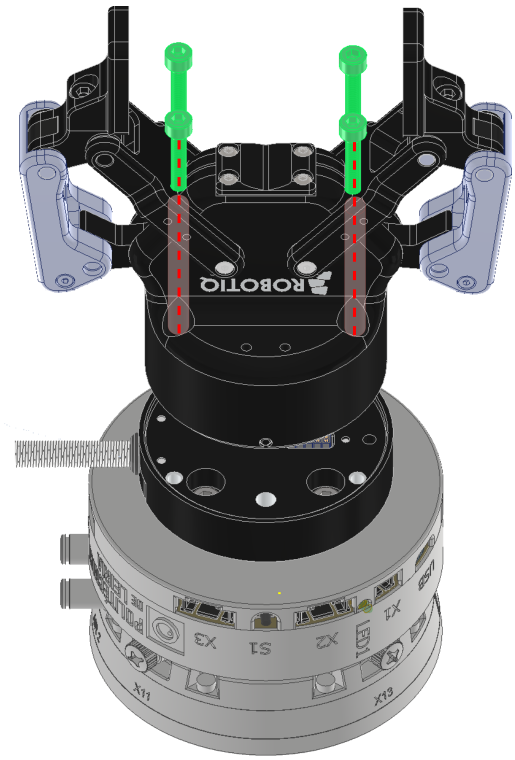


Figure 37: Gripper assembling.

3.1.8 *Electrical Cabinet*

The flange module is attached to the robot flange, but the control module and other devices are not, so an electrical cabinet was designed as shown in Figure 38 to accommodate all these devices. An electrical project was developed to describe the connections made in the electrical cabinet (see appendix D).

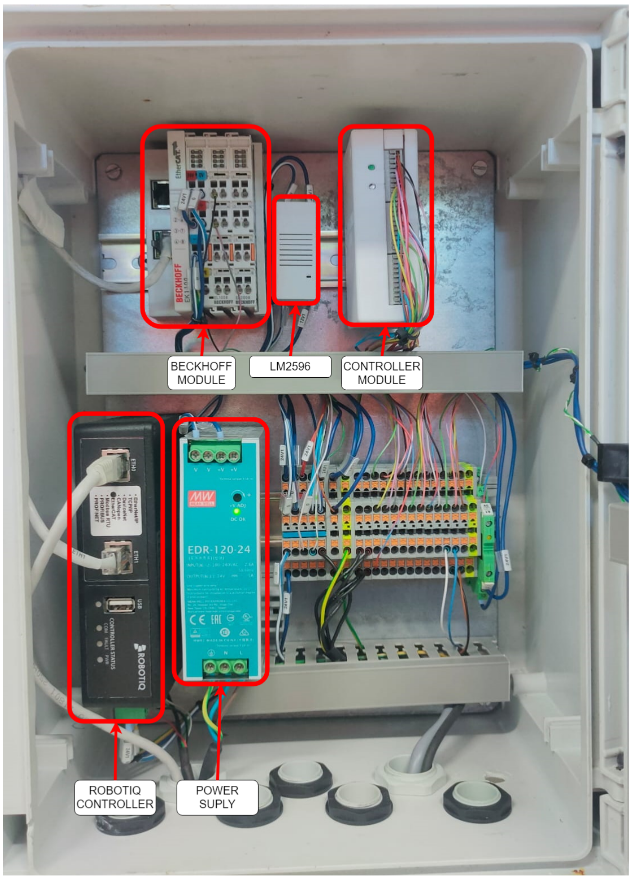


Figure 38: Electrical Cabinet.

3.2 MANIPULATION APPLICATIONS

The two modules developed together form the Multipurpose Flange Adapter System, composed of the flange module and the control module that form a complete and versatile device. With this system it is possible to develop numerous applications with greater ease, as this device is capable of enabling the robot's manual guide mode and, in addition, it is possible to make use of the ATmega digital and analog pins that are available in the connectors around the board. This chapter describes some applications developed for the KUKA robot using the KUKA Sunrise software, with the help of the Multi-purpose Flange Adapter System.

Frames are coordinate transformations which describe the position of points in space or objects in a station [15]. This is a concept necessary to understand the motions made by the KUKA robot. The position of a frame relative to its parent frame is defined by the offsets of the origin along the axes X, Y and Z of the parent frame and the rotational offset of the axis angles A, B and C of the parent frame. These rotational angles are:

- Angle A: Rotation about the Z axis
- Angle B: Rotation about the Y axis
- Angle C: Rotation about the X axis

3.2.1 *Handguiding Application*

As it was said in the introduction of this work on chapter 1, handguiding is a mode where the KUKA robot can be handguided by the operator. In this way the operator can somehow teach movements or points for the robot to follow. With the use of the Multi-purpose Flange Adapter System, the operator can control when to activate this mode on the robot but, to reach this mode, it is still necessary to prepare the robot controller for this, that is, it is necessary to load some safety settings and enable some options in the controller that will be shown below.

According to the robot programming manual, to use the KUKA robot manual guidance mode, at least 2 Event-Driven Safety Monitoring states (ESM) must be configured [15]. The first ESM chosen was for monitoring manual guidance devices and, in the safety settings, the chosen input signal was "CIB_SR.4". To enable the manual guidance mode, it is necessary to use the signals "Output Test A", "Input A", "Test Output B" and "Input B" which are available at pins 7, 8, 16 and 17, respectively.

Besides, it is necessary to choose, in the settings of the file "StationSetup.cat", in which modes the manual guidance can be used. Figure 39 shows a configuration where the guidance mode can be used in both automatic and robot test modes.

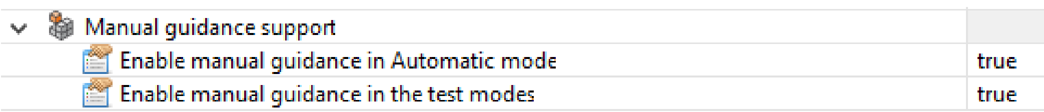


Figure 39: Manual Guidance Support on the StationSetup file.

After loading the robot with the minimum settings described above, and with the instructions provided by the manual, it was possible to develop two handguiding applications with different functionality.

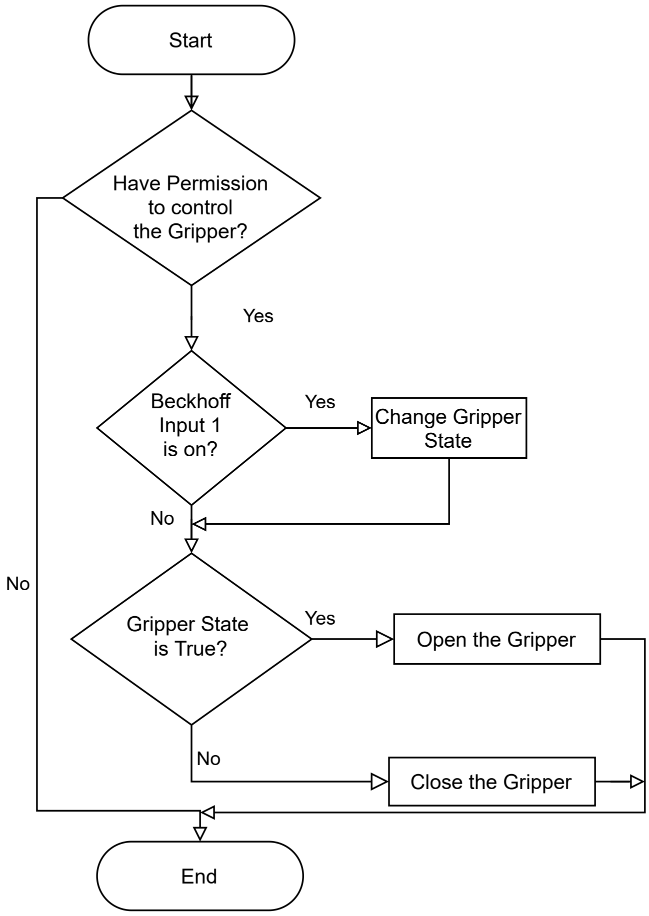


Figure 40: Background Task 1 Flowchart.

First, an application that runs in the background was created, based on the flowchart shown in Figure 40, that shows how it works. After turning on the application, it will run cyclically in an interval of 50ms, the application monitors if it has authorization from other applications to control the gripper and, if it has authorization, it will check the status of input number 1 of the Beckhoff EL1008 module. If this input is activated, the status of the gripper is changed and a command is sent to the gripper which will open or close, depending on the current state variable. Mechanical keys, like button, have an undesirable behavior called "bouncing", which is when the microcontroller interprets multiple taps when in fact the button was only pressed once. As such, a software debouncer was used to control the status of input 1 of the Beckhoff EL1008 module.

The first application for handguiding was created in such a way, that the operator teaches points in the workspace to the robot, where the position of the robot and the current state of the gripper are also taken into account.

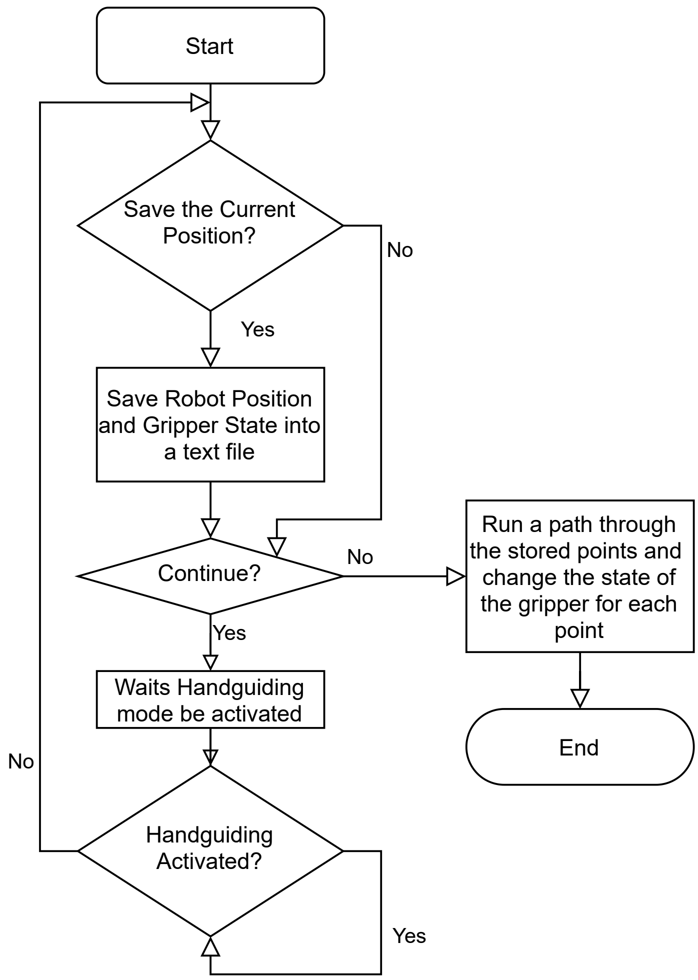


Figure 41: Handguiding_Points Application Flowchart.

The flowchart shown in Figure 41 exemplifies how the Handguiding_Points application works. When the operator starts the application, a menu is displayed on the robot's SmartPad screen asking if the robot's current position and the current state of the grip should be stored by the application. If the answer is yes, then this data is stored in a text file by the application, then another question is issued, asking if the operator wants to continue the application and store more points or not. If the answer is yes, the message closes and the application waits for the handguiding mode to be activated and/or deactivated. At this point in time, the operator can activate the handguiding mode and position the robot in another position and, even open or close the gripper. After deactivating the handguiding mode, the first menu reappears and the operator, may or may not store the current point of the robot. Regardless of the choice, the second menu is shown again and, now it is possible to execute a path that is formed by the points stored previously, remembering that the robot will move to each point and, after reaching each point, the gripper state at that point is executed.

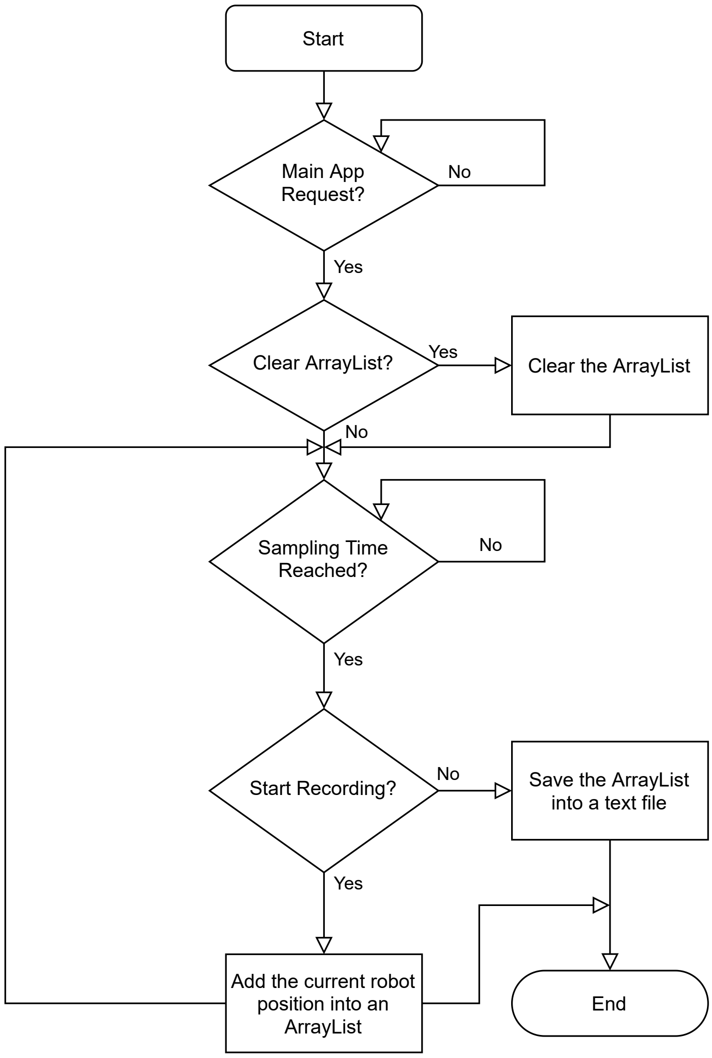


Figure 42: Background Task 2 Flowchart.

For the Handguiding_Path application to work correctly, another application was developed that runs cyclically in background with a 100ms period. Figure 42 shows the flowchart of this application, after the application is started, it waits for the request of some command coming from the main application. If the main application sends a command, this command is analyzed. If it is the "ClearFiles" command, the application clears the files containing the previously saved robot positions, then erases any possible value added on earlier cycle in the array list. If the clear command is not sent, then the command that will be sent before the others is an integer value ranging from 1, 3, 5 or 10, which corresponds to the sampling time. As the application runs in an interval of 100ms , at the end of each cycle it increments the variable "Sampling" by one unit, so, if the operator wants a sampling time of 1s, he just needs to send a value of 10 and the application will perform the following steps only after 10, interactions of 100ms .

After the sampling time is reached, the application will check if the command "isStartRecording" is activated, to be able to store the robot's current position. If so, the robot's current position is added to an array list. This cycle is repeated, creating a list with the various robot positions, that are cyclically added after the given sampling time. This repetition lasts until the application determines if the "isStartRecording" command has been disabled and, if so, it stops adding everything to the list and saves the values from the list to a text file, adding a specific character to indicate that a list of points ends there.

The application Handguiding_Path is an application where the operator, instead of teaching points as mentioned above, now he basically teaches paths to the robot, thus, in this application, the robot almost imitates the movements that the operator makes while driving the robot in handguiding mode.

Figure 43 shows the application operation flowchart, when the application is started, the operator needs to chosen between three options from the main menu. The first option "Start New", is choose in case the operator wants to start everything from scratch. Choosing this option, the application will send the command "ClearFiles" to the background task and the operator chooses the sampling time that will also be sent to the background task. After, the current status of the gripper is stored in a text file. The application starts monitoring ESM 1, which monitor if the hand guiding device is disabled, and activates the command "isStartRecording", at this time the operator can activate the handguiding mode using the Multi-purpose Flange Adapter System, moving the robot along a trajectory which points will be stored, according to the sampling time chosen by the operator. After deactivating the robot's handguiding mode, three other options appear, and the operator decides whether to press, "Continue" to continue and add one more trajectory to the previously saved one, or "Finish", to store the last state of the gripper and, go back to the main menu, or "Check Path", to check the previously saved files.

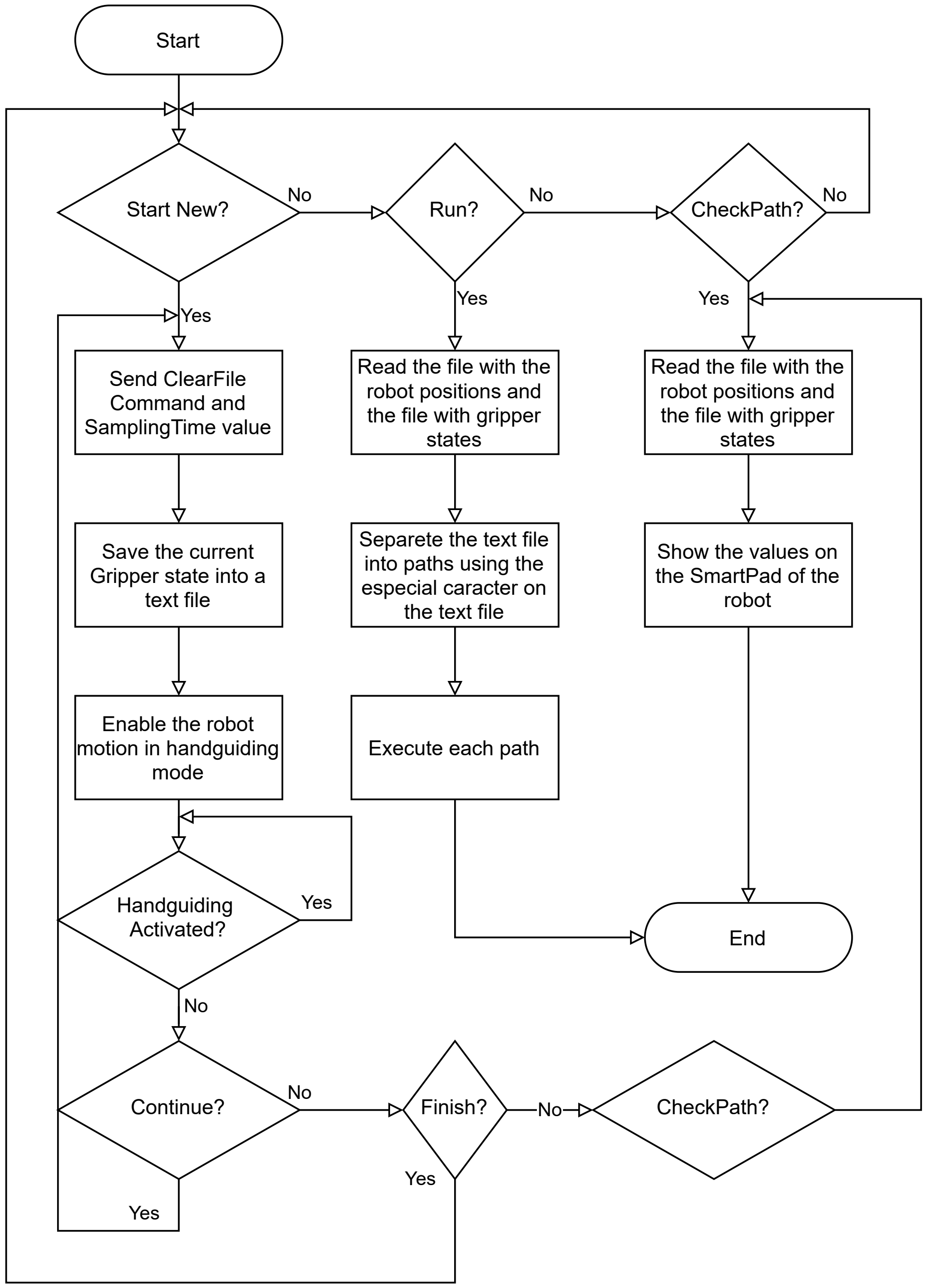


Figure 43: Handguiding_Path Application Flowchart.

In the main menu, the "Run" option is used to read the points stored in the text file, using the special character that was added at the end of each path to separate them, and also reads the gripper states. Remembering that the number of gripper states, is always a higher value than the number of saved paths, as there is always the initial gripper state, then a points path and, at the end, the final gripper state. Thus, if the operator has stored two paths, three gripper states have been saved, and now the robot will execute the movement as follows: first it returns to its initial position, then it executes the first gripper state, then it goes along the path, at the end of the first path it executes the second state of the gripper, then it goes along the second path and at the end it executes the third and last state of the gripper.

With these two handguiding applications developed, it is possible to demonstrate the functioning of the developed Multi-purpose Flange Adapter System. With this device it is possible to develop numerous applications with a shorter development time and in a simpler way, which will be shown in the following applications developed for the KUKA iiwa robot.

3.2.2 *Pick Up and Place Application using the Multi-purpose Flange Adapter System*

The pick and place task is very common in modern manufacturing environments. In this type of application, the robot usually grabs and moves objects from one place to another, it is a simple task but with great applicability. For example, pick and place robots can be used to take items for an order and place them in a box for packing, or can be used to take parts needed for assembly operations and move them to another place.

There are numerous ways to develop this type of application, one of them would be to create a program that knows the starting position of the part and its final position. In this way the robot could move to the starting position, grab the part and then deliver it to the final position. In this work, this application was developed on top of the application mentioned in the section 3.2.1. In fact, both the applications created previously can be used for this type of task, depending only on how the operator wants the robot to make the movements, for example, if the operator wants the robot to make the fastest path, it can work with the Handguiding_Points application and choose only 5 points (Figure 47, Figure 44, Figure 45, Figure 46 and Figure 48), to perform a simple and quickly pick and place task.

Now, if the operator wants the robot to follow exactly the path guided by the handguinding, in these circumstances, the Handguiding_Path application would be the most suitable option, because with it it would be possible to adjust the robot

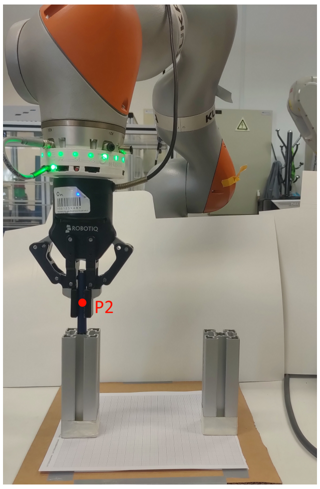


Figure 44: P2 position.

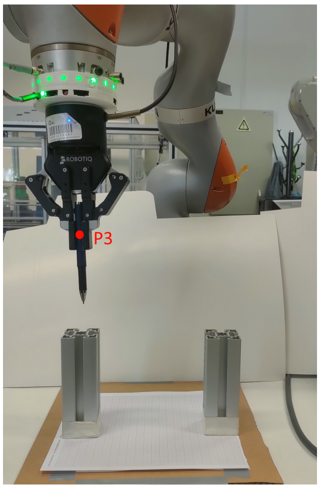


Figure 45: P3 position.

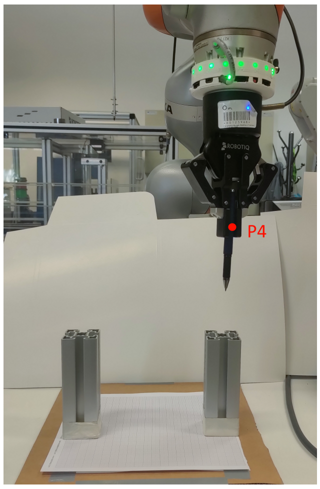


Figure 46: P4 position.

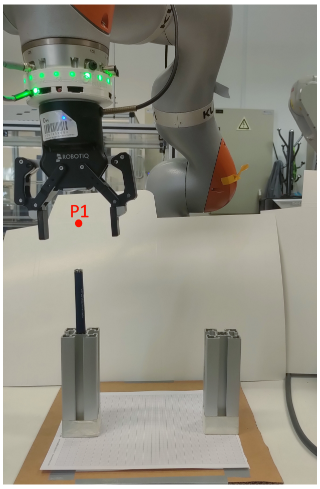


Figure 47: P1 position.

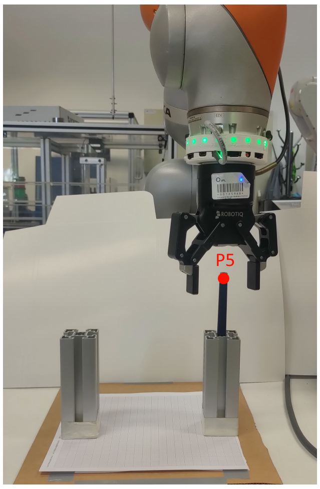


Figure 48: P5 position.

in such a way that it would be able to repeat exactly the movements made by the operator during the handguiding process.

Figure 49 shows what a path created with this application might look like for a short sampling time. It is possible to notice that even small operator oscillations were saved and replayed from P1 to Pn.

This pick and place application were quickly developed, showing how fast and easy can be the development of this type of application using the help of the developed Multi-purpose Flange Adapter System.

3.2.3 *Path Following with Force Feedback*

As already mentioned in chapter 2, the KUKA robot has force sensors spread across the 7 joints of the robot, this is a special feature of this robot that can be used for

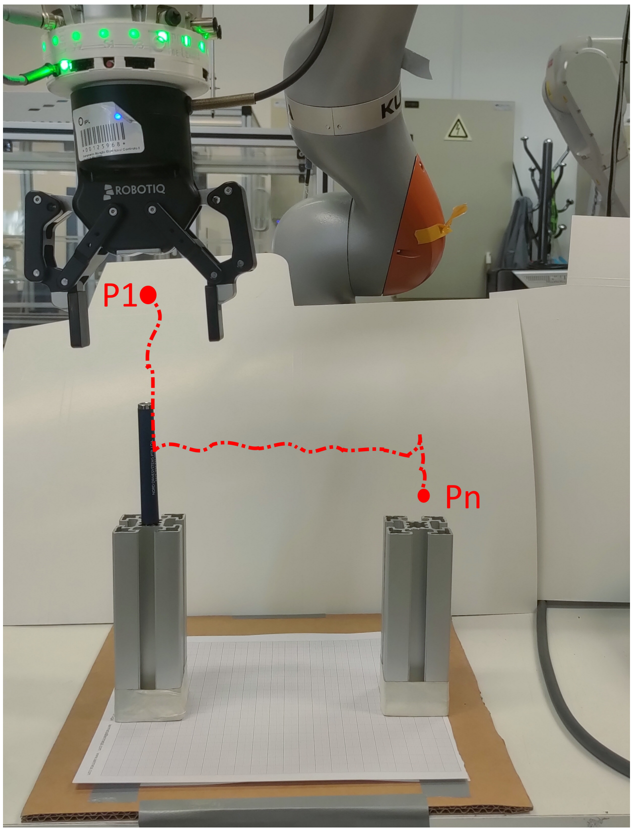


Figure 49: Pick and Place example.

the most diverse applications. A small example of an application that explores this feature of the robot will be presented in this chapter.

The developed application flowchart is shown in Figure 50. This application presents a motion control mode called impedance control. Under impedance control, the robot behavior is complacent. It is sensitive and can react to external influences such as obstacles or process forces. The application of external forces can cause the robot to go off the planned path, and this type of behavior can be exploited in some applications. The flowchart shows that the robot will make a trajectory to the base position and to the P0 point with a very strict impedance control, that is, for the robot to leave the planned path, large external forces must act on it, in this example, values of 5000 N were used for the X, Y and Z axes.

When the robot starts to follow the path from P1 to P2, the impedance control is changed in such a way that now for the robot to go out of the planned path in relation to the X and Y axes, it needs to feel an external force of 3000 N /m in the direction of these axes, but for the Z axis the value has been changed to 1N/m. This means that any small force in the Z direction will be enough to deflect the robot out of the way. With this configuration the robot is able to follow a path adapting to the plan.

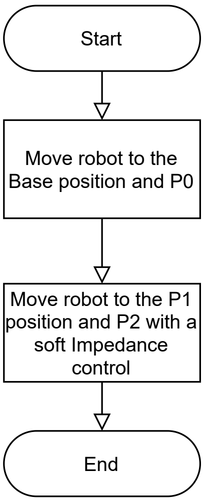


Figure 50: Path Following with Force Feedback application flowchart.

3.2.4 *Assembly operation with force sensing*

An assembly operation is a task where the operator needs to fit two or more parts together. However, when the fitting of the parts is very tight, and depends heavily on the orientation of the part, this task becomes more difficult, not just for a human being, but also for robots. In these situations, it is preferable to use a sensitive robot such as the KUKA robot, which is able to adjust its trajectory according to the forces felt in the workpiece, in order to fit it into its final position.

To better demonstrate this ability of the KUKA robot to sense external forces, the following application was developed, where the robot needs to fit a rod in a rod, normally used in combustion engines.

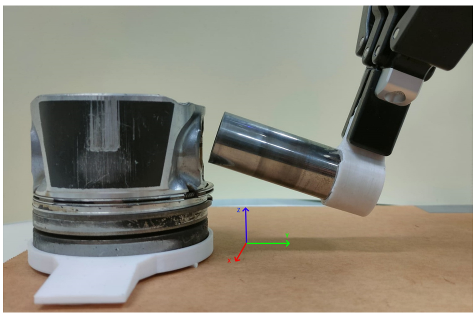


Figure 51: Assembly operation, view 1.

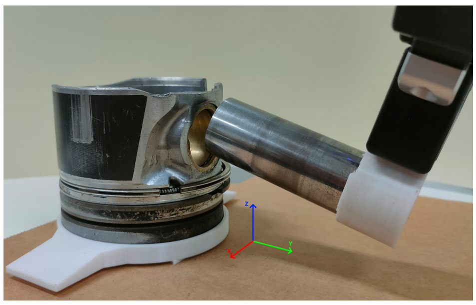


Figure 52: Assembly operation, view 2.

In this type of application, the fitting of the pieces is very tight, which makes it difficult to implement solely by means of precise positioning, as very small variations in the part position might yield the fitting unfeasible. Therefore, an approach that takes into account the robot's compliance was used, where the robot needs to be initially positioned close to the engine part, preferably with the rod in a slightly tilted position relative to the axis of the XY plane, as shown in Figure 51 and

Figure 52. With the robot in place, the application is executed according to the flowchart shown in Figure 53, explained in more detail next.

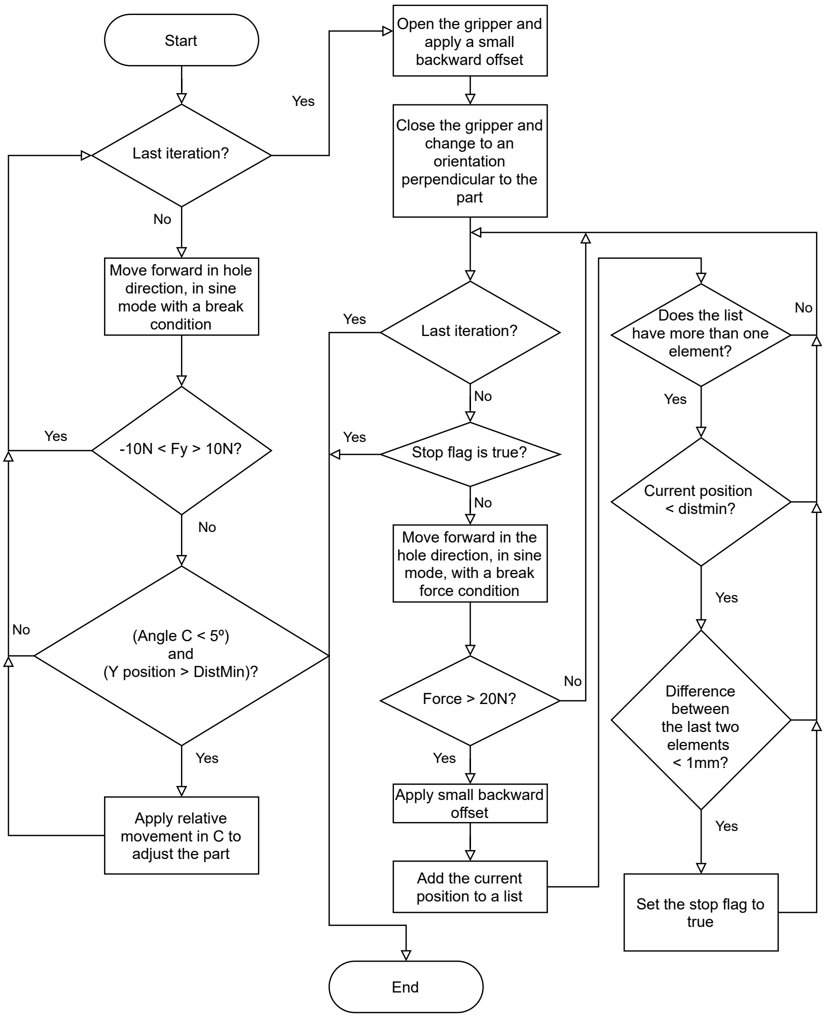


Figure 53: Flowchart of the assembly operation with force feedback application.

In the developed approach, the application performs several attempts which, in this case, a limit of ten attempts was chosen (in a real scenario, the operator would be called to intervene if the operation did not succeed in the specified number of attempts). When the application starts, it checks if it is the last iteration. If not, the KUKA robot moves the rod towards the piston with a sinusoidal impedance control, which causes the rod to vibrate as it moves, and with a force break condition. As soon as the rod hits the piston, the force exerted on the part increases and if this force in the axis Y is outside the chosen limit, the force break condition is reached and the robot knows that the part is hitting on something. The robot now checks if the position of the part is greater than the chosen minimum value, this value was defined in the program code itself, and in this case the chosen value was the -25 mm position in Y, in the robot's reference frame. This position indicates that the part is not yet fitted to the piston, and also checks if the current orientation C is less than 5°, if so, the robot starts rotating to fit the part to the piston, while moving in the snapping direction. This cycle is performed until the part is minimally engaged, or if it reaches the last iteration. After reaching the last iteration, the robot opens the gripper, releasing the rod, moves back and closes the gripper again. Once again it is executed another loop where the robot now pushes the rod into the piston, every time the total force felt on the rod is greater than 20N, the current position is stored and added to a list. After each iteration, the last two elements of the list are compared to check the difference between the last two positions of the robot. If this difference is less than 1mm, then it considers that the part is fitted, and finishes the application. If the difference is larger than 1mm, then the robot continues to push the piece until it reaches the maximum limit of ten iterations to then finish the application.

The first application was developed to test how the robot would fit into the part, this application is best suited for situations where the operator delivers the part to the robot and then positions the robot where it should fit the part. The second assembly application was developed, for situations where it is necessary to repeat the process as many times as necessary, such as on an assembly line, for example. In this way, the application allows the operator to teach not only the position of inserting the part, but also the position that the robot must go to get the rod. With this application, it is possible to store the pick up and insertion positions of the part in a text file, which is later used to load these positions and carry out the rod assembly process.

The flowchart in Figure 54 shows how this application works. After running the application, the operator needs to choose between three options: the first, is the rod position which, if chosen, the actual position of the robot is saved on a text file for this position; the second option is the insert position, which is the position where the robot must insert the rod on the piston which, if chosen, the current position of the robot is saved on a text file; the last option is the run option, which, if chosen, the robot will execute the motions needed to execute the assembly operation. The execution starts with the KUKA robot going to the rod position, then it picks the rod with the gripper, moves to the support direction to adjust the rod on the gripper, after sensing the end of the support, it goes backward to remove the rod from the support. With the rod on the gripper, the robot moves to the insert position, and starts to move forward with an impedance control with a force break condition, which is used to detect the collisions between the piston and the rod. After each collision, the robot makes a relative rotation in C to adjust the rod in the piston, but with a compliant movement towards the piston. This means that the robot reacts quickly to the assembly forces of the process and makes it easier to fit the rod into the piston. These motions are done until the orientation of the rod is higher than 1°between the plane XY. With the part approximately in a perpendicular position to the XZ plane where the piston is, the robot opens the

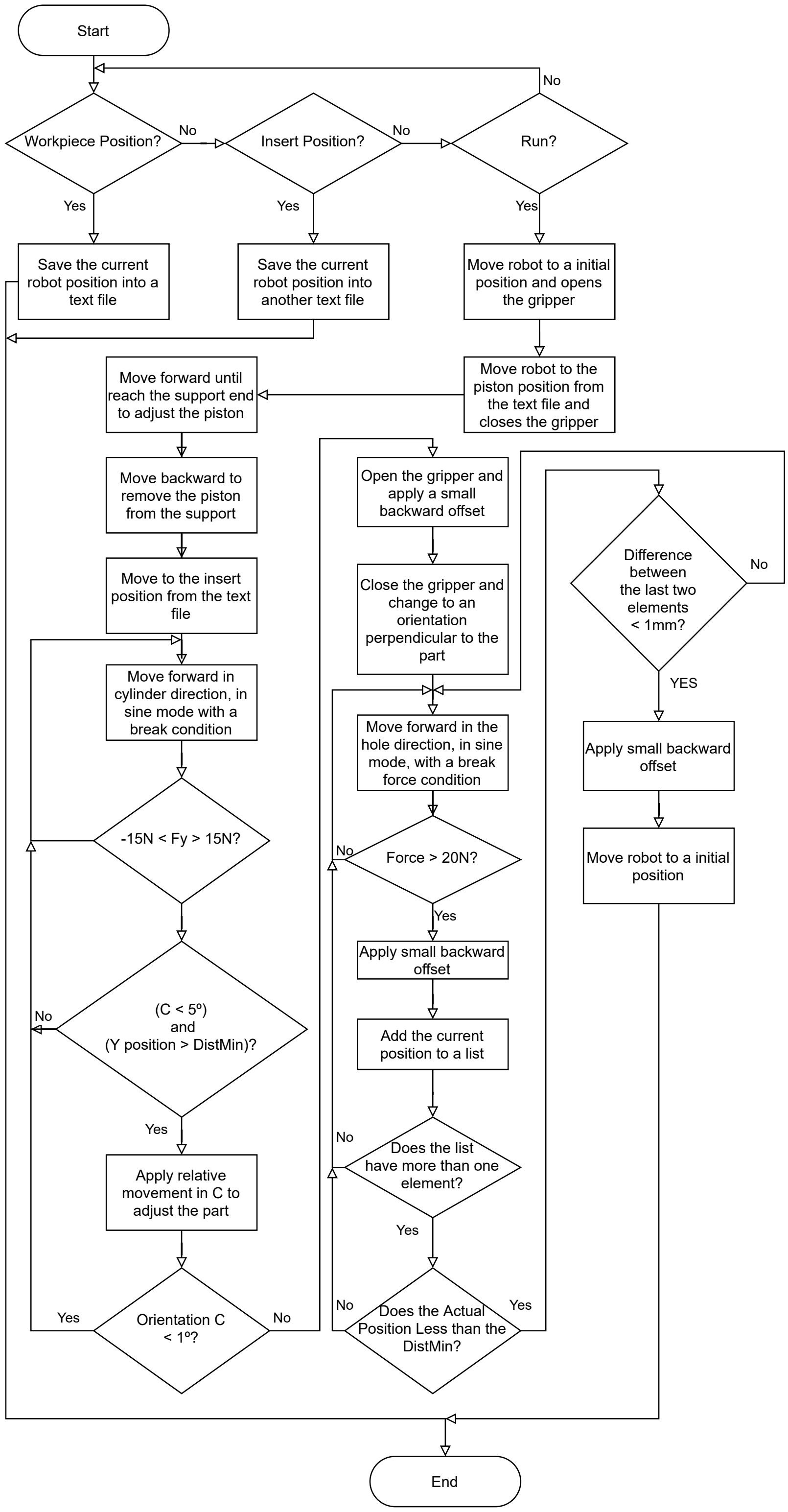


Figure 54: Flowchart of the second assembly operation with force feedback flowchart.

gripper and moves to the back of the rod, then it closes the gripper and start to push the rod into the piston. This motion is made using impedance control and a force break condition. This break condition is used in case of the robot is pushing the rod in a wrong position so if the break condition is triggered, the robot moves back and tries to push the rod again. Note that because of the parameters used in the impedance control, when the robot performs a new push, the gripper is in a slightly different position.

4

TESTS AND RESULTS

In this chapter, through the application Path Following with Force Feedback of Section 3.2.3, data from the robot's force sensors were obtained. Different tests and analyzes were performed for each test set.

The following experiment was carried out, the robot holds a pen using the gripper and moves from the initial position to point P0, this point is slightly above a notebook, now there is point P1 which is just below and point P2 which is 202 mm ahead on the Y axis. After reaching point P0, the force and torque data on the robot's joints are collected as it travels from P0 to P1 and then P2, as shown in Figure 55. The method getExternalForceTorque() (see for more information in [15]) was used to read the external Cartesian forces and torques currently acting on the robot flange, the TCP of a tool or any point of a gripped workpiece. With it is possible to get a vector with the Cartesian forces which act in the X, Y and Z directions (unit: N) and a vector with the Cartesian torques which act about the X, Y and Z axes (unit: N/m). The method getExternalForceTorque() was used to collected and save data with a frequency of 20Hz. The points used were, X, Y and Z are offset of the origin along the axes of the parent frame, and A is the rotation offset about the Z axis, B is the rotation offset about the Y axis and C is the rotation offset about the X axis. The X, Y, and Z axes shown in the following images only indicate the orientation of the plane, not the current position of the origin in the images.

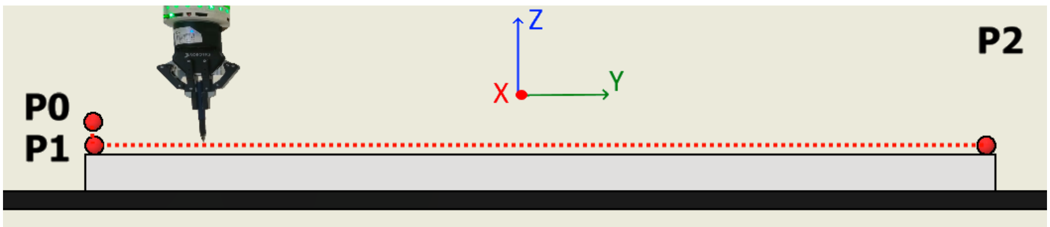


Figure 55: Test 1 with the Path Following with Force Feedback application.

	X (mm)	Y (mm)	Z (mm)	A (°)	B (°)	C (°)
P0	577.00	-180.00	280.00	-177.62	0.00	-177.62
P1	577.00	-180.00	270.00	-177.62	0.00	-177.62
P2	577.00	22.00	270.00	-177.62	0.00	-177.62

Table 1: Frame points used on test.

	X (mm)	Y (mm)	Z (mm)	A (°)	B (°)	C (°)
1º	578.28	21.58	272.59	-177.54	-0.16	-177.72
2º	578.30	21.60	272.58	-177.54	-0.16	-177.72
3º	578.29	21.55	272.60	-177.63	-0.16	-177.73

Table 2: Robot positions when the motion stops.

Five different tests were performed with the application, and each test was performed three times. The first test was done with the notebook plane parallel to the XY plane as shown in Figure 55.

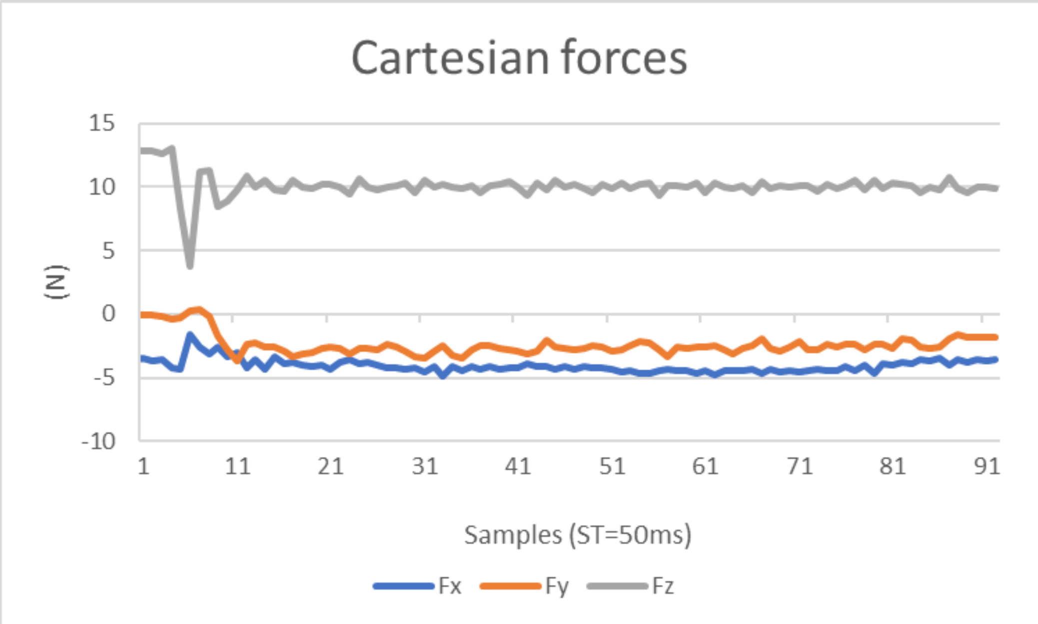


Figure 56: Forces measured during the first test.

Analyzing chart 56, it is possible to notice a variation at the beginning that represents the transition from point P0 outside the notebook to dyeing the notebook at P1, in this interval is when the light impedance mode is used, at this time the pen has a small drop in P0 to P1 where it reached the notebook, it is possible to notice that the force measured on the Z axis is around 10N, this is due to the fact that in the chosen impedance control, an external force of 10N on the Z axis was added to compensate for the weight of the robot's flange so that during the transition from P0 to P1, the robot would descend slowly until touching the notebook, this drop between the two points is very sudden and causes a collision detection stopping the movement. After the 3 repetitions for this test the robot finished its movement in the following positions on Table 2:

The next test were performed for the notebook positioned at an inclination of 10º Figure 57 and 30º Figure 58 in relation to the XY plane, keeping the same points as the first test.

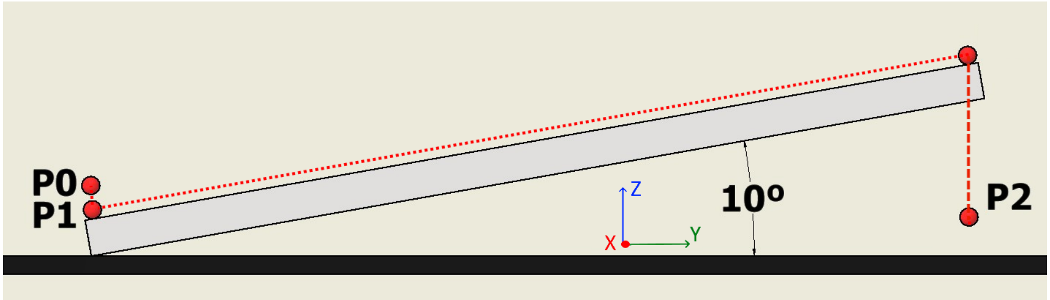


Figure 57: Test 2 with the Path Following with Force Feedback application.

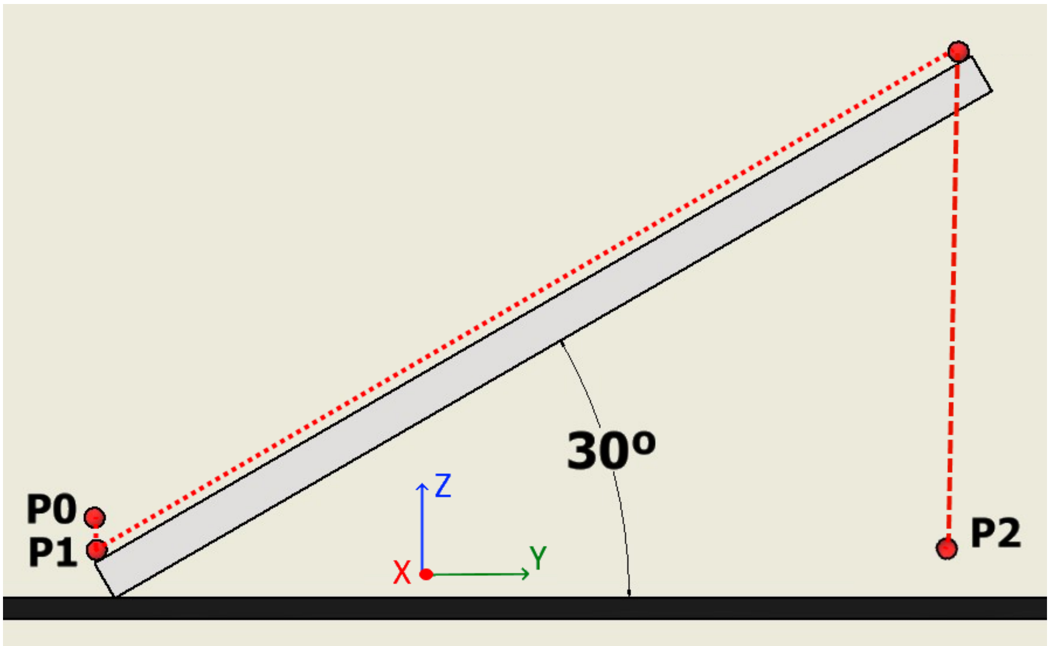


Figure 58: Test 3 with the Path Following with Force Feedback application.

Analyzing the graphs obtained in the tests, shown in Figure 59 and Figure 60, a pattern can be found. The forces on the Z axis have a small variation when finding the notebook, but, soon the variation becomes small. This can be explained, because after finding the notebook, the KUKA robot always keeps the pen in the notebook, exerting the same force. Analyzing the results for the X axis, the value oscillates little, how it do not have large forces acting in this direction, but when the results for the Y axis are verified, the strength of the force increases as the notebook's tilt increased, that's because when the notebook is tilted, the robot senses a resultant diagonal force composed of a Z and a Y component.

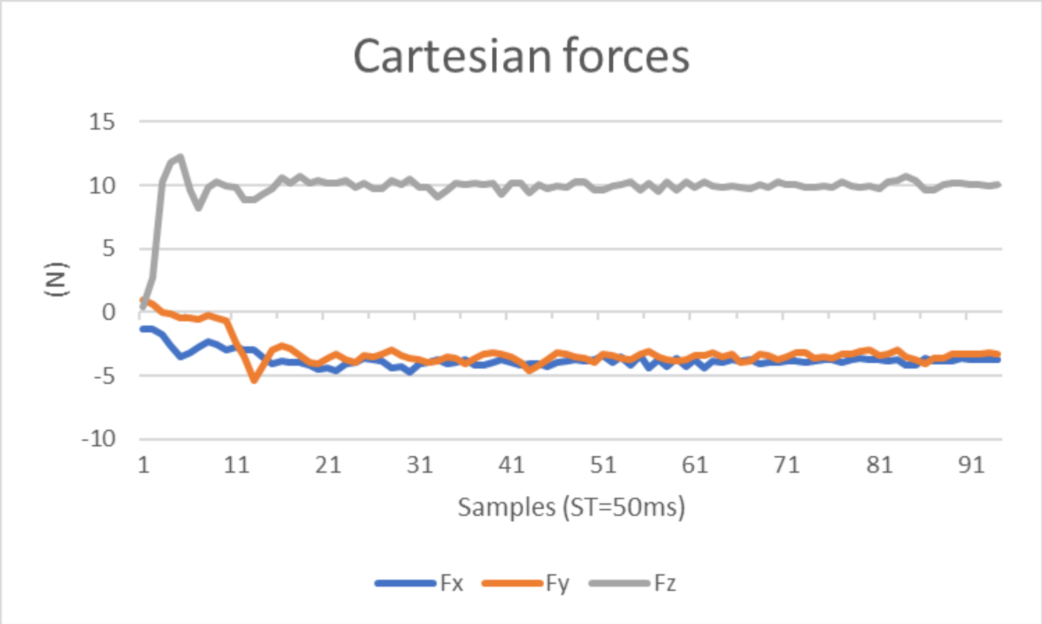


Figure 59: Forces measured during the second test.

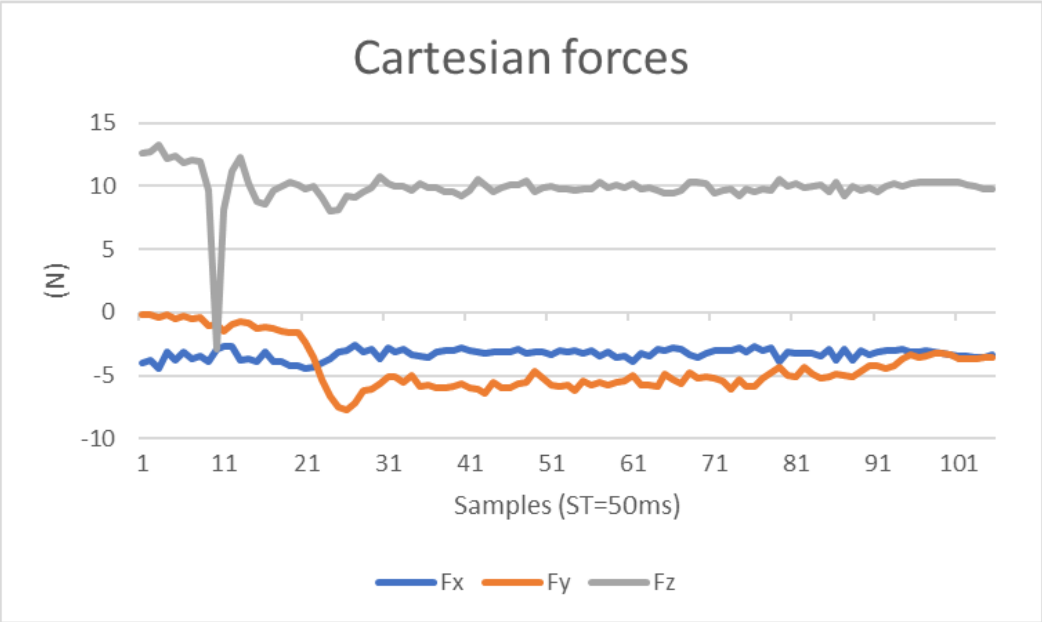


Figure 60: Forces measured during the third test.

The results obtained provide an idea of how force and torque sensors behave during a robot motion, and how this feature can be used in application development.These were simple tests that show how it is possible for the KUKA robot to make movements, sensing and adapting to the obstacles of the environment in which it is inserted. Furthermore, theses tests successfully shown that the developed Multi-purpose Flange Adapter System allowed for the handguide application run as expected.

Data were also collected in relation to the application of Section 3.2.4, in this application the Cartesian forces and Cartesian torques in relation to the X, Y and Z axes were verified during a stretch of the path executed by the KUKA robot. In this application the robot leaves its home position, and moves to the position to pick up the rod, this position was stored by positioning the robot through the handguiding mode. Once in position, the getExternalForceTorque() method is used to collect and save data with a frequency of 5Hz. Now the robot picks up the rod, moves it back far enough to remove the rod from the holder, and then moves to the rod insertion position, which was also stored by positioning the robot via handguiding mode, as shown in Figure 61.

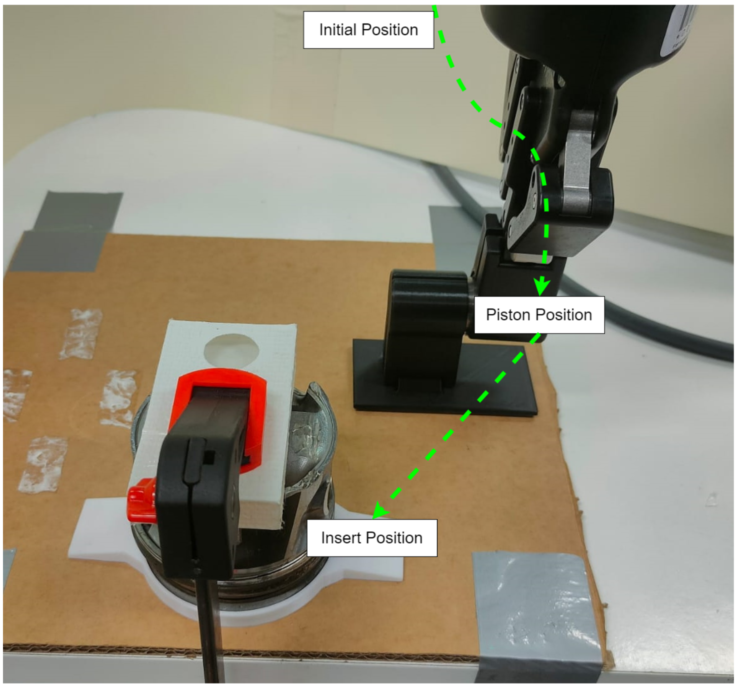


Figure 61: Test of the assembly operation with force feedback application.

Analyzing the graphs obtained in the tests, shown in Figure 62, it is possible to highlight three steps during the process, and see how the application works.

The first part of the graph shows a force variation in X of -15N, this is when the robot moves towards the support to adjust the rod in the gripper. Afterwards the robot removes the rod and moves to the position of inserting the rod, these movements are performed in air, so, at this stage the forces do not vary much as

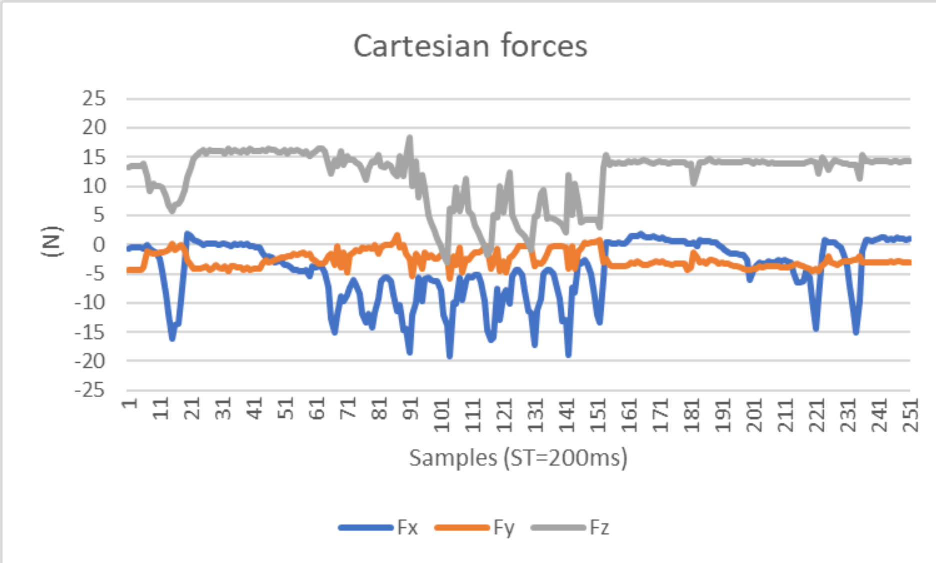


Figure 62: Forces measured during the test.

shown between samples 21 and 61 of the graph. In the central part of the graph it is possible to notice force peaks with larger variations in X and Z, the forces vary between -15N and -20N in X and between 15N and 20N in Z, these values are in accordance with the flowchart of the Figure 54, indicating that at that moment the robot was making the fitting of the rod in the piston, at each peak of force less than -15N in Y, the robot stops and adjusts the rod's orientation in C, this explains the fact that the forces in Z react together with the forces in X. Between samples 121 and 161 it is possible to notice that the forces remain more constant, at this moment the rod is already partially inside the piston, now the robot starts to push the rod until it fits. The fit is detected in the final part of the graph, where it has two forces peaks in X, the first peak is when the robot senses the first resistance of the rod to movement, then it stops, backs up and pushes again, thus detecting the second peak, the robot compares its position on the Y axis to the two peaks of force and checks the distance, if the variation is less than 1mm, the robot considers the rod fitted.

Below is the link to access an online video playlist, where you can find the videos of each application developed, and the link to the online repository, where you can find the Arduino programs, the robot application source-code, the 3D design project files, as well as the and the PCB layouts and schematics of the developed boards.

- Playlist with all videos: https://www.youtube.com/playlist?list=PLXXiQrfIoC9R-gz5CtGT_aw7T15yv6N4l
- Online repository: https://github.com/ipleiria-robotics/iiwa

5

CONCLUSIONS AND FUTURE WORK

The KUKA LBR iiwa robot is a very versatile robot, but the available flange did not enable handguide operations easily, and was very limited in terms of electrical connectivity and communication. The goal was set to develop an adapter system to be mounted on the flange, the proposed system should allow greater functionality and ease of use. The developed Multi-purpose Flange Adapter System added the Handguiding function to the robot, which is very useful and greatly facilitates the development of applications with the robot, as shown in this work. The KUKA robot is a collaborative and sensitive robot, allowing to do jobs that were not possible in the past. As was shown by the "Path Following with Force Feedback" and the "Assembly Operation with Force Sensing" applications, that non-sensitive robots would not be able to perform.

With the analysis of the obtained results on chapter 4, the precision of the application was proven, when comparing the position values obtained after the end of each motion with the end point goal value (P2). It is possible to verify an error of less than 0.2% for the X axis position, 2% for the Y axis position, 0.9% for the Z axis position and, for the rotational offsets A, B and C, the error was around 0.05%. These values show that the robot did not go far from its goal, taking into account that, in these tests, a lower stiffness impedance control was used, making the robot more compliant to external forces, so these values could still be improved depending on the stiffness values chosen for the robot motion. The data obtained from the application shown in Section 3.2.4, showed the behavior of the robot during the process of assembling the part, coinciding with what was described in the application and therefore proving its operation. All developed applications made use of the developed Multi-purpose Flange Adapter System, proving that the system works correctly and helps in the development of applications of different models.

The data obtained showed the operation of the robot's force and torque sensors, which can be used for the most diverse applications. As future work, the main points to be addressed are:

- Graphic applications more complete: it would be interesting to be able to use an application to analyze the data obtained by the robot, for example, an application that allows the operator to see the values of force and torque of the robot in a graph that is generated in real time, this application could be generated in robot's joystick or even by another external system that communicates in some way with the robot to acquire the data and display the graphics.

- Develop a java library that encompasses the entire process of assembling a part, so that new users can use certain methods of this library to facilitate the development of assembly applications.

BIBLIOGRAPHY

[1] *Arduino*. https://www.arduino.cc/en/guide/introduction. Accessed: 08-26-2021. 2021.

[2] *Arduino Micro*. https://store.arduino.cc/products/arduino-micro. Accessed: 08-26-2021. 2021.

[3] *Arduino Uno*. https://store.arduino.cc/products/arduino-uno-rev3/. Accessed: 08-26-2021. 2021.

[4] ATMEL Corporation. "8-bit AVR Microcontroller Atmega328p data sheet". In: *Power* (2009), pp. 1–23.

[5] *Bluetooth 4.0 BLE module*. HM-10. V507. JNHuaMao Technology Company. 2013.

[6] *Bluetooth 4.0: Low Energy*. https://californiaconsultants.org/wp-content/uploads/2014/05/CNSV-1205-Decuir.pdf. Accessed: 08-26-2021. 2021.

[7] *Bluetooth® Dual Mode Module*. RN4678. DS50002519F. Microships. 2016.

[8] *Bootloader Development*. https://www.arduino.cc/en/Hacking/Bootloader. Accessed: 2021-06-29. 2021.

[9] *Collaborative Robot CR-7iA and CR-7iA/L*. https://www.fanuc.eu/it/en/robots/robot-filter-page/collaborative-robots/collaborative-cr7ial.. Accessed: 2021-08-28. 2021.

[10] *Collaborative Robot Technology and Applications*. https://www.robotics.org/userAssets/riaUploads/file/4-KUKA_Beaupre.pdf. Accessed: 2019-06-29. 2019.

[11] Ethercat Bus Coupler. "EK110x-00xx, EK15xx". In: (2020).

[12] *Dual-arm SCARA Robot "duAro"*. https://robotics.kawasaki.com/userAssets1/productPDF/Europe-Africa/kawasaki_Dual-Arm_Scara_Robot.pdf. Accessed: 08-26-2021. 2021.

[13] *e-Series FROM UNIVERSAL ROBOTS*. https://www.universal-robots.com/media/1802432/e-series-brochure.pdf. Accessed: 08-26-2021. 2021.

[14] Naveen Erasala and David C Yen. "Bluetooth technology: a strategic analysis of its role in global 3G wireless communication era". In: *Computer Standards & Interfaces* 24.3 (2002), pp. 193–206. ISSN: 0920-5489. DOI: https://doi.org/10.1016/S0920-5489(02)00018-1. URL: https://www.sciencedirect.com/science/article/pii/S0920548902000181.

[15] Programming Instructions. "KUKA Sunrise.OS 1.14 KUKA Sunrise.Workbench 1.14". In: (2017).

[16] Texas Instruments. "LM317 3-Terminal Adjustable Regulator". In: (2020).

[17] *ISO 10218-1:2011 Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots*. https://www.iso.org/standard/51330.html. Accessed: 2019-06-28. 2019.

[18] *ISO 10218-2:2011 Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration*. https://www.iso.org/standard/41571.html. Accessed: 2019-06-29. 2019.

[19] *ISO/TS 15066:2016 Robots and robotic devices – Collaborative robots*. https://www.iso.org/standard/62996.html. Accessed: 2019-06-29. 2019.

[20] *KINOVA® Gen3 Ultra lightweight robot*. https://www.kinovarobotics.com/sites/default/files/UG-014_KINOVA_Gen3_Ultra_lightweight_robot_User_guide_EN_R06_0.pdf. Accessed: 08-26-2021. 2021.

[21] *KUKA - Sensitive robotics LBR iiwa*. https://url.gratis/tTGJ52. Accessed: 2019-06-26. 2019.

[22] *Lexico powered by Oxford*. https://www.lexico.com/en/definition/sensitive. Accessed: 2019-06-26. 2019.

[23] YuKang Liu and YuMing Zhang. "Toward Welding Robot With Human Knowledge: A Remotely-Controlled Approach". In: *IEEE Transactions on Automation Science and Engineering* 12.2 (2015), pp. 769–774. DOI: 10.1109/TASE.2014.2359006.

[24] Douglas Morrison, Juxi Leitner, and Peter Corke. "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach". In: June 2018. DOI: 10.15607/RSS.2018.XIV.021.

[25] Dario Panariello et al. "Using the KUKA LBR iiwa Robot as Haptic Device for Virtual Reality Training of Hip Replacement Surgery". In: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. 2019, pp. 449–450. DOI: 10.1109/IRC.2019.00094.

[26] *Panda*. https://www.franka.de/. Accessed: 08-26-2021. 2021.

[27] Gaoyang Pang, Geng Yang, and Zhibo Pang. "Review of Robot Skin: A Potential Enabler for Safe Collaboration, Immersive Teleoperation, and Affective Interaction of Future Collaborative Robots". In: *IEEE Transactions on Medical Robotics and Bionics* 3.3 (2021), pp. 681–700. DOI: 10.1109/TMRB.2021.3097252.

[28] C. Pintea. "Advances in Bio-inspired Computing for Combinatorial Optimization Problems". In: *Intelligent Systems Reference Library*. 2014.

[29] "RN4678 Bluetooth ® Dual Mode Module Command Reference User ' s Guide". In: (2016).

[30] Robotiq Inq. "Robotiq 2F-85 & 2F-140 for Cobots". In: (2021), pp. 1–99.

[31] Kongkiat Rothomphiwat et al. "Advanced Collaborative Robots for the Factory of the Future". In: *2021 IEEE/SICE International Symposium on System Integration (SII)*. 2021, pp. 578–579. DOI: 10.1109/IEEECONF49454.2021.9382664.

[32] ON Semiconductor. "1 . 0 A Low-Dropout Positive Fixed and Adjustable Voltage Regulators". In: (2008), pp. 1–17.

[33] *Serial*. https://www.arduino.cc/reference/en/language/functions/communication/serial/. Accessed: 08-28-2021. 2021.

[34] Rui Song et al. "Skill learning for robotic assembly based on visual perspectives and force sensing". In: *Robotics and Autonomous Systems* 135 (Jan. 2021), p. 103651. ISSN: 0921-8890. DOI: 10.1016/J.ROBOT.2020.103651.

[35] Meseret Abayebas Tadese et al. "Passivity Guaranteed Dynamic Friction Model With Temperature and Load Correction: Modeling and Compensation for Collaborative Industrial Robot". In: *IEEE Access* 9 (2021), pp. 71210–71221. DOI: 10.1109/ACCESS.2021.3076308.

[36] Digital Input Terminals. "EL10xx, EL11xx". In: (2021).

[37] Digital Output Terminals. "EL20xx, EL2124". In: (2021).

[38] *TM12*. https://www.hmkrobotics.com/images/1_products/actuators___robotcs/cobots/1tm12/tm_robot_smart_simple_safe.pdf. Accessed: 08-26-2021. 2021.

[39] Valeria Villani et al. "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications". In: *Mechatronics* 55 (Nov. 2018), pp. 248–266. ISSN: 0957-4158. DOI: 10.1016/J.MECHATRONICS.2018.02.009.

[40] *YuMi - IRB 14000 - Overview*. https://search.abb.com/library/Download.aspx?DocumentID=9AKK106354A3256&LanguageCode=en&DocumentPartId=&Action=Launch. Accessed: 2019-06-26. 2019.

APPENDIX

A

BLUETOOTH DEVICES FOR WIRELESS COMMUNICATION

This appendix describes some different Bluetooth devices that were evaluated during this project. The last section of this appendix, shows the device that was selected for use in this work.

When it comes to short-range wireless communication between devices, Bluetooth is soon remembered. Designed so that devices could communicate wirelessly, Bluetooth uses UHF radio waves in the 2.402GHz to 2.48GHz bands.[14] Communications between Bluetooth devices are typically peer-to-peer. However, when making a connection between two or more devices, one device acts as master and the others are slaves during the connection, these connection types are called piconet and support up to 8 Bluetooth devices. Bluetooth technology has basically two categories, Bluetooth Classic which is efficient for devices that know when they need to be turned on and off and Bluetooth Low Energy which can be a device (server) that will advertise for attention only if you need it or a host (customer) who will hear often enough to hear the ads.[6]

A.1 RN4678

Figure 63: Dual Module Bluetooth - RN4678

The RN4678 (Fig 63) is a Bluetooth Dual Mode module, which enables the designer to easily add classic Bluetooth and Bluetooth Low Energy capability to devices. The RN4678 bridges the devices to Smart Phones and Tablets for convenient data transfer, control and access to cloud applications [7].

A.2 HM-10

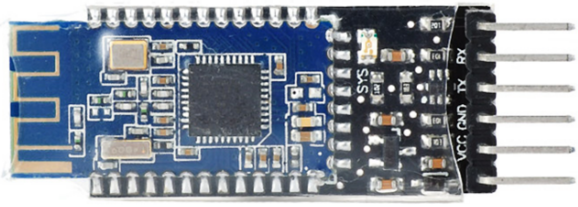


Figure 64: Bluetooth Low Energy - HM-10

The HM-10 (Fig. 64)is a Bluetooth 4.0 module compatible with Android and iOS phones. In addition, it has a low energy consumption as it is a Bluetooth Low Energy (BLE) device. This module has a built-in signal converter, which can be used in 5 or 3.3V circuits without the need for additional hardware, it supports operating as a MASTER or as a SLAVE, just configuring it by AT commands. It is possible to communicate between two bluetooth modules (one MASTER and the other SLAVE) and also communicate between one module and a notebook, cell phone, among others [5].

A.2.0.1 *RN4678 Signal Tests*

The RN4678 module supports a variety of commands for controlling and configuration, that are described on the datasheet [29]. A very basic command can be sent to check the basic settings, the character "D" was sent through the arduino serial monitor and after the module respond, the Mac Address and Bluetooth Name of the module were read.

After verifying that the operation of the RN4678 was OK, the command "F" was sent to search for nearby bluetooth devices, thus, the mac address of the second module was found and then the command "C, "addr"" was sent to the module connect to the chosen mac.

The connection to the device was made, however it was noticed that the signal was lost sometimes. So it was decided to do some bluetooth range tests, since the input module is positioned on the robot's flange and the output module should be close to the robot controller, so measuring the distance from the flange to the base of the robot is 800mm, plus the distance from the robot to the controller which is over 600mm, so the connection between the two modules must be greater than 1400mm, with one more safety margin it is safe to say that the modules must connect at least 2m apart.

To check if the RN4678 module connection complied with the said specifications, the following tests were performed to verify the influence of objects between the modules, as it is very likely that the modules were separated by a table, and tests were also carried out for different modules of the antenna of both modules, as one robot flange can move in different ways in the robot workspace, this can change the position of the RN4678 in relation to the other module.

The first test was done with the antenna of the input module in a fixed direction, horizontal, varying the antenna of the output module to the same direction, but in the same sense or placing it in the opposite sense, and with line of sight (LOS) or with non line of sight(NLOS), with this it was possible to assemble the following table with the results.

Table 3: First test made for one RN4678 in horizontal direction.

Antenna Position	Obstacles	Max Connection Distance (mm)	Max Distance after Connected (mm)
Same Sense	LOS	670	970
Same Sense	NLOS	600	970
Opposite Sense	LOS	300	820
Opposite Sense	NLOS	240	760

The second test was done with the antenna of the input module in a fixed direction, vertical, varying the antenna of the output module to the same direction, but in the same sense or placing it in the opposite sense, and with or without obstacles, with this it was possible to assemble the following table with the results.

Table 4: First test made for one RN4678 in vertical direction.

Antenna Position	Obstacles	Max Connection Distance (mm)	Max Distance after Connected (mm)
Same Sense	LOS	150	770
Same Sense	NLOS	100	770
Opposite Sense	LOS	650	900
Opposite Sense	NLOS	550	800

In the previous tables, column "Max Connection Distance" indicates the maximum distance that one module was from another to be able to establish the connection and column "Max Distance afetr Connected" indicates what was the maximum value that was reached after a connection was made. Analyzing the data, it is possible to notice that the greatest distances that were reached were when the bluetooth modules had their antennas pointing in the same sense, but to find out if the module has a good connection, the worst case was analyzed and for it the greatest distance reached was 100 mm with obstacles to be able to establish a connection between the modules, so the module as it is does not meet the project specifications, which would be a distance of at least 2000mm.

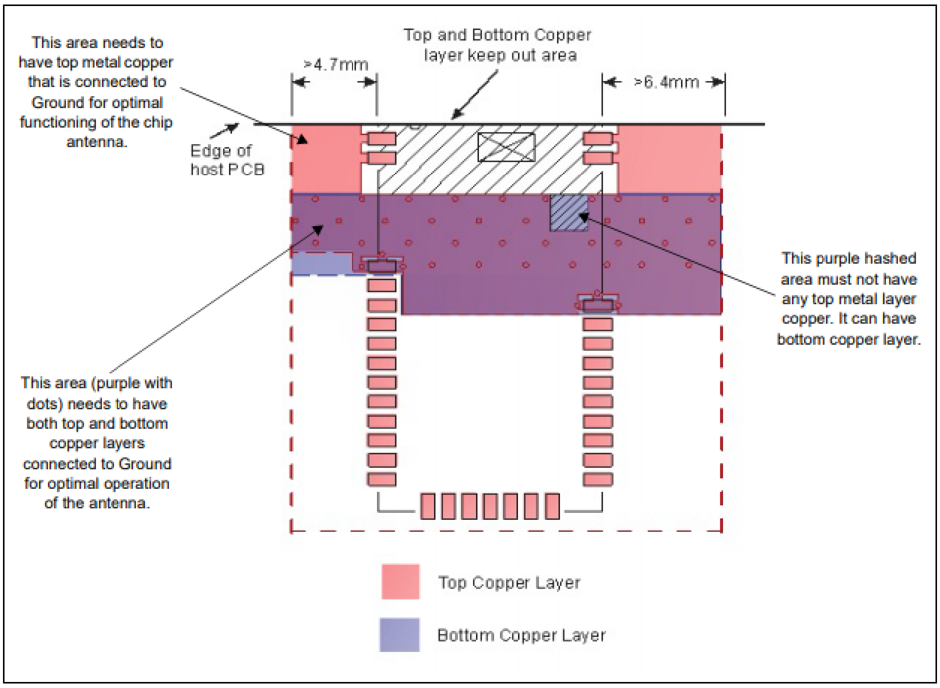


Figure 65: Mounting Suggestion for RN4678.

The RN4678 device did not present good results, not being enough to keep the module as an option for bluetooth communication of the hand-guided device.

A.2.0.2 *HM-10 Signal Tests*

In this section, communication tests were carried out between two HM-10 bluetooth modules and both modules connected and received information at high distances and with obstacles, as in the RN4678, but the connection between both HM-10 modules was made and it was possible to send and receive information over great distances, which was not possible with the RN4678, so the HM-10 is enough to keep the module as an option for bluetooth communication of the hand guided device.

To make the connection it is necessary to leave one module as master using the command "AT+ROLE1" and the other as slave wich can be set by send the command "AT+ROLE0", only then is it possible to make the master connect to the slave.

B

BURNING THE BOOTLOADER ON THE ATMEGA32U4

This appendix describes the circuit diagram used in case the acquired ATmega microcontroler IC that does not include the Arduino bootloader since, without it, it is not possible to connect to the ATmega and program it using the Arduino IDE.

When performing the initial tests with the developed ATmega32U4-based microcontroller system, a problem was found, when connecting the board through the USB port to the PC to program the ATmega, the board was not listed in any COM port on the computer, that is, without it, it was impossible to upload the program and proceed with the project.

Microcontrollers are usually programmed through a programmer unless you have a bootloader in your microcontroller that allows installing new firmware without the need of an external programmer.

As the goal was to develop an easy-to-use board, the bootloader was required and had to be written to the ATmega32U4. There are several ways to do this, one of which is to burn the bootloader using another arduino, as described in [8]. To facilitate the connection and avoid wiring problems between the ATmega32U4 and the Arduino UNO, a small board was developed with the necessary connections to burn the bootloader.

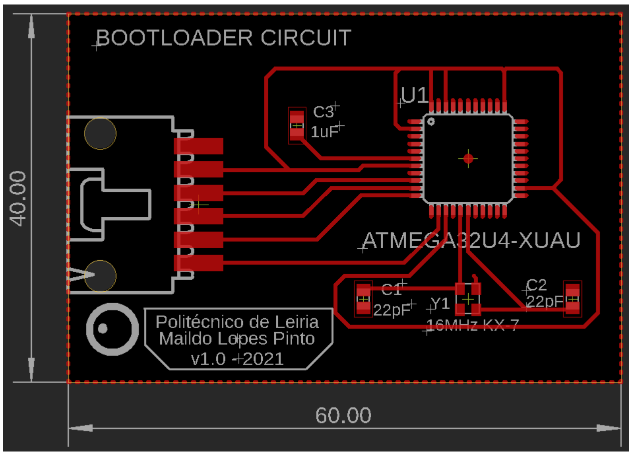


Figure 66: Bootloader PCB developed.

Figure 66 shows that the developed PCB has 40mm x 60mm, it has an 70634-POS6 molex connector to facilitat connections with the board, speeding up the process of writing the bootloader. This is a process that needs to be done only once per microcontroller, and only if the purchased ATmega does not already have the bootloader.

To burn the bootloader proceed as follows:

1. Download the ArduinoISP code for the Arduino UNO.
2. Connect the connector to the Arduino Uno board and the wires according to the ArduinoISP code, as shown in the following table.

Bootloader Circuit Connector Pin	Arduino Uno Pin Number
1	10 - Reset
2	12 - MISO
3	11 - MOSI
4	13 - SCK
5	VCC
6	GND

3. In the arduino IDE, select in "Tools", "Board", select the board "Arduino Leonardo" (for the ATmega32U4 case), in "Programmer" choose "Arduino as ISP" and finally click in "Burn Botloader".

The process starts and, after a few seconds the bootloader is written to the microcontroller memory.

C

I/O CONFIGURATION WITH WORKVISUAL

The Workvisual is used to make the I/O configuration, which contains the inputs and outputs of the used field buses. These inputs and outputs, can be used to configure some devices like the Robotiq 2F-85 gripper, and the Beckhoff modules to be used in the Sunrise applications.

To correctly configure these devices, follow the following sequence of steps:

- First, it is necessary install the workvisual addon for it works with KUKA sunrise, so, on WorkVisual, select the menu sequence "Extras" > "Option package management...". A new window is opened, select the option "Add a new option package from a file" and select the Sunrise file with extension "kop", and install it.
- In Section 11.3 from the document [15], is shown how to create a new IO configuration.
- On WorkVisual, add the robotiq gripper's EtherCat description file "Robotiq_AG_ECS_20181109.xml" or the Beckhoff modules description files. For this, in WorkVisual select the menu sequence "File" > "Import/Export" > "Import device description file" > "Browse", select the file type to "EtherCAT ESI" and look for the folder where the ".xml" of the devices are located.
- Add the gripper's controller "NIC 50-RE/ECS", or the specific Beckhoff modules to the KUKA Extension BUS (SYS-44) as shown in Figure 67
- Follow the steps on KUKA's Manual Chapter 11 "Bus configuration" [15], for the configuration and I/O mapping in WorkVisual. The configuration should be made as shwon in Figure 68.
- Export the configuration file to Sunrise Workbench, in WorkVisual select the menu sequence "File" > "Import/Export" > "Export I/O configuration to Sunrise Workbench project" and restart the robot.

For more information see, KUKA's Manual Chapter 11 "Bus configuration" [15].

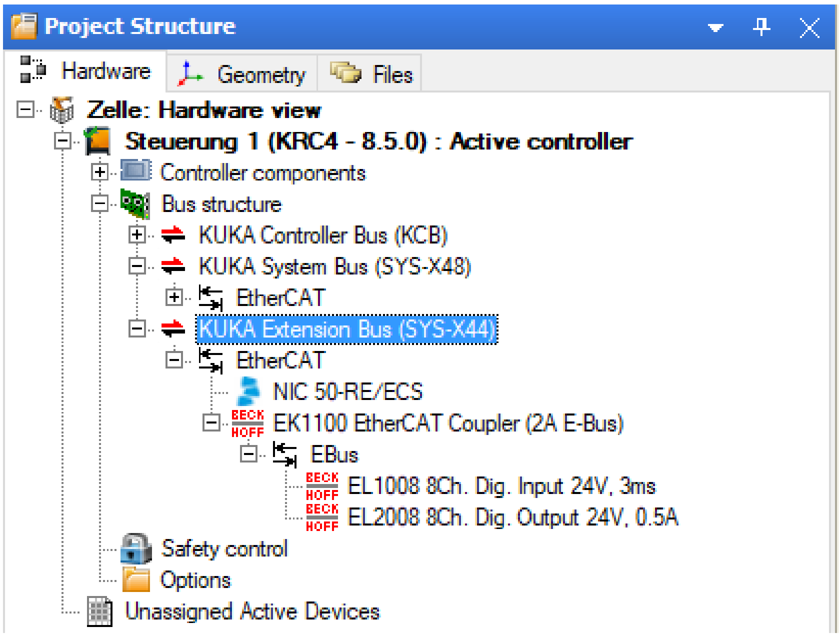


Figure 67: Devices on SYS-X44.

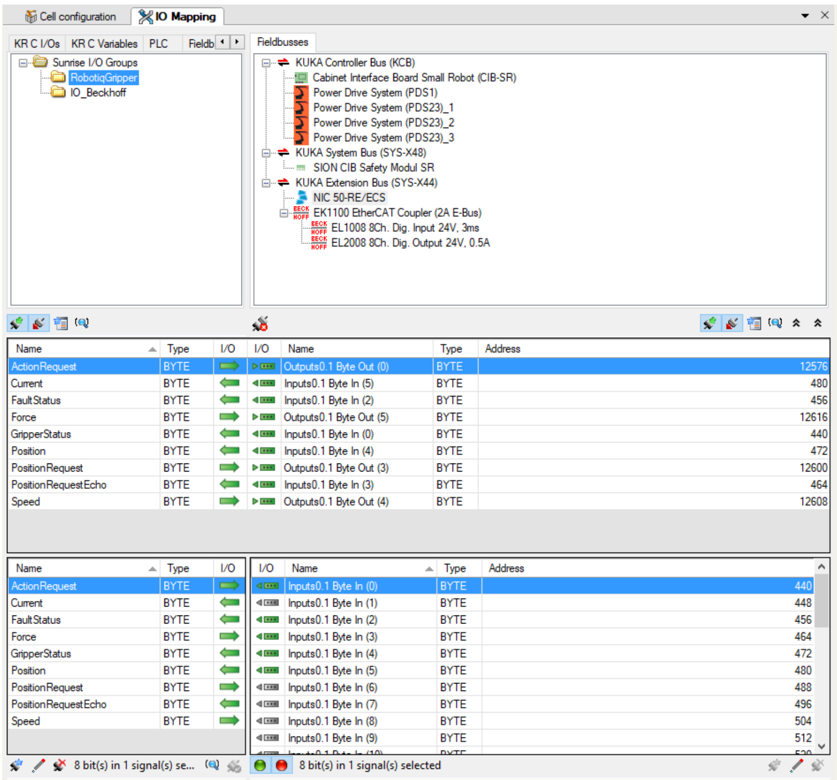


Figure 68: I/O mapping.

D

ELECTRICAL CABINET

This appendix show the electrical panel containing the other components that are not attached to the flange, which includes the control module described in Section 3.1.5, the power supply, the voltage regulator, the Beckhoff EK1100, EL1008 and EL2008 modules, the gripper controller and the necessary cables. In the electrical project, a relay module controlled by the controller module was added to delay the Beckhoff module energization, necessary to avoid communication problems that occurred if the Beckhoff module and the gripper controller are turned on at the same time. The electrical design of the electrical cabinet was developed using the software EPLAN Eletric P8, version 2.7.

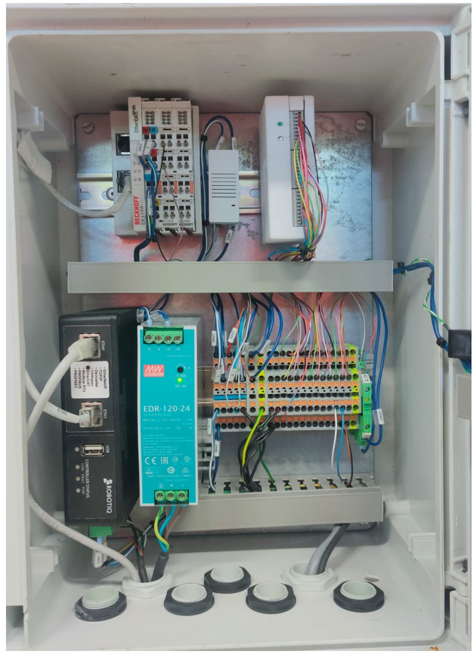
POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Polytechnic of Leiria - School of Technology and Management

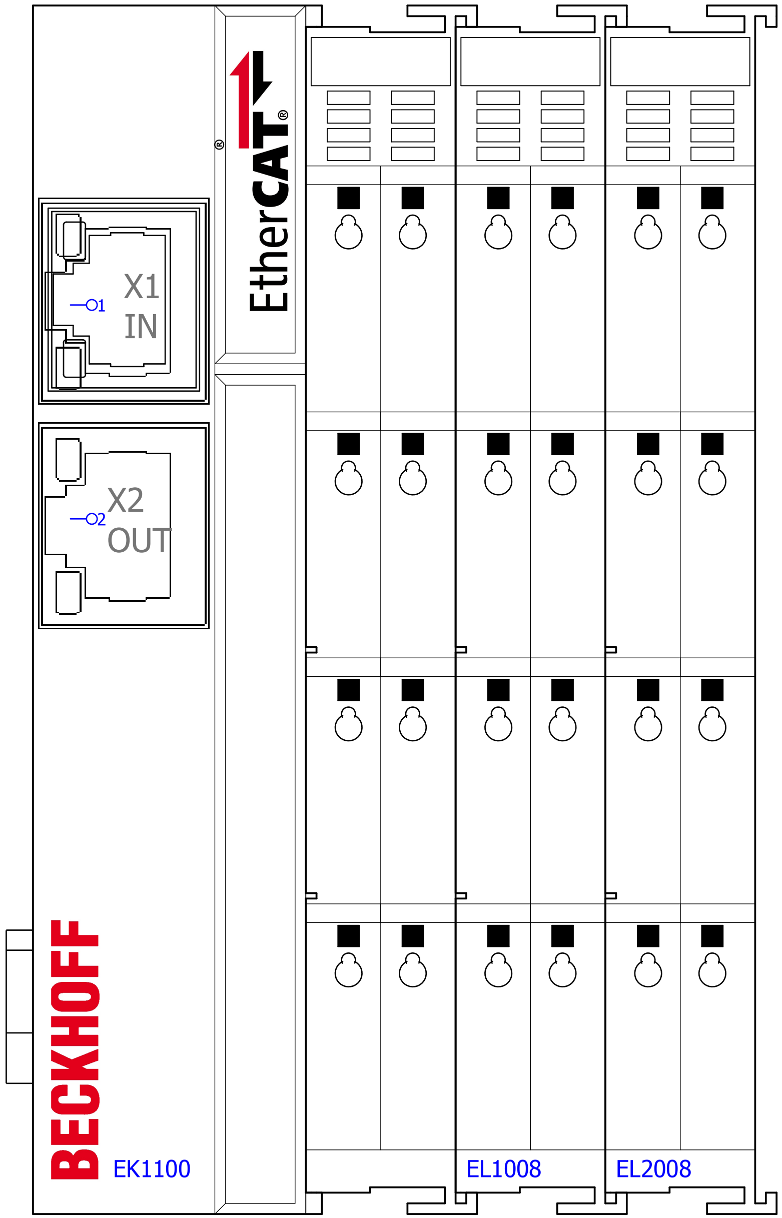
Campus 2
Morro do Lena – Alto do Vieiro
Apartado 4163
2411-901 Leiria – Portugal

Tel.: (+351) 244 820 300
estg@ipleiria.pt
www.ipleiria.pt/estg/

Project description	Project of the Electrical Cabinet for KUKA Interface
Job number	v1.0
Responsible for project	Maildo Lopes Pinto



Created on	8/24/2021				
Edit date	11/17/2021	by (short name) Maildo Lopes Pinto		Number of pages	13



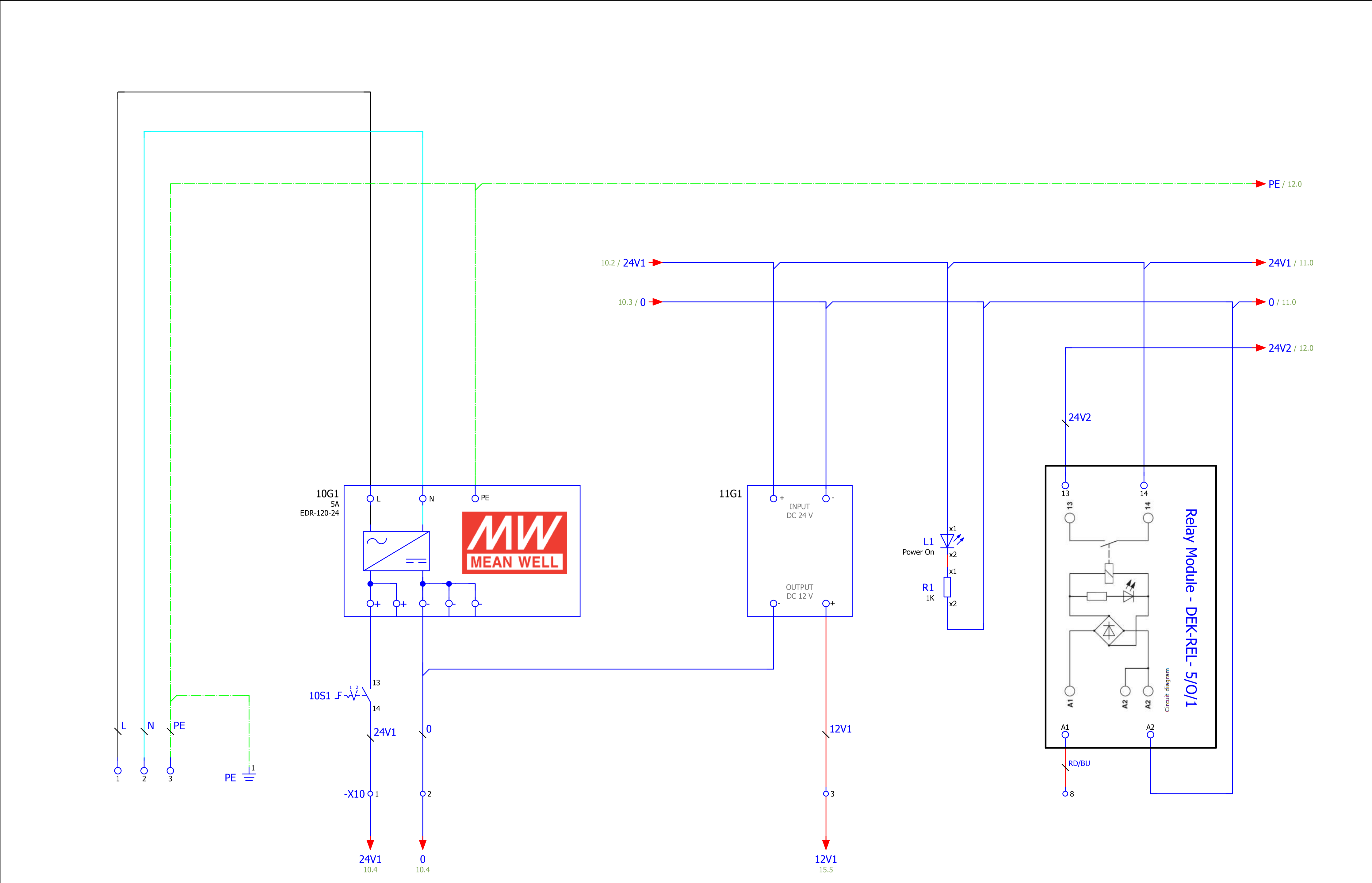
Date	9/9/2021
Ed.	MLP
Appr.	
Original	

POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO
Polytechnic of Leiria
Morro do Lena – Alto do Vieiro
Apartado 4163
2411-901 Leiria – Portugal
Tel.: (+351) 244 820 300
estg@ipleiria.pt
www.ipleiria.pt/estg/

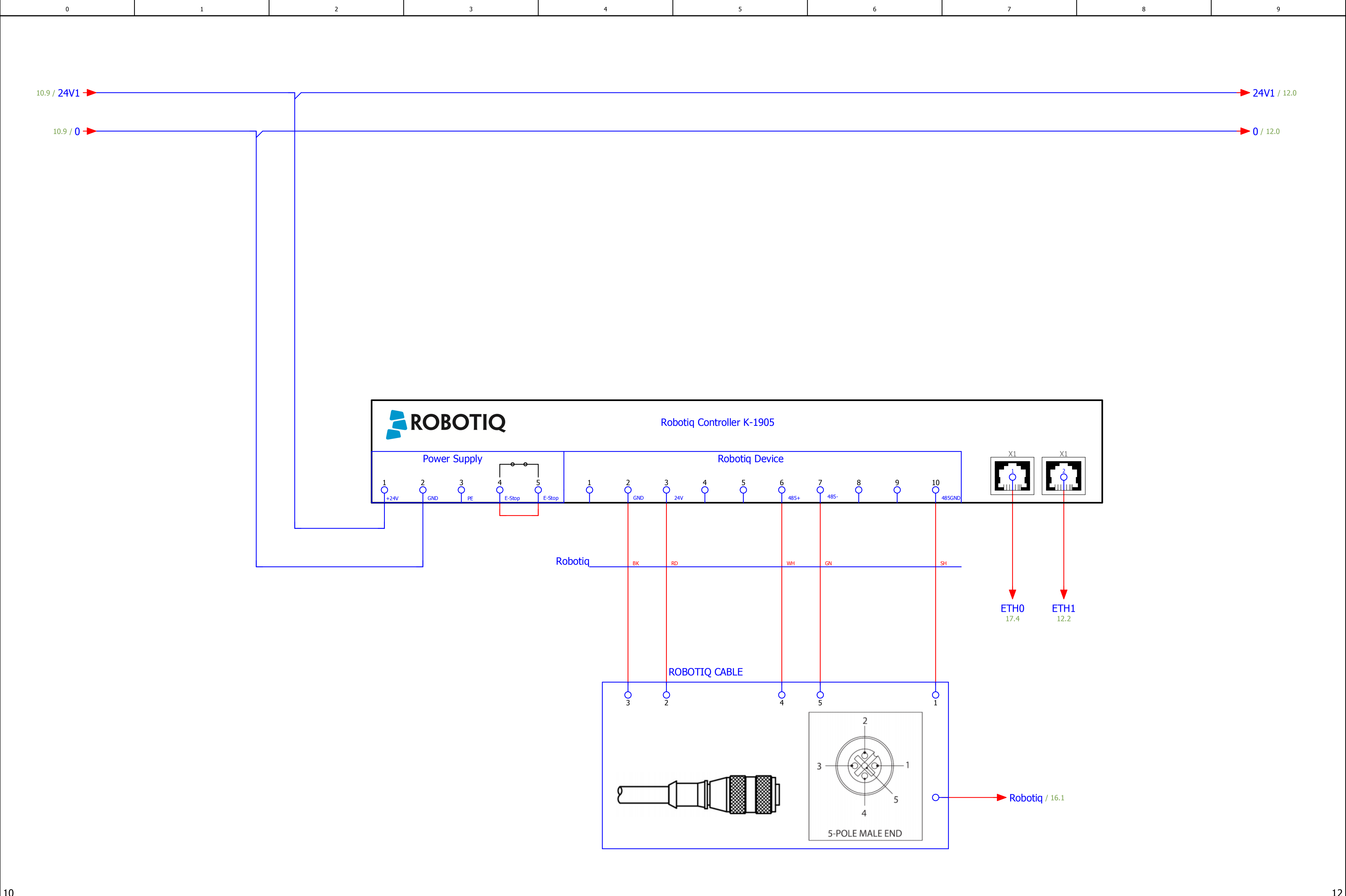
Descrição: Project of the Electrical Cabinet for KUKA Interface

Página: Beckhoff Connections

=
+
Nº Obra: v1.0



Date	11/17/2021	POLITÉCNICO DE LEIRIA — ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	**Polytechnic of Leiria** Morro do Lena – Alto do Vieiro Apartado 4163 2411-901 Leiria – Portugal	Tel.: (+351) 244 820 300 estg@ipleiria.pt www.ipleiria.pt/estg/	Descrição: Project of the Electrical Cabinet for KUKA Interface
Ed.	MLP				Página: Entry and Distribuition
Appr.					= / + / Nº Obra: v1.0
Original					Page 10 / Page 3 / 13



10 | 12

Date	9/30/2021
Ed.	MLP
Appr.	
Original	

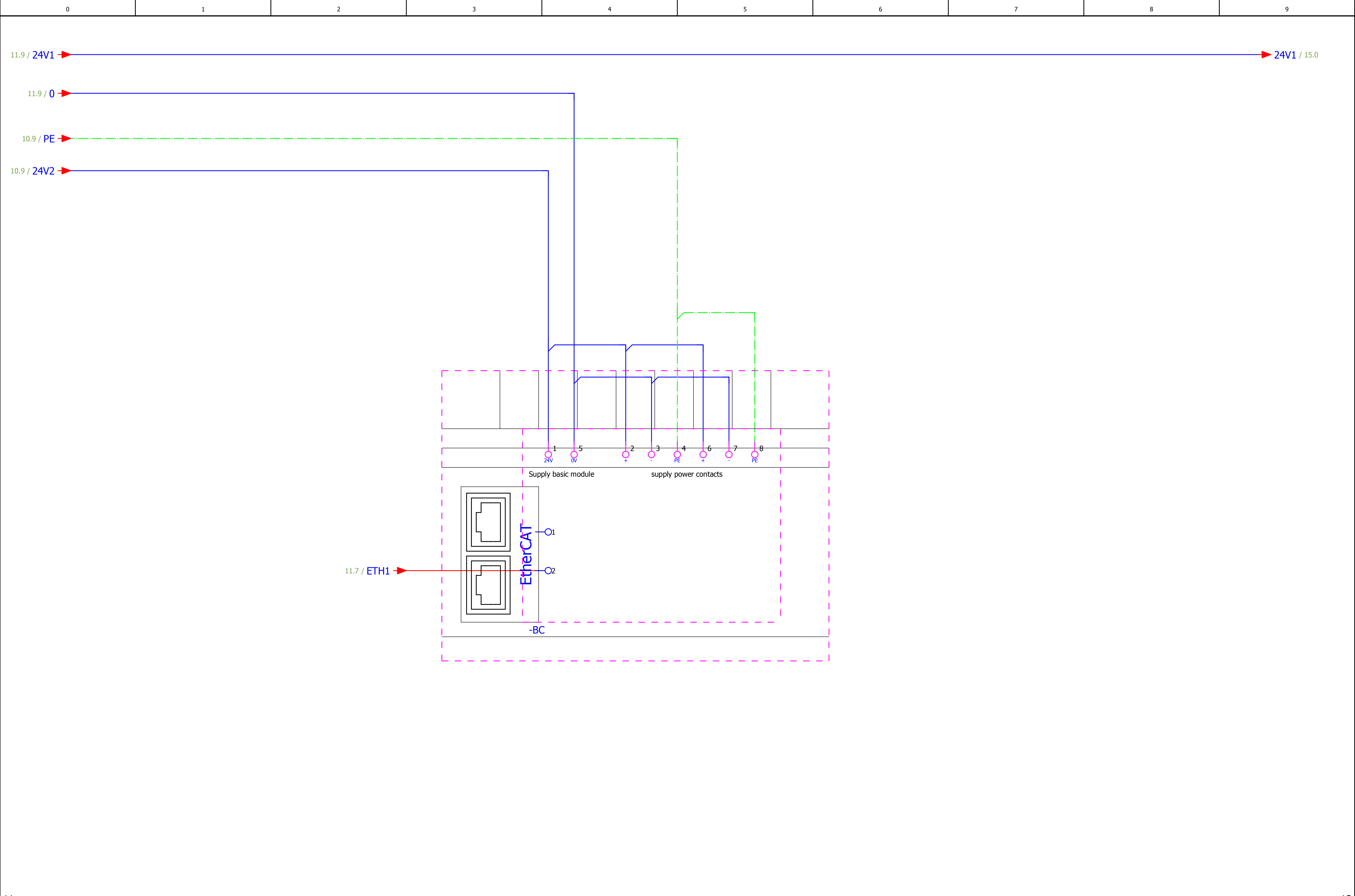
POLITÉCNICO DE LEIRIA — ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Polytechnic of Leiria
Morro do Lena – Alto do Vieiro
Apartado 4163
2411-901 Leiria – Portugal
Tel.: (+351) 244 820 300
estg@ipleiria.pt
www.ipleiria.pt/estg/

Descrição: Project of the Electrical Cabinet for KUKA Interface

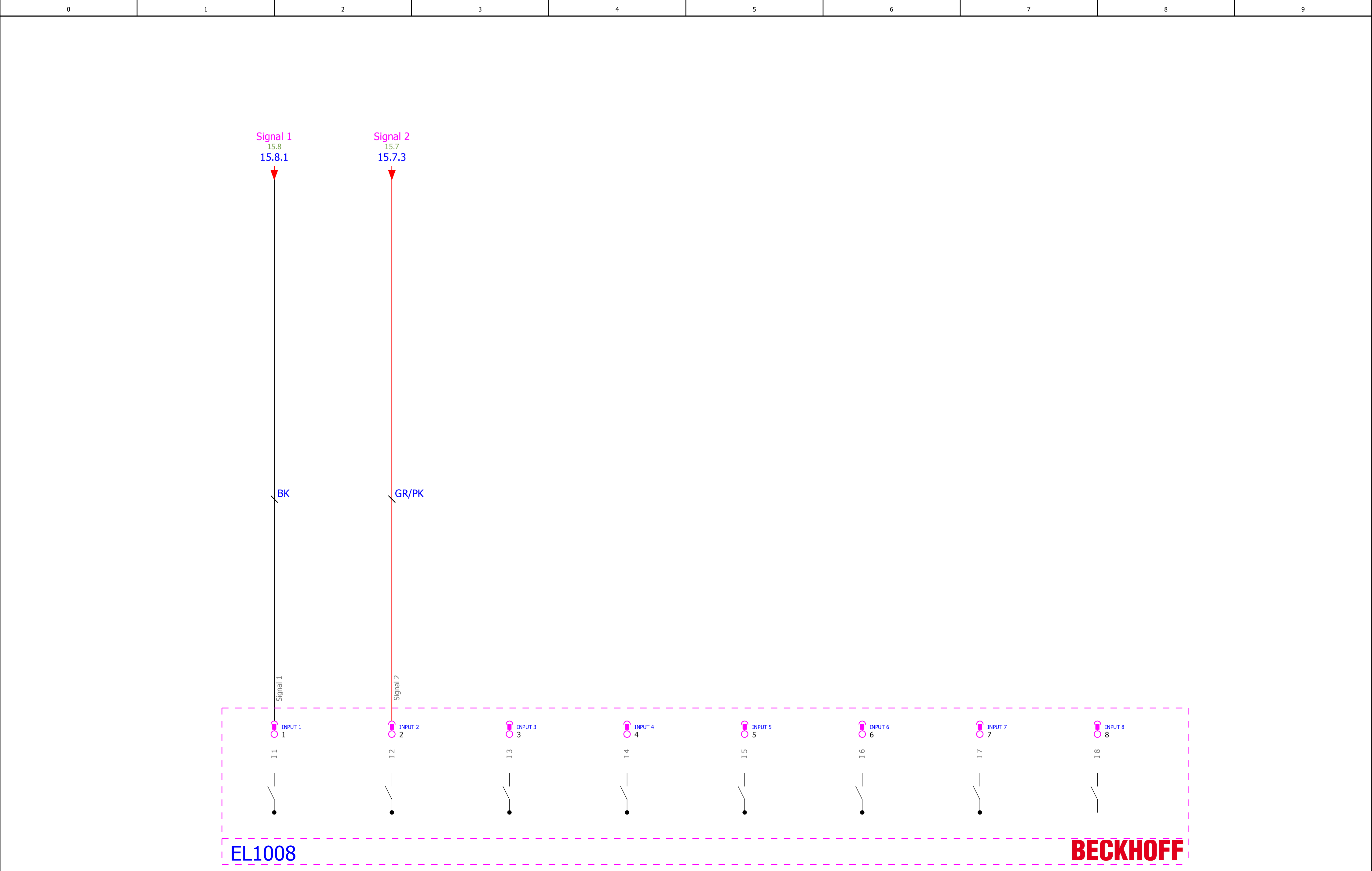
Página: Robotiq Controller

=
+
Nº Obra: v1.0



Date	9/30/2021	Polytechnic of Leiria	Descrição: Project of the Electrical Cabinet for KUKA Interface	Página: Beckhoff EK1100
Ed.	MLP	Morro do Lena – Alto do Vieiro Apartado 4163 2411-901 Leiria – Portugal		
Appr.		Tel.: (+351) 244 820 300 estg@ipleiria.pt www.ipleiria.pt/estg/		
Original				

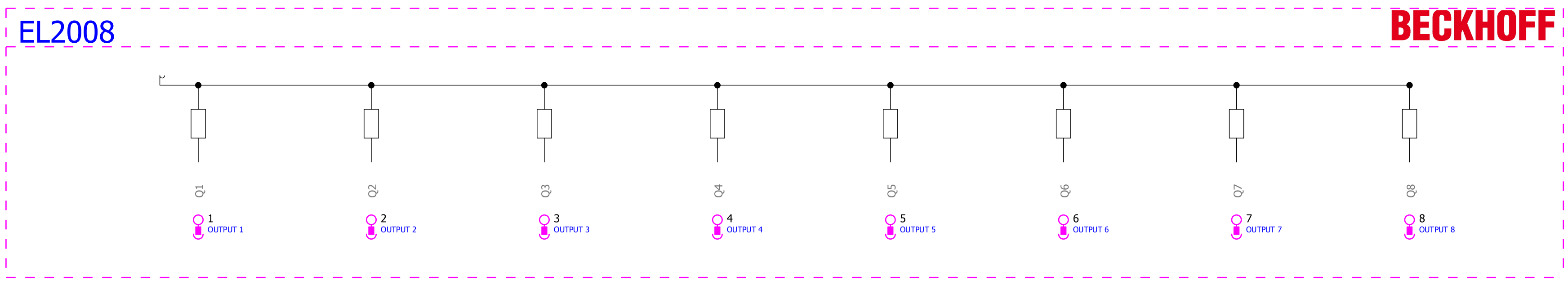
=
+
Nº Obra: v1.0
Page 12
Page 5 / 13

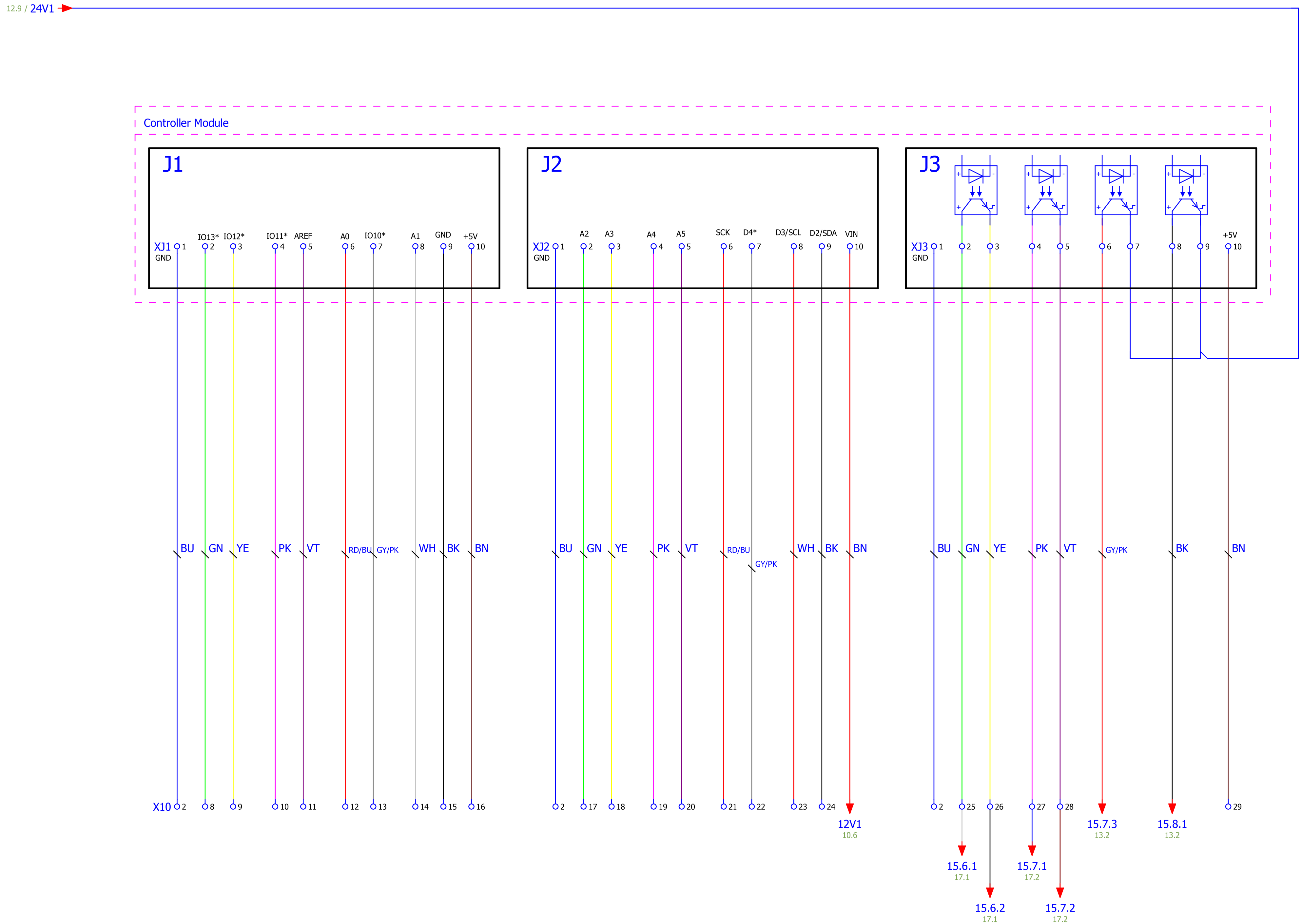


12

14

Date	9/16/2021	POLITÉCNICO DE LEIRIA — ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	**Polytechnic of Leiria**	Descrição:	Página:	=
Ed.	MLP		Morro do Lena – Alto do Vieiro / Apartado 4163 / 2411-901 Leiria – Portugal	Project of the Electrical Cabinet for KUKA Interface	Beckhoff EL1008	+
Appr.			Tel.: (+351) 244 820 300 / estg@ipleiria.pt / www.ipleiria.pt/estg/			Nº Obra: v1.0 — Page 13
Original						Page 6 / 13


EL2008
BECKHOFF
Q1
1 OUTPUT 1
Q2
2 OUTPUT 2
Q3
3 OUTPUT 3
Q4
4 OUTPUT 4
Q5
5 OUTPUT 5
Q6
6 OUTPUT 6
Q7
7 OUTPUT 7
Q8
8 OUTPUT 8



Date	11/17/2021
Ed.	MLP
Appr.	
Original	

POLITÉCNICO DE LEIRIA — ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

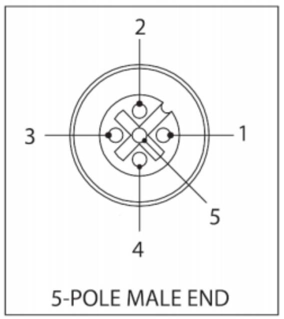
Polytechnic of Leiria
Morro do Lena – Alto do Vieiro
Apartado 4163
2411-901 Leiria – Portugal
Tel.: (+351) 244 820 300
estg@ipleiria.pt
www.ipleiria.pt/estg/

Descrição: Project of the Electrical Cabinet for KUKA Interface

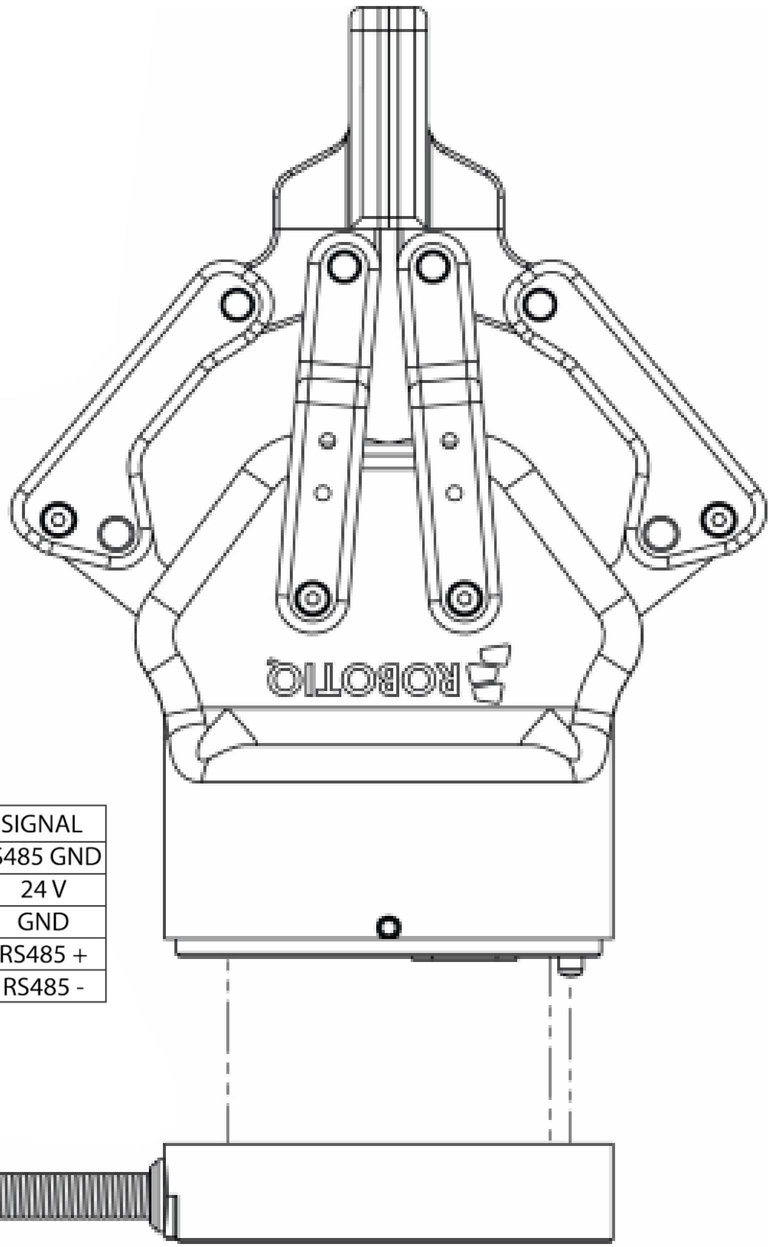
Página: Controller Module

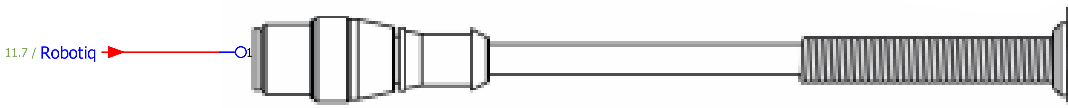
=
+
Nº Obra: v1.0

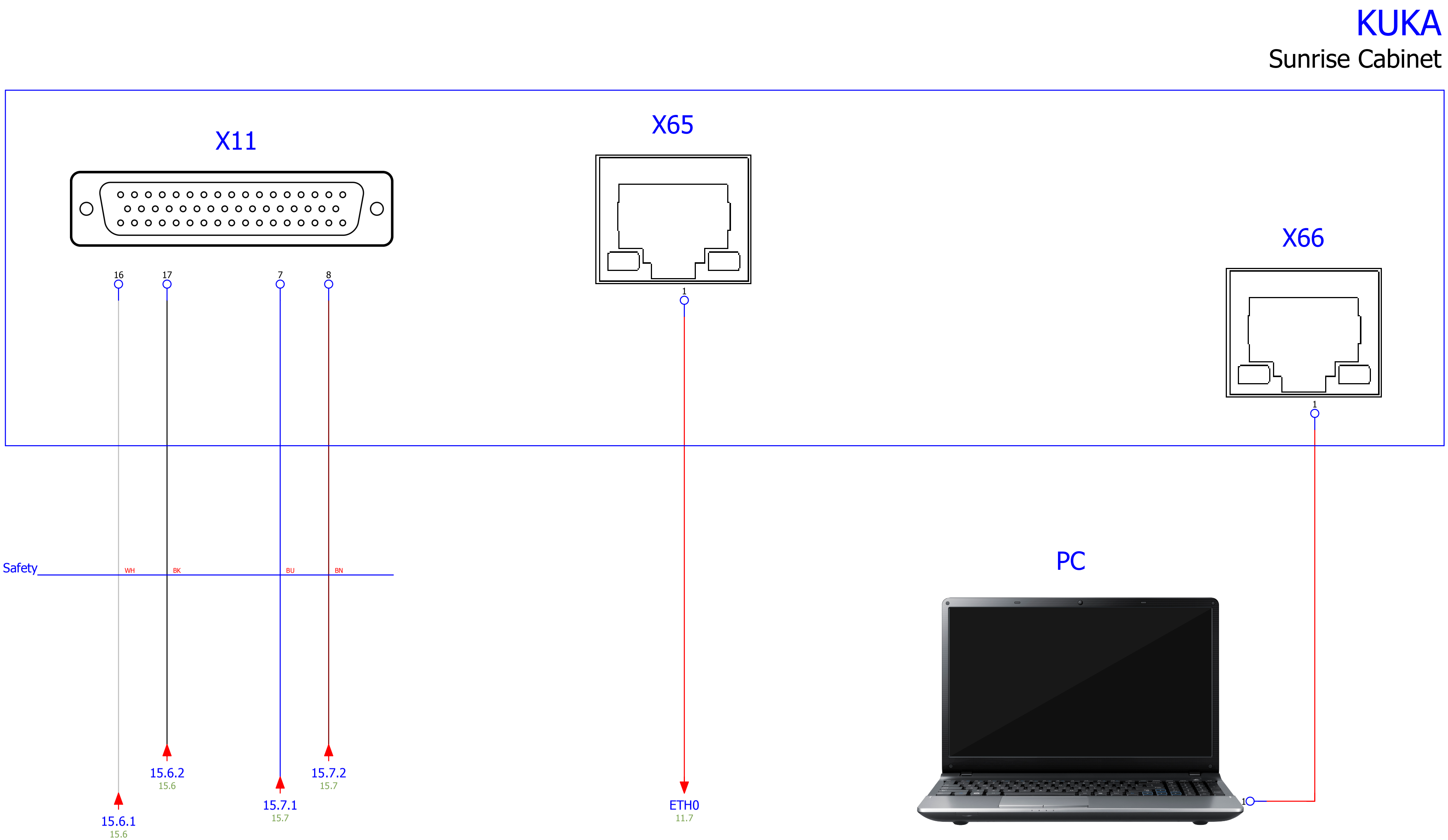
ROBOTIQ GRIPPER 2F-85



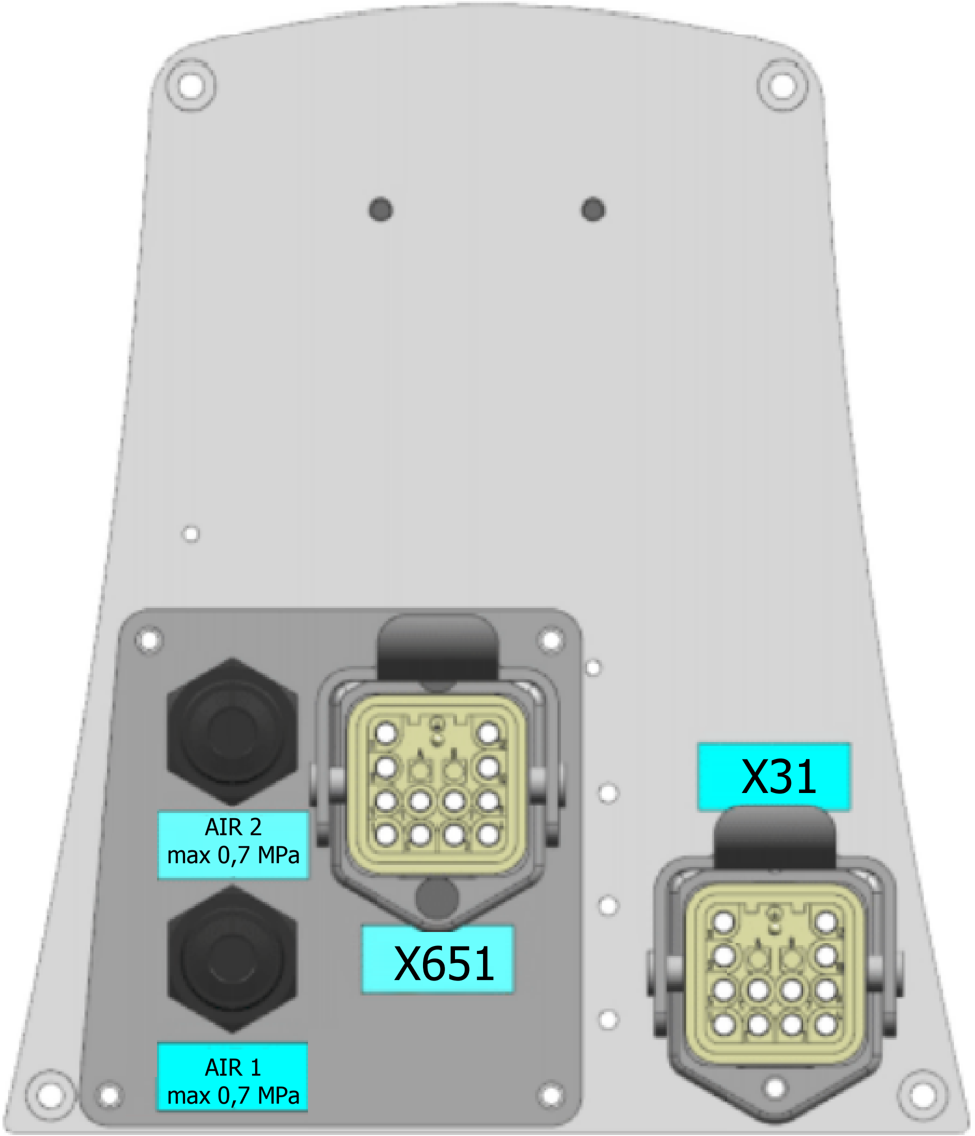
PIN	END OF CABLE COLOR	SIGNAL
1.	(SHIELD)	RS485 GND
2.	RED	24 V
3.	BLACK	GND
4.	WHITE	RS485 +
5.	GREEN	RS485 -

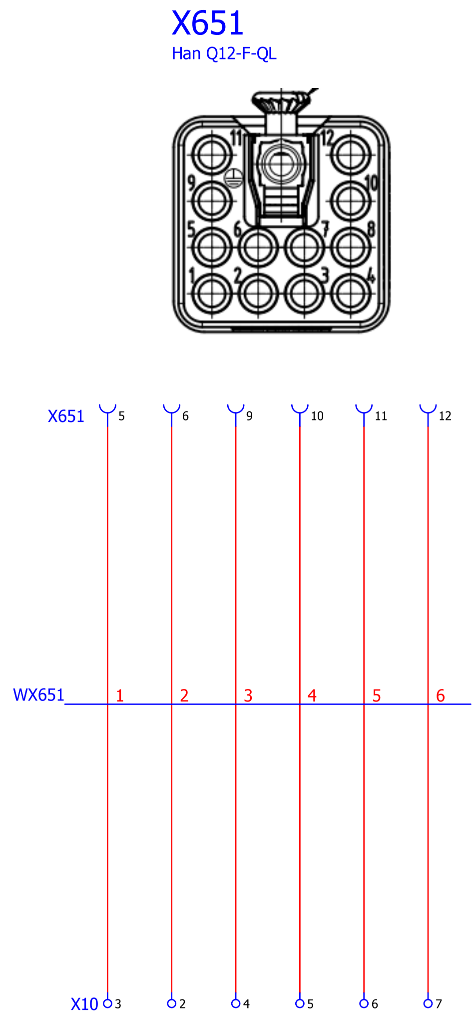






Date	9/9/2021	POLITÉCNICO DE LEIRIA / ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	**Polytechnic of Leiria** Morro do Lena – Alto do Vieiro Apartado 4163 2411-901 Leiria – Portugal	Tel.: (+351) 244 820 300 estg@ipleiria.pt www.ipleiria.pt/estg/	Descrição: Project of the Electrical Cabinet for KUKA Interface	Página: KUKA Sunrise Cabinet
Ed.	MLP					=
Appr.						+
Original						Nº Obra: v1.0 / Page 17 / Page 10 / 13





Date	9/16/2021	Polytechnic of Leiria Morro do Lena – Alto do Vieiro Apartado 4163 2411-901 Leiria – Portugal	Tel.: (+351) 244 820 300 estg@ipleiria.pt www.ipleiria.pt/estg/	Descrição: Project of the Electrical Cabinet for KUKA Interface
Ed.	MLP			Página: Interface A1 Pneumatic
Appr.				Nº Obra: v1.0
Original				Page 18 Page 11 / 13

Terminal diagram

F13_007

1	-Kuka Safety
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Strip

=+-X10

1	-W1
2	-Safety
3	
4	
5	
6	
7	
8	
9	
10	
11	

Function text	1	2	3	4	5	6	7	8	9	10	11	Designation (source)	Connection point	Terminal designation	Jumpers	PLC connection point	Designation (target)	Connection point	11	10	9	8	7	6	5	4	3	2	1	Page / column
												-15U1-XJ2	10	3	·		-11G1													/10.6
														4	·		-X651	1												/18.6
												-11G1		1	·		-10S1	14												/10.2
												-11G1		2	·		-10G1	-												/10.3
														2	·		-15U1-XJ1	1												/15.1
														8	·		-15U1-XJ1	2												/15.1
														9	·		-15U1-XJ1	3												/15.1
														10	·		-15U1-XJ1	4												/15.2
														11	·		-15U1-XJ1	5												/15.2
														12	·		-15U1-XJ1	6												/15.2
														13	·		-15U1-XJ1	7												/15.2
														14	·		-15U1-XJ1	8												/15.2
														15	·		-15U1-XJ1	9												/15.3
														16	·		-15U1-XJ1	10												/15.3
														2	·		-15U1-XJ2	1												/15.3
														17	·		-15U1-XJ2	2												/15.4
														18	·		-15U1-XJ2	3												/15.4
														19	·		-15U1-XJ2	4												/15.4
														20	·		-15U1-XJ2	5												/15.4
														21	·		-15U1-XJ2	6												/15.5
														22	·		-15U1-XJ2	7												/15.5
														23	·		-15U1-XJ2	8												/15.5
														24	·		-15U1-XJ2	9												/15.5
														2	·		-15U1-XJ3	1												/15.6
														29	·		-15U1-XJ3	10												/15.8
														5	·		-X651	2												/18.6
														6	·		-X651	3												/18.7
														7	·		-X651	4												/18.7
	WH											-KUKA	16	25	·		-15U1-XJ3	2										GN		/15.6
	BK											-KUKA	17	26	·		-15U1-XJ3	3										YE		/15.6

Terminal diagram

F13_007

1	-Kuka Safety
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Strip

=+-X10

1	-Safety
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Function text	1	2	3	4	5	6	7	8	9	10	11	Designation (source)	Connection point	Terminal designation	Jumpers	PLC connection point	Designation (target)	Connection point	11	10	9	8	7	6	5	4	3	2	1	Page / column
	BU											-KUKA	7	27	•		-15U1-XJ3	4											PK	/15.7
	BN											-KUKA	8	28	•		-15U1-XJ3	5											VT	/15.7

26

			Date	8/26/2021	IPL	Polytechinc of Leiria	Terminal diagram =+-X10		=
			Ed	MLP					+
			Appr		Project of the Electrical Cabinet for KUKA Interface			v1.0	Page 26.a
Modification	Date	Name	Original		Replacement of / Replaced by				Page 13 / 13

E

MODULE SCHEMATIC AND LAYOUT

This appendix shows the electrical schematics and PCB layouts for the flange module and the controller module.

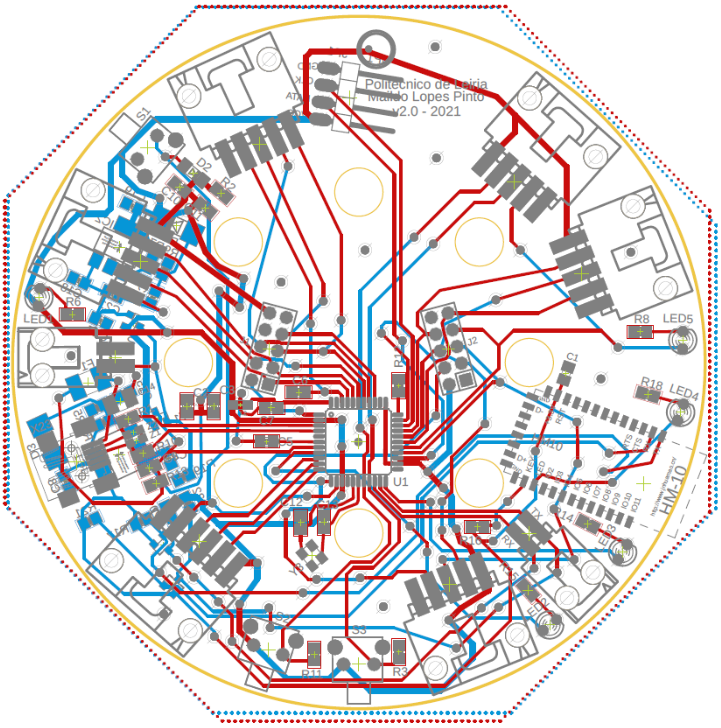
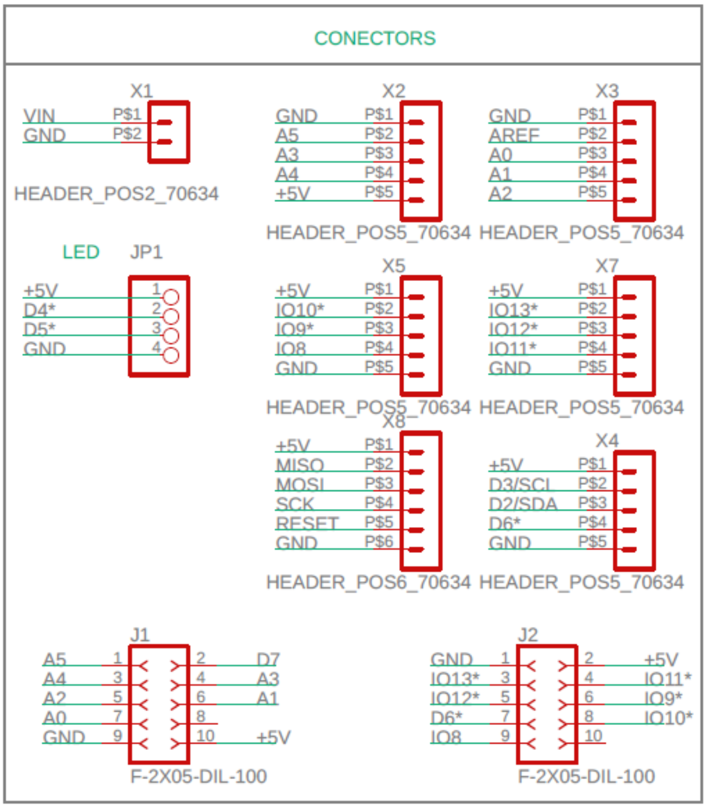


Figure 69: Flange module PCB layout.



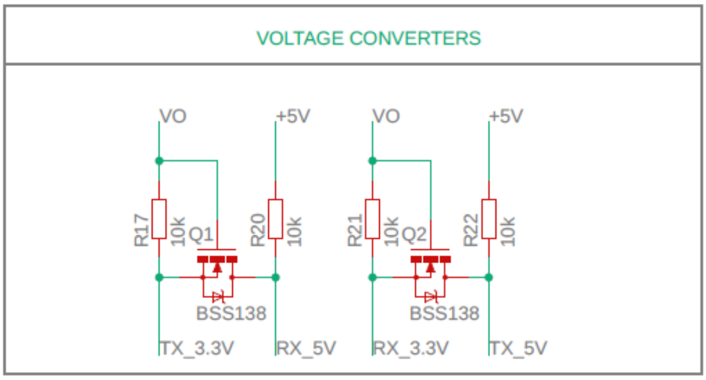


Figure 70: Flange module schematic - Conectors and voltage converters.

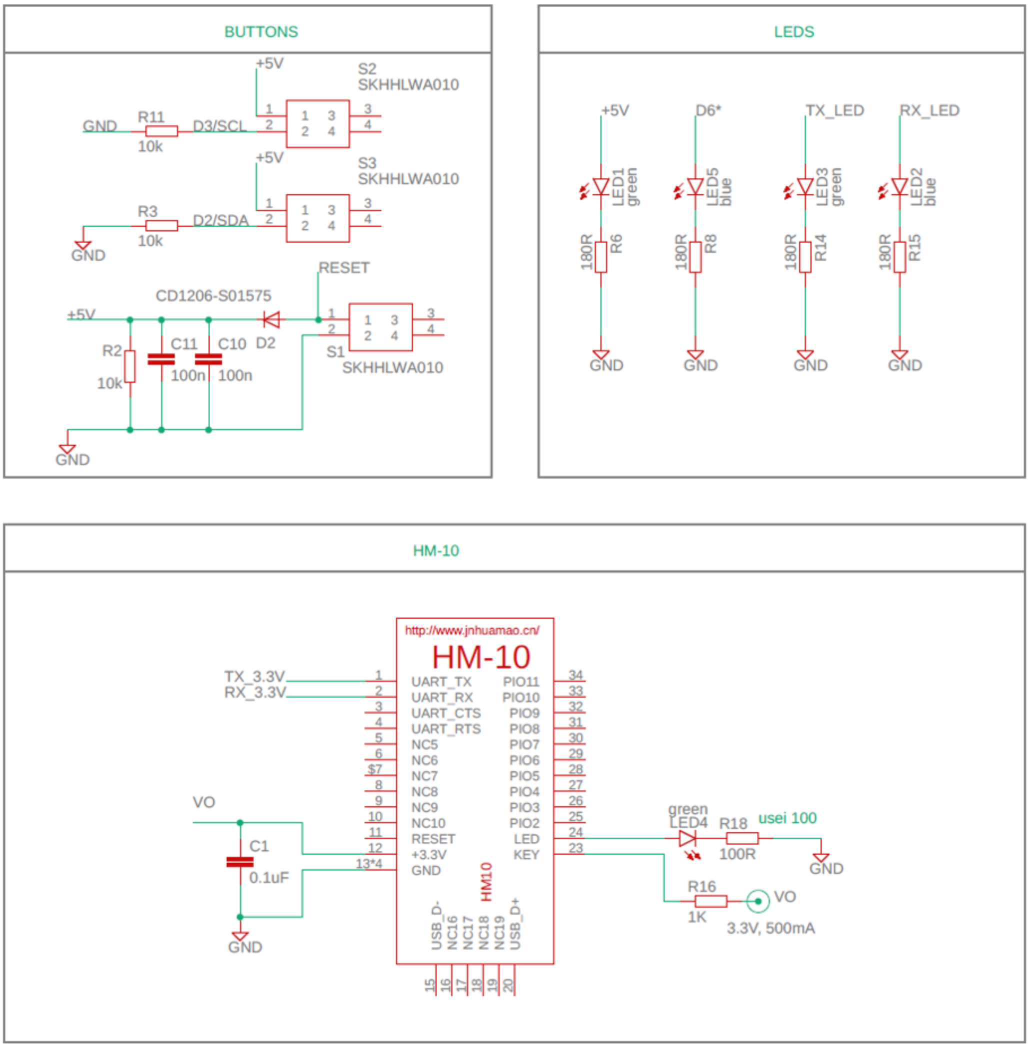


Figure 71: Flange module schematic - Buttons, leds and HM-10.

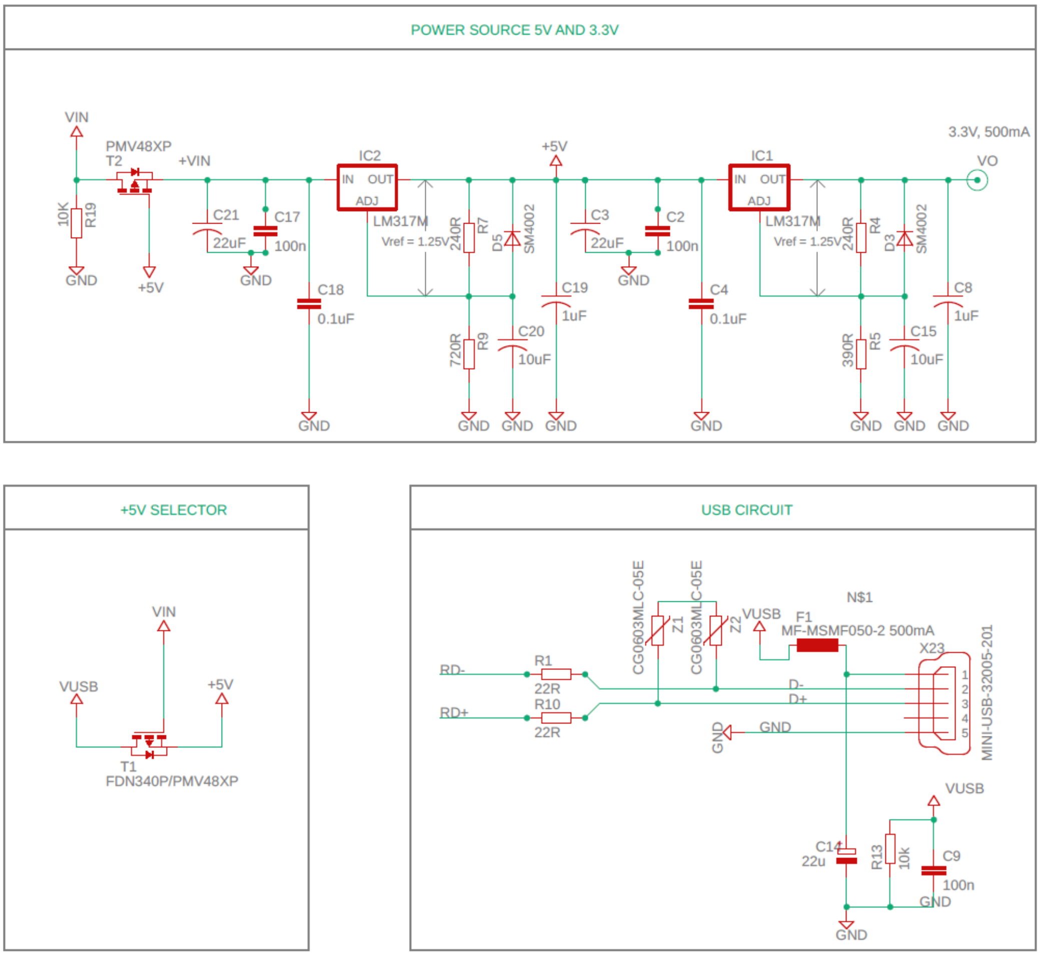


Figure 72: Flange module schematic - Power source, 5V selector and USB circuit.

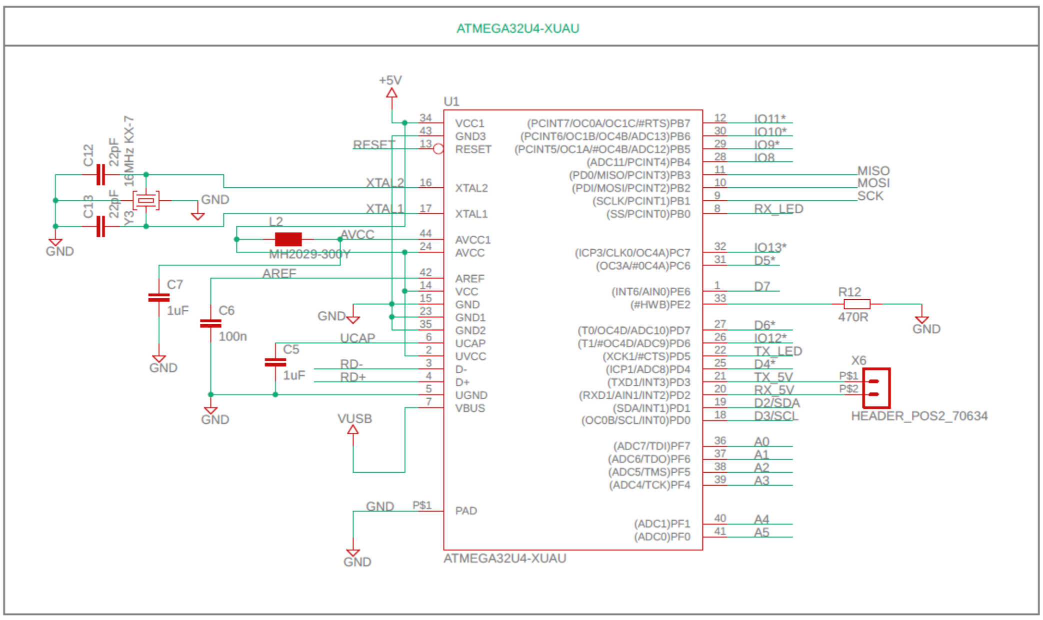


Figure 73: Flange module schematic - ATMEGA32U4-XUAU.

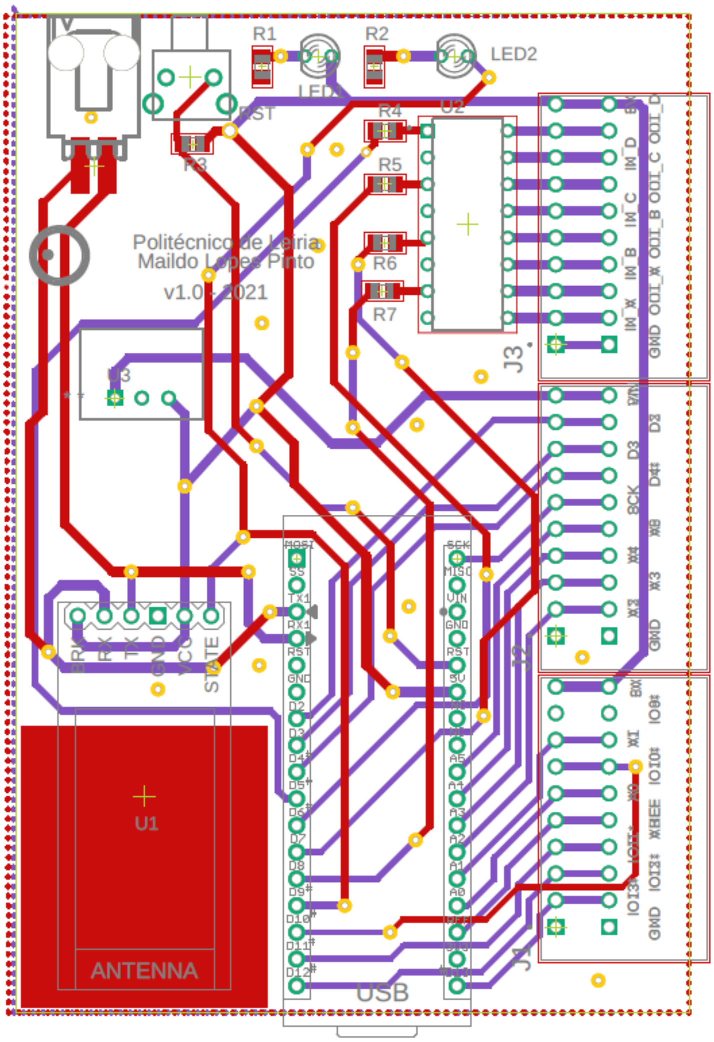


Figure 74: Controller module layout.

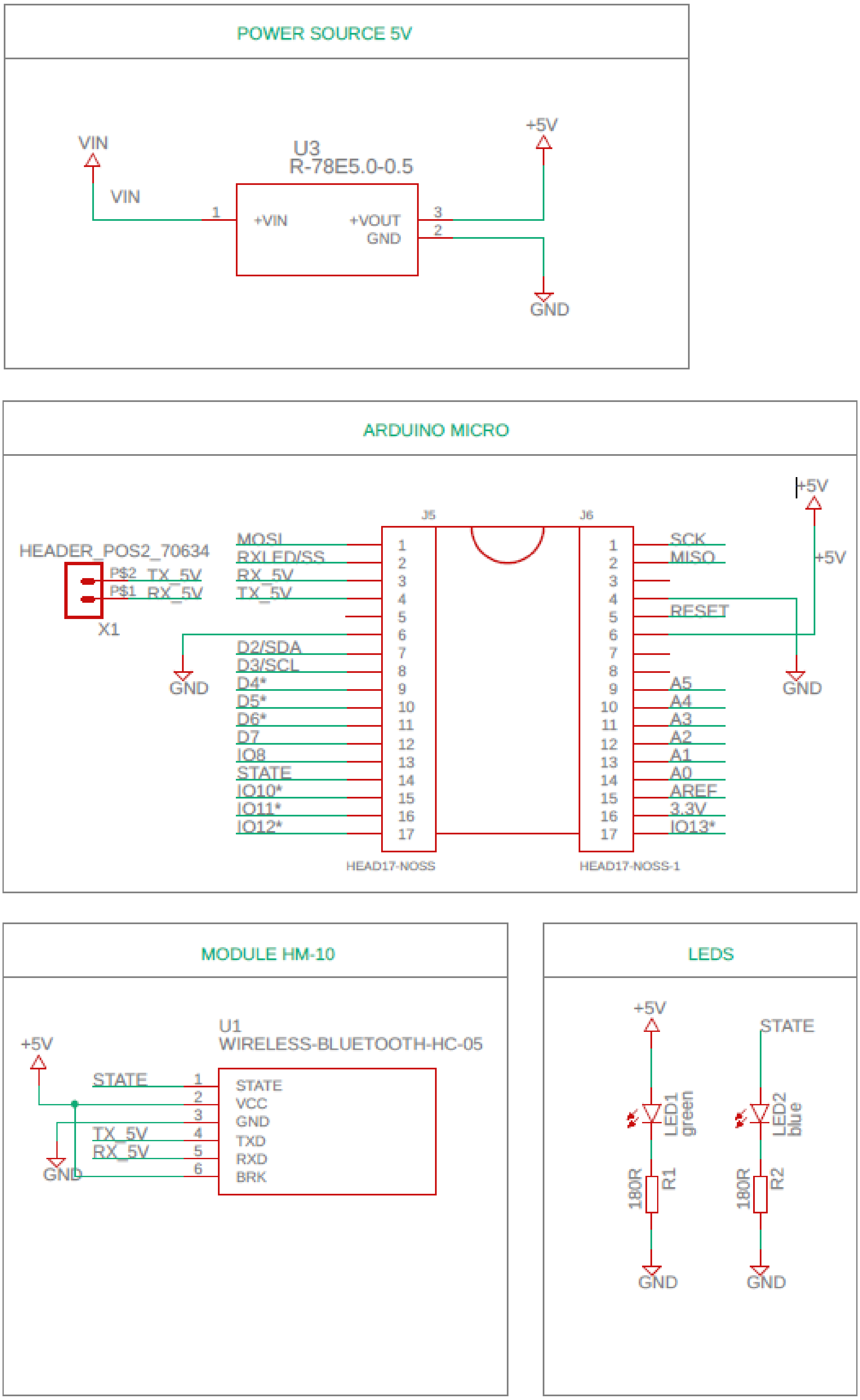


Figure 75: Controller module schematic - Power source, Arduino micro, HM-10 and led.

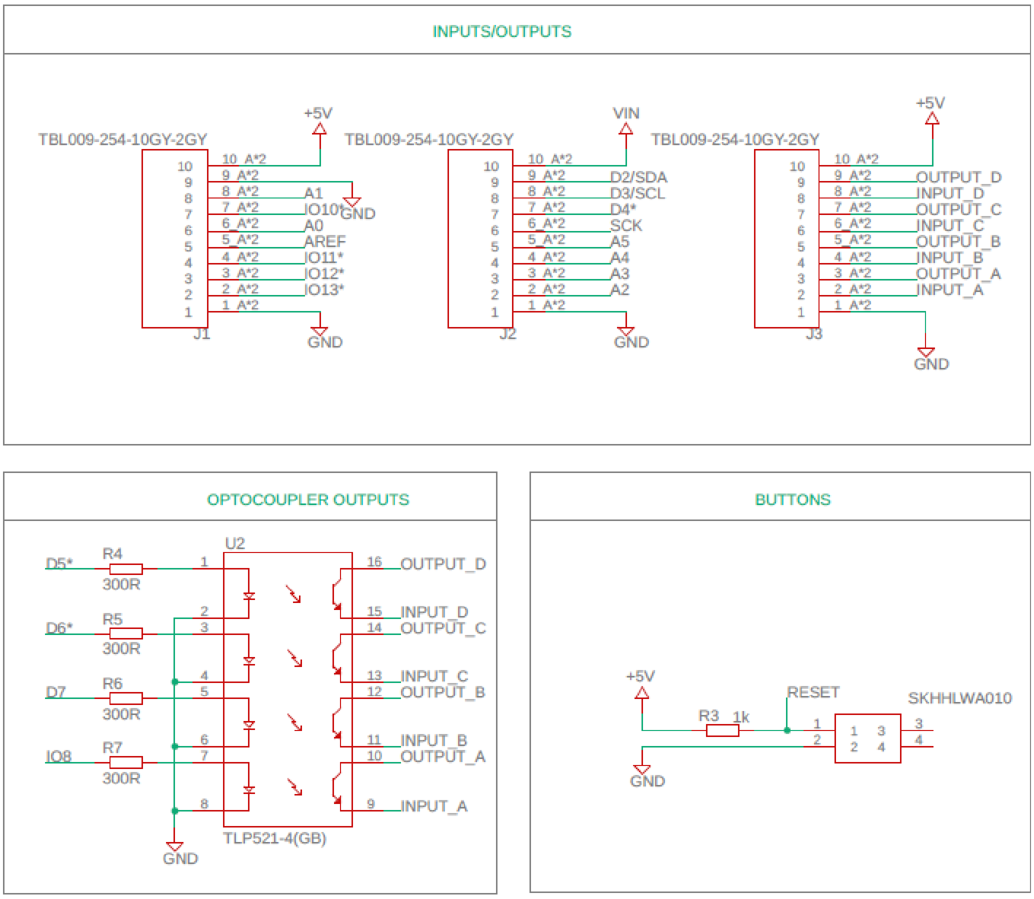


Figure 76: Controller module schematic - Inputs/outputs, optocoupler outputs and buttons.