Programming of industrial robot Fanuc LR Mate 200iD

CIRILO-PIÑON, Oscar Ivan
MASTACHE-MASTACHE, Jorge Edmundo
LÓPEZ-RAMÍREZ, Roberto
NAMIGTLE-JIMÉNEZ, Jesús

ECORFAN®

ECORFAN®

Chief Editor
VARGAS-DELGADO, Oscar. PhD

Authors
CIRILO-PIÑON, Oscar Ivan. BsC
MASTACHE-MASTACHE, Jorge Edmundo. PhD
LÓPEZ-RAMÍREZ, Roberto. PhD
NAMIGTLE-JIMÉNEZ, Jesús. PhD

Executive Director
RAMOS-ESCAMILLA, María. PhD

Editorial Director
PERALTA-CASTRO, Enrique. MsC

Web Designer
ESCAMILLA-BOUCHAN, Imelda. PhD

Web Diagrammer
LUNA-SOTO, Vladimir. PhD

Editorial Assistant
SORIANO-VELASCO, Jesus. BsC

Translator
DÍAZ-OCAMPO, Javier. BsC

Philologist
RAMOS-ARANCIBIA, Alejandra. BsC

Programming of industrial robot Fanuc LR Mate 200iD

First edition ISBN: 978-607-8695-79-9
Editorial Seal ECORFAN: 607-8695
Control number B: 2022-02
Classification B (2022): 280922-0002

Books

Definition of Books

Scientific Objetives

Support the International Scientific Community in its written production of Science, Technology in Innovation in the Research Areas CONACYT and PRODEP

ECORFAN-Mexico SC is a Scientific and Technological Company in contribution to the formation of Human Resource focused on the continuity in the critical analysis of International Research and is attached to the RENIECYT of CONACYT with number 1702902, its commitment is to disseminate research and contributions of the International Scientific Community, of academic institutions, agencies and entities of the public and private sectors and contribute to the linking of researchers who carry out scientific activities, technological developments and training of specialized human resources with governments, companies and social organizations.

Encourage the interlocution of the International Scientific Community with other study centers in Mexico and abroad and promote a wide incorporation of academics, specialists and researchers to the publication Seriate in Niches of Science of Autonomous Universities - State Public Universities - Federal IES - Polytechnic Universities - Technological Universities - Federal Technological Institutes - Normal Schools - Decentralized Technological Institutes - Intercultural Universities - S & T Councils - CONACYT Research Centers.

Scope, Coverage and Audience

Books is a Product edited by ECORFAN-Mexico S.C in its Holding with repository in Mexico, is an arbitrated and indexed scientific publication. It supports a wide range of contents that are evaluated by academic peers by the Double-Blind method, around topics related to the theory and practice of the Research Area CONACYT and PRODEP, respectively, with different approaches and perspectives, contributing to the diffusion of the development of Science Technology and Innovation that allow the arguments related to the decision making and influence in the formulation of international policies in the Field of Sciences. The editorial horizon of ECORFAN-Mexico® extends beyond the academy and integrates other segments of research and analysis outside the scope, as long as they comply with the requirements of rigorous argumentative and scientific, as well as addressing issues of general and current interest of the International Scientific Society.

Editorial Board

Arbitration Committee

Assignment of Rights

The sending of a Scientific Work to ECORFAN Books emanates the commitment of the author not to submit it simultaneously to the consideration of other scientific publications for it must complement the Originality Format for its Scientific Work.

The authors sign the Authorization Format for their Scientific Work to be disseminated by means that ECORFAN-Mexico, S.C. in its Holding Mexico consider relevant for the dissemination and dissemination of its Scientific Work by giving up its Scientific Work Rights.

Declaration of Authenticity

Indicate the Name of 1 Author and 3 Coauthors at most in the participation of the Scientific Work and indicate in extensive the Institutional Affiliation indicating the Unit.

Identify the Name of 1 Author and 3 Co-authors at most with the CVU Scholarship Number-PNPC or SNI-CONACYT- Indicating the Researcher Level and their Google Scholar Profile to verify their Citation Level and H index.

Identify the Name of 1 Author and 3 Coauthors at most in the Science and Technology Profiles widely accepted by the International Scientific Community ORC ID - Researcher ID Thomson - arXiv Author ID - PubMed Author ID - Open ID respectively

Indicate the contact for correspondence to the Author (Mail and Telephone) and indicate the Researcher who contributes as the first Author of the Scientific Work.

Plagiarism Detection

All Scientific Works will be tested by plagiarism software PLAGSCAN if a plagiarism level is detected Positive will not be sent to arbitration and will be rescinded from the reception of the Scientific Work notifying the Authors responsible, claiming that academic plagiarism is criminalized in the Penal Code.

Arbitration Process

All Scientific Works will be evaluated by academic peers by the Double Blind method, the Appeal arbitration is a requirement for the Editorial Board to make a final decision that will be unappeasable in all cases. MARVID® is a derivative brand of ECORFAN® specialized in providing the expert evaluators all of them with Doctorate degree and distinction of International Researchers in the respective Councils of Science and Technology the counterpart of CONACYT for the chapters of America-Europe-Asia-Africa and Oceania. The identification of the authorship should only appear on a first removable page, in order to ensure that the Arbitration process is anonymous and covers the following stages: Identification of ECORFAN Books with their author occupation rate - Identification of the Authors and Coauthors - Detection of Plague PLAGSCAN - Review of Formats of Authorization and Originality-Allocation to the Editorial Board- Allocation of the pair of Expert Arbitrators-Notification of Opinion-Declaration of Observations to the Author-Comparison of the Scientific Work Modified for Editing-Publication.

Programming of industrial robot Fanuc LR Mate 200iD

Programación de robot industrial Fanuc LR Mate 200iD

CIRILO-PIÑON, Oscar Ivan*[1], MASTACHE-MASTACHE, Jorge Edmundo[2,3], LÓPEZ-RAMÍREZ, Roberto[2] and NAMIGTLE-JIMÉNEZ, Jesús[2,3]

[1]*Student TESJO, Mechatronic Engineering Division*
[2]*TecNM/Tecnológico de Estudios Superiores de Jocotitlán*
[3]*Universidad de Ixtlahuaca, Facultad de Ingeniería*

ID 1st Author: *Oscar Ivan, Cirilo-Piñon* / **ORC ID**: 0000-0002-5108-6268

ID 1st Co-author: *Jorge Edmundo, Mastache-Mastache* / **ORC ID**: 0000-0001-6104-6764, **CVU CONACYT ID**: 544943

ID 2nd Co-author: *Roberto, López-Ramírez* / **ORC ID**: 0000-0001-8341-3684, **CVU CONACYT**: 233228

ID 3rd Co-author: *Jesús, Namigtle-Jiménez* / **ORC ID**: 0000-0002-0980-4592, **CVU CONACYT**: 624757

DOI: 10.35429/B.2022.2.1.121

*jorge.mastache@tesjo.edu.mx

O. Cirilo, J. Mastache, R. López & J. Namigtle (AA. VV.). Programming of industrial robot Fanuc LR Mate 200iD. Books-©ECORFAN-México. 2022.

Programming of industrial robot Fanuc LR Mate 200iD

The Book will offer selected contributions of researchers that contribute to the scientific dissemination activity of the Tecnológico de Estudios Superiores de Jocotitlán for its research area in the function of the University facing the challenges of the Knowledge Society. In addition to having a total evaluation, in the hands of the directors of the Tecnológico de Estudios Superiores de Jocotitlán, quality and punctuality are collaborated in their chapters, each individual contribution was arbitrated to international standards (RESEARCH GATE, MENDELEY, GOOGLE SCHOLAR and REDIB), the Book thus proposes to the academic community, the recent reports on the new developments in the most interesting and promising areas of research in the function of the University before the challenges of the Knowledge Society.

Content

Chapter 1 Fanuc LR Mate 200iD robot system composition

Abstract

The fourth industrial revolution brought a great change in our lives, since it provided a technological evolution developing elements such as the internet, artificial intelligence and robotics. Things that today are essential to learn since they are basic topics if we observe that the evolution continues to increase. Focusing on robotics, its application in industry offered nothing but benefits for them due to the increase in production capacity, reduction of manpower, operation at high precision, etc.. All this, using the well-known "Industrial Robots", which are robotic systems designed to respond to stimuli perceived by him, performing an intelligent action, where the biggest challenge to manipulate them is to know their composition and learn to program them.

On the other hand, different companies such as FANUC, have developed new technologies that maintain communication between the user and the robot, in addition to facilitating programming by generating intuitive elements such as controllers and remote-control devices (teach pendant). The importance of these elements lies in the fact that they can be used to develop trajectories that allow the robot to perform actions or applications in an industrial environment, from the recording of a finite series of points in space. Therefore, the aim of the chapter is to explain the basic composition of the Fanuc LR Mate 200id robot, describing its main features, as well as adding technical data that will help to better understand the subject.

Robotic system, Evolution, Manipulate

1.1 Introduction

Since ancient times, with the arrival of the machines, civilization has always shown an interest in replicating movements, reactions, behaviors, and tasks, both human and animal. Scientists also supported the evolution of robotics, among them : Juanelo Torriano with "The stick man" (16th century), Jacques de Vaucanson with "Duck with a digestive system" (17th century), Walter Gray with "Spectaculatrix Machine or Turtles" (1948-1949), among many others. (Salazar García, 2018), (Guarnizo Marin et al., 2021).

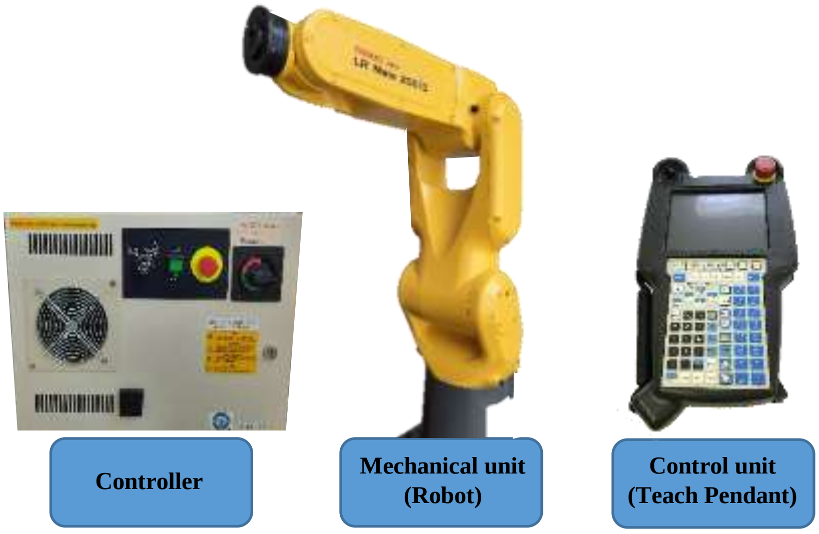
The constant technological advancement and the integration of robots more and more noticeable every day, promoted that companies did not hesitate to experiment with adding these robots to their companies. In 1961, General Motors integrated the first industrial robot to help in the production of automobiles, which generated profits, offered efficiency, and also increased the production capacity. Nowadays, there are many manufacturing companies of industrial robots, within which FANUC could be considered as one of the most famous companies, which in turn manufactures all the tools to manipulate them. From this arises the need to learn to control them while knowing their conformation (Mikhail & Wang, 2018).

This chapter focuses on describing the composition of the Fanuc LR Mate 200id robotic system, explaining the most important parts and characteristics that constitute it, such as its mechanical unit, the R-30iB plus controller and the TP remote control unit.

1.2 Fanuc LR Mate 200iD robot composition

Although the Fanuc LR Mate 200iD robot contains hundreds of parts that make it up, these can be divided into three main ones: the controller, a mechanical unit (the robot) and a control unit (iPendant or Teach pendant). That is why these components will be explained below.

Figure 1.1 Main parts of the Fanuc LR Mate 200iD robot



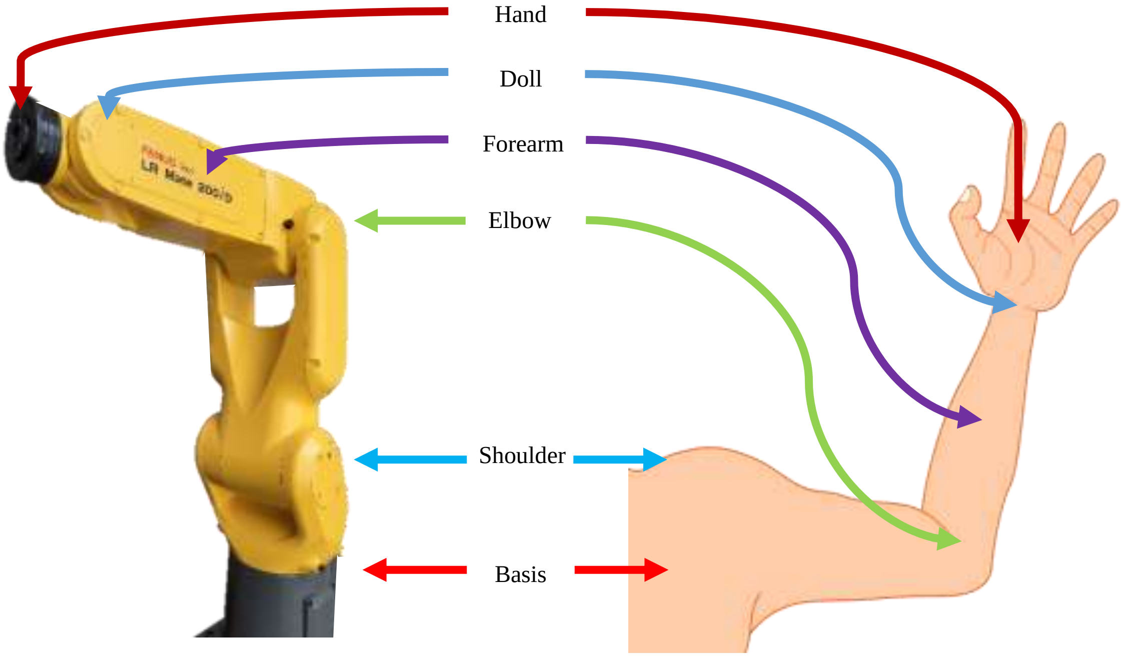
Source: Elaborated by the work team

1.2.1 Description of the mechanical unit (Fanuc LR Mate 200iD)

The LR Mate series is a family of robots considered as "small payload tabletop robots", which are mostly made up of six degrees of freedom and are compact robots like the size of a human arm, that are used for handling and fast processing of pieces between 7 to 14 kg depending on their termination in the nomenclature. In addition, its modular construction is driven by servomotors designed to offer stability that eliminates the greatest amount of vibration and allows more precise movements. (FANUC America Corporation, 2016)

The LR Mate 200iD mechanical unit version is named by the same company that manufactured it as "minirobot". This in turn is made up of 6 axes that, fully stretched, can reach up to 717 mm long, approaching the standard length of a human arm. In addition, it can support a wrist load of up to 7kg. The main constitution is based on axes and links where, the axes are numbered in ascending order if we compare it to a human arm (see figure 1.2), then the constitution would start from a base (J1) and continue with the shoulder (J2), elbow (J3), upper arm rotation (J4), wrist tilt (J5) and wrist rotation (J6), and on some occasions, the fingers may be considered are the tools that are mounted on the robot.

Figure 2.2 Physical comparison of an industrial robot and a human arm



Source: Elaborated by the work team

1.2.1.1 Characteristics

As previously was mentioned, the LR Mate 200iD robot is generally made up of a total of 6 axes. This constitution is mainly based on links and axes where the former are the rigid parts of the robot that function as the union between axes that also allow movement between them. On the other hand, the axes are the moving parts that contain a specific range of rotation and are listed in ascending order (guiding from figure 1.2) starting from:

- The base (J1).
- The shoulder (J2).
- The elbow (J3).
- Rotation in the upper arm (J4).
- The tilt of the wrist (J5).
- Rotation on the robot flange (J6).

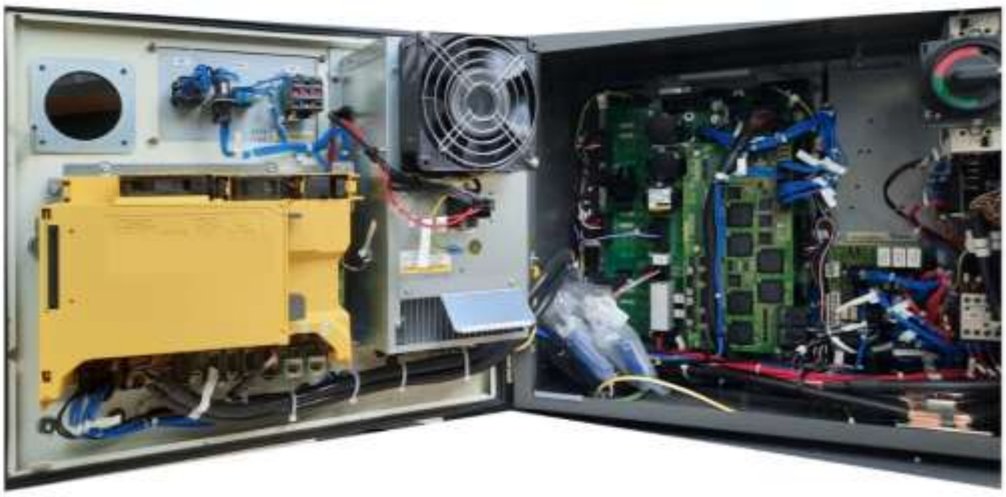
This mechanical unit was developed in a compact way to offer more efficient productivity, using smooth movements at high speeds without generating vibration, thus providing shorter cycle times (Fanuc, 2018a). Other important features for this version of the robot are:

- Reach: With all axes extended, the maximum reach is 717mm.
- Mounting method: On floor, upside down or at an angle.
- Maximum payload weight at the wrist: 7 kg.
- Repeatability: 0.01mm
- Mechanical weight: 25 kg

1.2.2 Controller R-30iB PLUS

The controller is the heart of the robot, this is where essential elements such as the power switch, an emergency stop button, the operation mode switch, etc., are located. In addition, it contains the computer and all the components that make the mechanical unit work (see figure 1.3) and that maintains communication with the user through the Teach pendant. This controller was designed to improve robot performance through cycle times, speed, accuracy, and safety. In addition to the fact that its construction supports the energy saving, it saves space and has easy communication with the user, this is due to the fact that it has intuitive control and maintains an easy-to-understand diagnostic system (Fanuc, 2018b).

Figure 3.3 Controller R-30iB PLUS



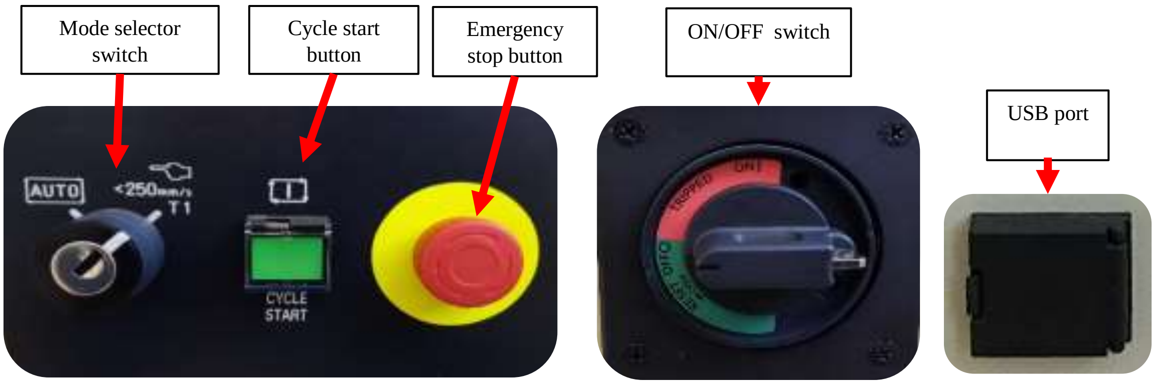
Source: Elaborated by the work team

1.2.2.1 Operator's panel

On the outside of the controller is the so-called "operator panel", which is an area where the buttons and switches that allow the operation of the mechanical unit are located. These elements can be better visualized in figure 1.4 and are mainly made up of:

- **ON/OFF switch:** Turns the on/off state of the robot, the main controller and teaching terminal. In addition, if it is turned in the reverse direction at power on, the controller cabinet can be opened.
- **Emergency stop:** Stops the actions or mobile functions of the robot when it is activated, and it can only move by deactivating it.
- **Operation mode selector:** It is a switch that allows us to choose the type of mode in which we will work automatic or manual, this is done through the TP.
- **Cycle start button:** Starts a programming cycle when the robot is in its automatic mode, unless a program works through loops, this button will have to be activated at each cycle start.
- **USB port:** It is a port designed to have better access to the controller from a USB memory and thus be able to work with backups, among other options.

Figure 4.4 Operator's panel for controller R-30iB Plus



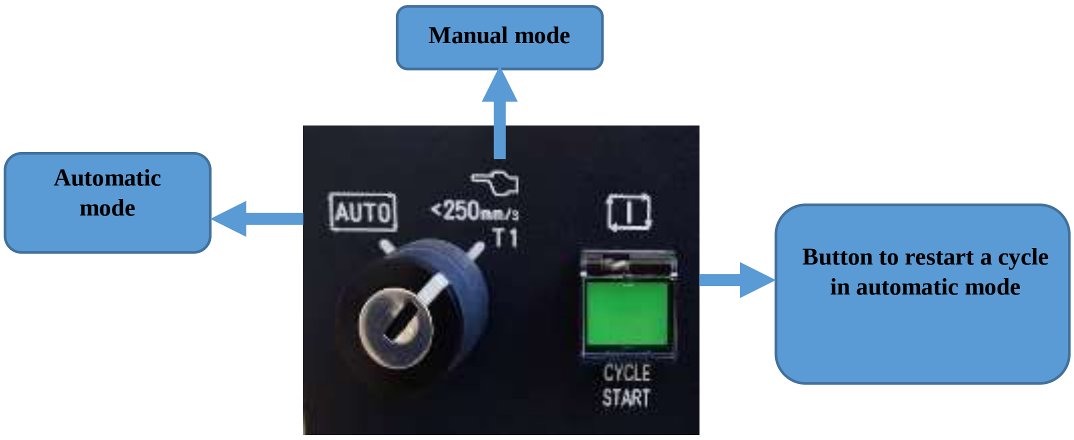
Source: Elaborated by the work team

1.2.2.2 Mode selector switch

This controller model contains two modes of operation for the robot, automatic and manual, depending on the user's selection, the mode is chosen through a master key (see figure 1.5). The difference between each mode is as follows:

- **Manual Mode:** This mode is better considered as programming mode, here the user has complete control through the teach pendant to program the robot. The movements do not exceed 250 mm/s and to move it, the user's intervention is necessary, in addition to the activation of the deadman switch.
- **Automatic Mode:** This mode is better known as production mode, in it, the speed at which the robot works is 100%. Here the activation of the deadman switch is not needed, however, to move the robot, the user intervenes by means of a cycle button as many times as required.

Figure 5.5 Mode selector switch for controller R-30iB Plus



Source: Elaborated by the work team

To run a program using automatic mode it is necessary to configure the system by following the steps below:

1) The TP and the controller are configured in automatic mode.
a. The automatic mode of the controller is selected by rotating the mode selection switch.
b. On the TP, press: Menu → Next → System →Config → Option Local / Remote = Local.
2) Then press FCNT → Abort All → Select → and choose the program to execute.
3) Finally, the CYCLE START button on the controller is pressed to execute.

1.2.3 Teach pendant control unit (TP)

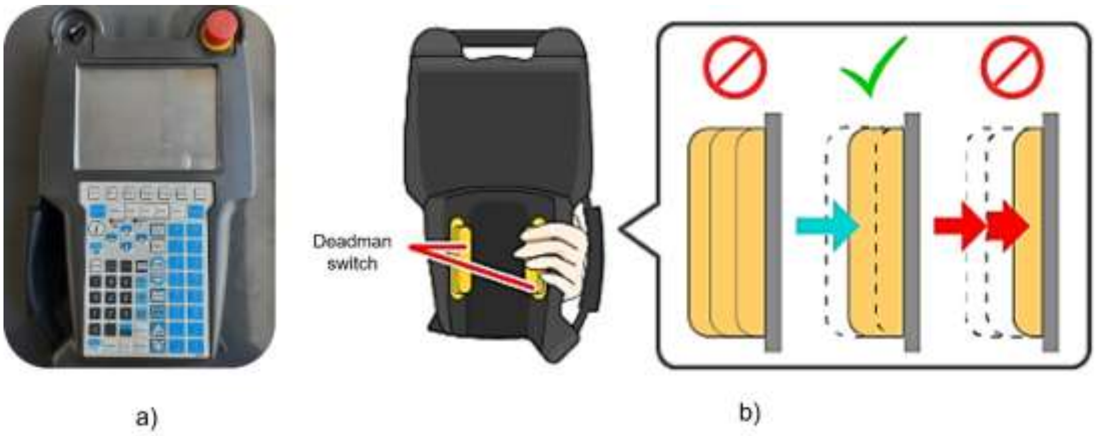
The iPendant, teach pendant, TP, teaching terminal or just teach (see figure 1.6a), is a device used to give control of the robot movement to the user, this from the manual mode in the controller. This control goes from moving the robot, to creating, executing, and modifying programs, etc. (Mikhail & Wang, 2018) Among the many options that the TP offers us, this teach model contains a touch screen, as well as a monitor that allows us to see our robot in 3D simultaneously with the actions we carry out, among hundreds of other functions, making it easy to interact with the operator, both to understand the system, as well as to use it. This is because it contains an intuitive and friendly system.

1.2.3.1 Deadman Switch

To guarantee the user's safety, the teach pendant has two switches located on the back, which interrupt the movements or programs that are being executed in the robot during emergency situations, activating some brakes and temporarily disabling the servomotors (Guerrero Ávila & Loyal Flame, 2017). To enable the robot again (see figure 1.6b), one of the two switches must be activated, understanding the following conditions:

1. Loosely – Robot does not move
2. The first step is pressed – the robot moves
3. It is pressed until the second step – The robot does not move

Figure 6.6 a) Teach Pendant, b) Deadman Switch

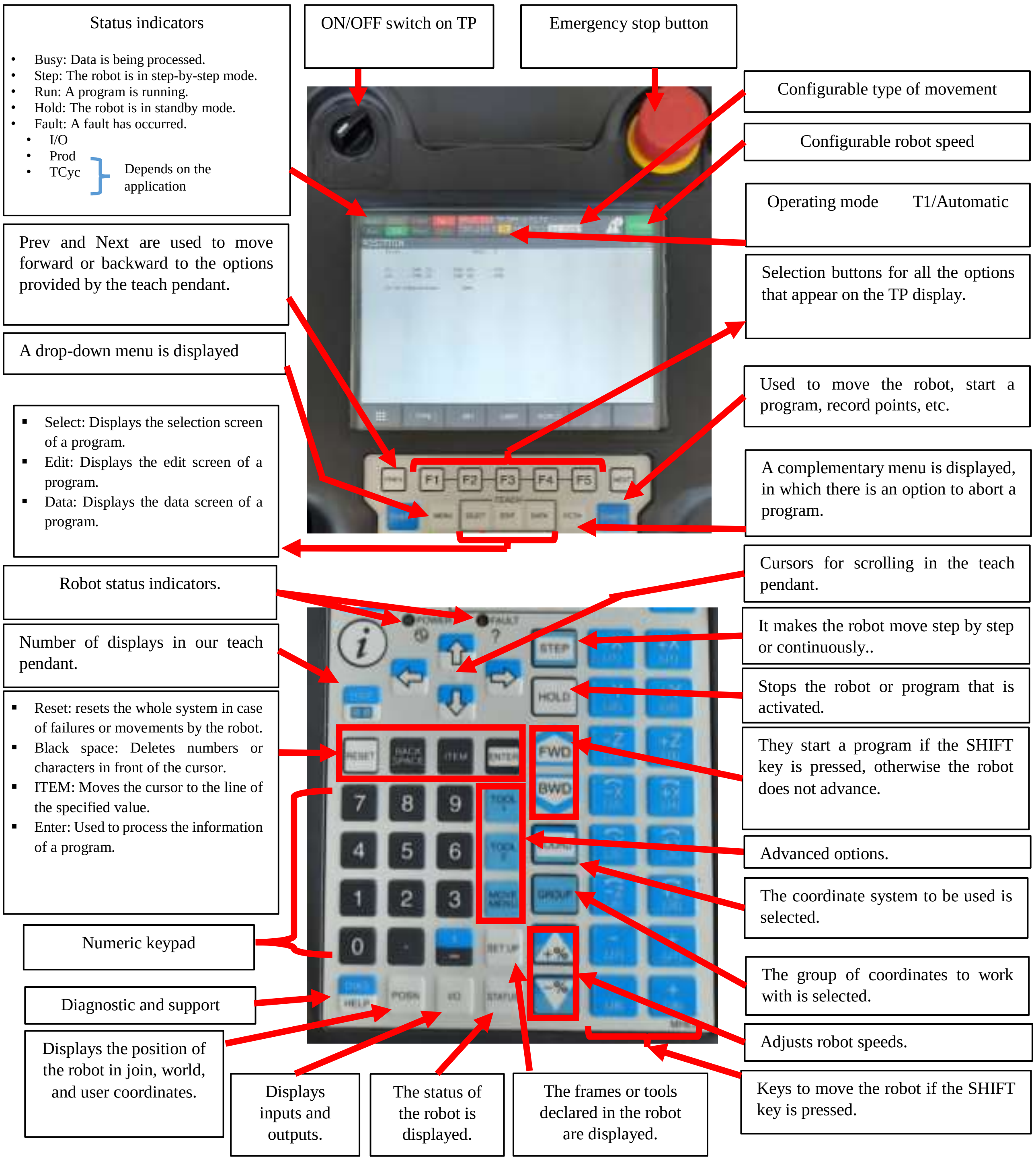


Source: Elaborated by the work team

1.2.3.2 TP buttons and indicators

To use the teach pendant more easily, it is necessary to know the keys, indicators and elements that make it up, since they are essential for programming, editing, diagnosing, etc. For this reason, figure 1.7 presents the basic characteristics of the teach pendant.

Figure 7.7 TP buttons and indicators



Status indicators

- Busy: Data is being processed.
- Step: The robot is in step-by-step mode.
- Run: A program is running.
- Hold: The robot is in standby mode.
- Fault: A fault has occurred.
  - I/O
  - Prod
  - TCyc } Depends on the application

Prev and Next are used to move forward or backward to the options provided by the teach pendant.

A drop-down menu is displayed

- Select: Displays the selection screen of a program.
- Edit: Displays the edit screen of a program.
- Data: Displays the data screen of a program.

Robot status indicators.

Number of displays in our teach pendant.

- Reset: resets the whole system in case of failures or movements by the robot.
- Black space: Deletes numbers or characters in front of the cursor.
- ITEM: Moves the cursor to the line of the specified value.
- Enter: Used to process the information of a program.

Numeric keypad

Diagnostic and support

Displays the position of the robot in join, world, and user coordinates.

ON/OFF switch on TP

Emergency stop button

Configurable type of movement

Configurable robot speed

Operating mode T1/Automatic

Selection buttons for all the options that appear on the TP display.

Used to move the robot, start a program, record points, etc.

A complementary menu is displayed, in which there is an option to abort a program.

Cursors for scrolling in the teach pendant.

It makes the robot move step by step or continuously..

Stops the robot or program that is activated.

They start a program if the SHIFT key is pressed, otherwise the robot does not advance.

Advanced options.

The coordinate system to be used is selected.

The group of coordinates to work with is selected.

Adjusts robot speeds.

Keys to move the robot if the SHIFT key is pressed.

Displays inputs and outputs.

The status of the robot is displayed.

The frames or tools declared in the robot are displayed.

Source: Elaborated by the work team

References

Fanuc. (2018a). *Robot industrial FANUC LRMate 200iD*. https://www.fanuc.eu/es/es/robots/página-filtro-robots/serie-lrmate/lrmate-200-id

Fanuc. (2018b). *Sistema de control de robot FANUC R-30iB Plus*. https://www.fanuc.eu/es/es/robots/accesorios/robot-controller-and-connectivity

FANUC América Corporation. (2016). *Serie LR Mate _ Robots ligeros y compactos _ FANUC América*. https://www.fanucamerica.com/products/robots/series/lr-mate

Guarnizo Marin, J. G., Bautista Díaz, D., & Sierra Torres, J. S. (2021). Una revisión sobre la evolución de la robótica móvil. *Una Revisión Sobre La Evolución de La Robótica Móvil*, 1–8. https://doi.org/10.15332/dt.inv.2021.02848

Guerrero Ávila, Y., & Llama Leal, M. A. (2017). Contribución al Estudio de la Selección de Parámetros de la Estación de Soldadura Robotizada Fanuc 100iC. *Tecnológico Nacional de México*. https://rinacional.tecnm.mx/handle/TecNM/610

Mikhail, M., & Wang, S. (2018). Lab design of FANUC robot operation for engineering technology major students. *ASEE Annual Conference and Exposition, Conference Proceedings*, *2018-June*(May 2001). https://doi.org/10.18260/1-2--30747

Salazar García, I. (2018). Los robots y la Inteligencia Artificial . Nuevos retos del periodismo. *Universidad CEU San Pablo, Madrid, España*, 295–315. http://hdl.handle.net/10637/9889

Fanuc. (2018). *Robot industrial FANUC LRMate 200iD*. https://www.fanuc.eu/es/es/robots/página-filtro-robots/serie-lrmate/lrmate-200-id

FANUC América Corporation. (2016). *Serie LR Mate _ Robots ligeros y compactos _ FANUC América*. https://www.fanucamerica.com/products/robots/series/lr-mate

Chapter 2 Commissioning and preparation for programming

Abstract

Commissioning, in a simple way, can be defined as the processes used to verify that a design, assembly or mechanism is being executed properly. Therefore, from the perspective of an industrial robot, it is important to maintain a knowledge in this regard, since, taking into account the multiple applications that can be performed with them, it is necessary to develop a preparation so that the programs or executions of movements work correctly. This, considering that the necessary knowledge to manipulate these movements in the robot is already available. In this chapter emphasis will be given to the description of the start-up for the Fanuc LR Mate 200iD robot, defining elementary characteristics to start the robotic system and how to manipulate movements. In the same way it is sought to describe the elementary topics to prepare the robot before performing a program.

Start-up, preparation

2.1 Introduction

The FANUC LR Mate series industrial robots have several robot models where some of them contain ISO class 4 standards for food grade and clean rooms (FANUC America Corporation, 2016). That is why, once the composition of the robot is understood, one of the main challenges to manipulate them is to determine how it is programmed and before that, how the movements of the robot are controlled, considering the most important characteristics such as its startup. In addition, due to the increase in applications that can be carried out with industrial robots, at present, to carry out a program, it is necessary to have a preparation that involves the declaration of different elements, such as tools, work areas, etc. For this reason, different companies have developed operating manuals that implement the aforementioned to have a more constant learning.

The objective of this chapter is to describe how the Fanuc LR Mate 200iD robot is started up and, likewise, what is the preparation that it must have before creating a program and what are some of the characteristics to consider. All this in an explanatory way so that it can be replicated by other people.

2.1 Robot start-up

The startup of the robot in simple words is the activation of its system by allowing the passage of electric current, as well as the deactivation of the brake mechanisms among other things.

2.2.1 Startup and shutdown

– **Robot startup**

To turn on the robot in a simple way, assuming that there is no other problem to solve before, it is enough to follow the following procedure:

1) The robotic system is connected to the electrical current and in case of having alternative disconnection elements such as safety switches, it must be disabled to allow the passage of energy to the R-30iB controller.

2) Then, turn the ON/OFF switch knob on the controller to the right, in such a way that you will hear how it turns on its fan, activating the entire system.

3) Subsequently, the emergency stop in the controller is unlocked and the mode in which it will work (automatic or manual) is selected.

4) Finally, to use the robot, unlock the emergency stop and turn the switch on the teach pendant to its ON state.

Note: Depending on the case, operation or action to be performed, there are also other types of start-up, such as start cold, start hot and controlled start

– **Robot shutdown**

1) To turn off the robotic system, you have to perform the same steps as when turning it on, but this time in reverse, taking into account the following considerations:

a. If a program is running, the HOLD key on the TP must be pressed to stop it.

b. The robot must be placed in a position that does not affect its condition in the future or avoiding possible collisions.

c. The TP switch must be turned to its OFF state and the emergency stop button must be pressed to prevent any instruction from being activated or executed.

Note: Before turning the robot on or off, make sure that the cell conditions are in good condition and that proper safety measures are being taken.

2.2.2 Start Hot

This type of startup is one of the most common ways to give power to the robot, here the robot is already on, but it seeks to save or initialize variables contained in the system again, so when this startup is activated, only the robot is restarted. internal system of the robot and in the same way, the teach pendant remote-control unit is restarted.

2.2.3 Start Cold

A cold boot is done for different reasons: the robot is behaving strangely, a backup is being made, some variables are being changed, etc. Whatever the case, this startup consists of quickly turning the robot off and on by turning the main knob on the controller to turn it off, waiting about 30 seconds, and turning the knob again to turn the robot back on. This makes it work as if it were a CPU restart, initializing the system variables and saving any changes that may be made.

2.2.4 Controlled start

Controlled start is a type of start that allows us to display additional options that are not available when the robot is running normally. In addition, it allows us to unlock variables, restore a backup, among other things. To obtain a controlled start, it is enough to have the TP activated in its ON state and press and hold PREV+NEXT while the controller is turned on, when the options menu comes out we choose controlled start, and you can select what you want.

2.3 Joggin

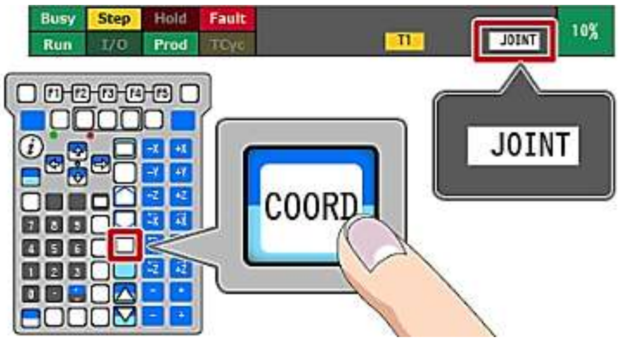
Having as one of the important points knowing how to perform movements with the LR Mate 200iD robot from the teach pendant, the following steps can be followed:

1) Once the controller has been enabled, the TP is now enabled by turning the switch to its ON state and the emergency stop is unlocked in the same way.

2) Subsequently, the type of coordinates to which the robot will move is chosen, by pressing the COORD key repeatedly until reaching the appropriate system, following the pattern in figure 2.1:

Figure 2.1 Selection button for the type of coordinate system to be used

JOINT --> JGFRM --> WORLD --> TOOL --> USER --> JOINT



Source: Modified image obtained from ROBOGUIDE software

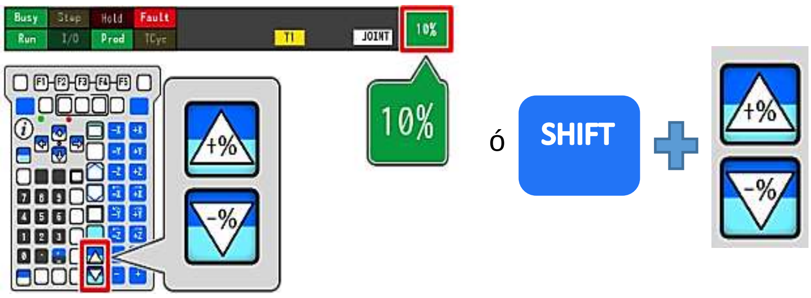
3) The speed at which the robot will operate is selected, taking into account the speed values that we are given to choose from. It should be noted that when pressing any of the two keys in figure 2.2, the percentage varies in a different way, pressing the simple keys, the percentage goes from 1 to 1% and after 10% it goes from 5 to 5%, if you press the combination of the speed and shift keys at the same time, the options will be handled from 50 to 50%. In addition, there are fine speeds that, depending on the type of movement, is the speed at which they are manipulated (see table 2.1).

Table 2.1 Achievable speeds

Speed Values	Vfine	Fine	Continuous Motion (1,2,3,4,5,50,100%)
Joint	Approx. 0,0001 degrees	Approx. 0,001 degrees	Speed %
World	Approx. 0,02 mm	Approx. 0,23 mm	Speed %

Source: (FANUC Robotics Ibérica S. L., 2018)

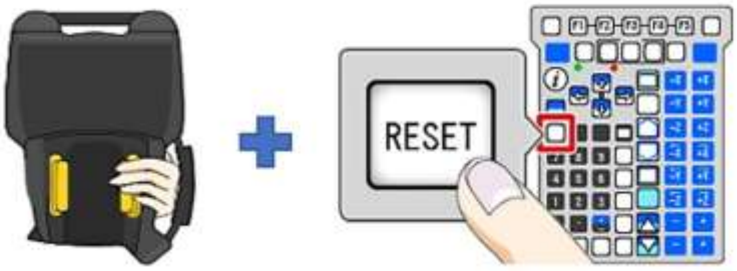
Figure 2.2 Keys for changing the speed of movement



Source: Modified image obtained from ROBOGUIDE software.

4) The deadman switch is pressed until the first step to activate the teach pendant and the RESET key is pressed instantly (see figure 2.3), this is so that the servomotors are turned on and the robot brakes are unlocked. We will realize that the above has happened, since a deactivation sound can be clearly heard.

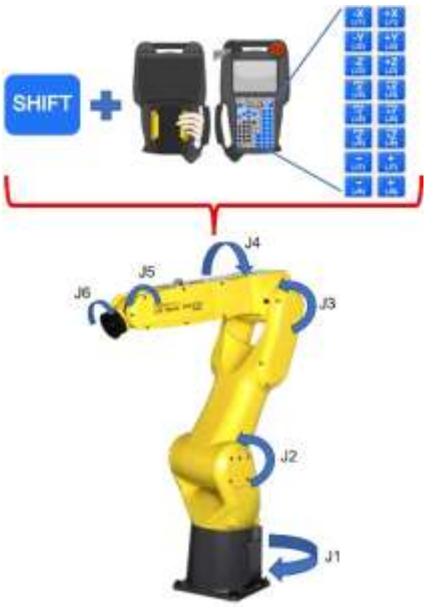
Figure 2.3 Key combination to deactivate the brakes of the robot.



Source: Modified image obtained from ROBOGUIDE software.

5) Finally, and with the deadman switch still activated, you have to press the shift key and then the key that corresponds to the movement that we want the robot to perform (in this case, the robot being in a Joint coordinate system), as like figure 2.4.

Figure 2.4 Use of the teach pendant for robot movements.



Source: Modified image obtained from ROBOGUIDE software and (Fanuc, 2018)

2.4 Movement of the robot according to the selected coordinate system

Coordinate systems are implemented to the robot to indicate the space, position or orientation in which it is located (Pinto Hernández, 2016). For this robot model, the systems are subdivided into 5: WORLD, TOOL, JOINT, USER AND JOGFRM. Being the first three, the most used within the industry.

2.4.1 Joint Coordinates

This coordinate system performs movements that are measured in degrees, here the robot moves each axis individually depending on a positive or negative orientation (Ponsa Asensio, 2003). The movement keys are defined by ±J1 to ±J6 respectively starting from the base (an example of this can be seen in figure 2.5A).

2.4.2 World Coordinates

This is a static, three-dimensional Cartesian coordinate system. Its origin is in the base of the robot and it is defined at the factory (Ponsa Asensio, 2003). In addition, the origin will almost always be oriented taking into account the robot's power connection, as can be seen in figure 2.5B.

2.4.3 Tool Coordinates

The origin of this type of Cartesian coordinates is defined by the user and is always located at the flange (or end effector) of the robot (see in figure 2.5E). However, the origin is reprogrammable, that is, we can define a new origin by declaring the dimensions of the tool to be used. This is known as TCP (this topic is explored in more detail in this chapter). (Ponsa Asensio, 2003). This system is mobile for each of the robot movements and this specific model has 10 programmable TOOL systems, so there is a wide range of choice to use any tool.

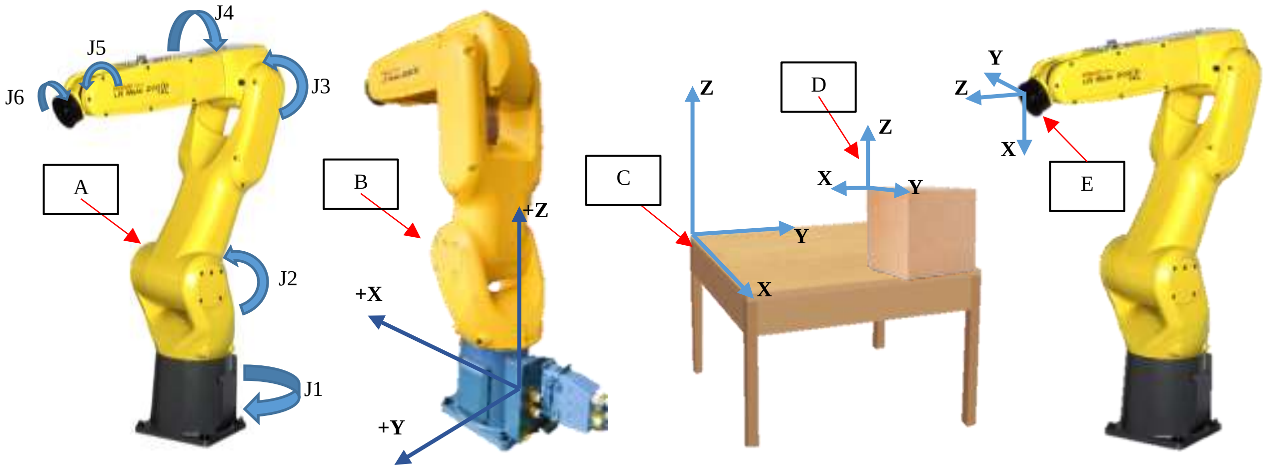
2.4.4 User frame and Jog frame coordinates

Both are Cartesian coordinate systems defined by the user and unlike the Tool coordinates, these are coordinates that, once the origin is declared, remain static. Its objective is to move the robot linearly and efficiently with respect to a work area. The difference between them is that JOGFRM coordinates have no special meaning, however, while USER coordinates focus on entire work areas (see in figure 2.5C), JOGFRM coordinates are used for work objects as in the figure 2.5D (Ponsa Asensio, 2003).

2.4.5 Checking the position of the robot from the TP

To see the position of the robot in real time through the teach pendant, you can press the POSN key or you can follow the address MENU → 0-NEXT → 4-POSITION. In any of the cases, we will obtain a screen that will show us the position of the robot, at this point we can choose the type of coordinates that we want to see. Also, the options available to you are: JOINT, USER, and WORD.

Figure 2.5 Types of coordinate systems on the robot A)Joint, B)World, C)User, D)Jogfrm



Source: Modified image obtained from (Fanuc, 2018)

2.5 Tool declaration

The preparation of a robotic system for later programming is equivalent to having to declare the elements with which it will work, where the tools, plans or work areas enter, among other characteristics. Being the focus on the declaration of tools, there are essential definitions to understand, one of them is the well-known "TCP", which will be covered below.

2.5.1 TCP

The center point of the tool or better known as TCP, is the point with respect to which all the positions of the robot are defined. In other words, it is a way of telling the robot where the main part of the tool to use is, so that distance precautions are taken into account, etc. Normally, when there is no tool on the robot, the TCP will always be in the center of the end effector (see figure 2.6A).

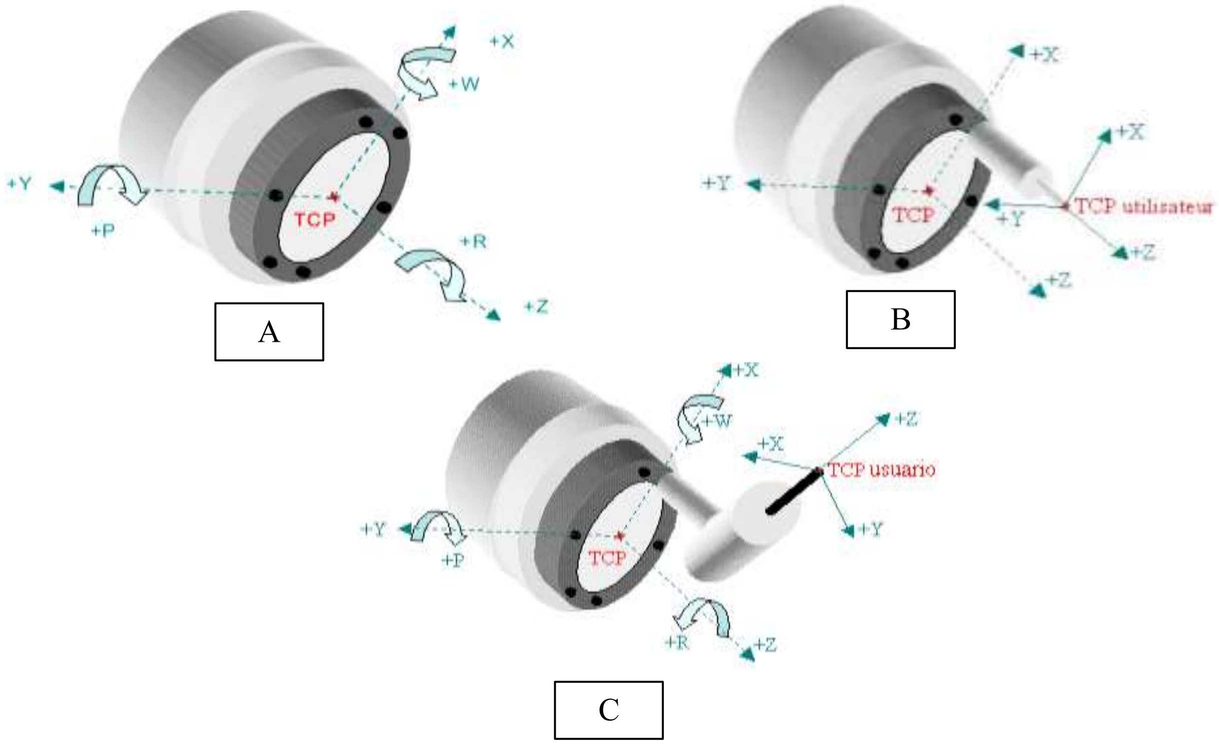
2.5.2 Simple TCP

A simple tool is a tool that contains an attack axis parallel to the z axis of the flange (Navarro Sanz, 2009). In this case, the orientation of the tool does not change, it only moves (see figure 2.6B), which is why it is advisable to use the three-point method to declare this type of tool.

2.5.3 Complex TCP

A complex tool contains an axis of attack that is not parallel to the z axis of the flange, since the TCP is displaced and its orientation is redefined, as can be seen in figure 2.6C (Navarro Sanz, 2009). For this type of cases, it is advisable to use the 6-point method to declare the tool.

Figure 2.6 TCP types A)Standard TCP, B) Simple TCP, C)Complex TCP



Source: Modified image obtained from (Sanz, 2009)

2.5.4 Declare a tool

To declare a tool we first have to know what a FRAME is. The frames are the boxes where the coordinate systems that the robot uses to locate itself in a workspace are defined. In other words, the robot has a list of frames where all the tools declared in the system are shown, this specific robot allows up to 10 tools to be declared. To get to that list, follow the address:

MENU → SET UP → F1: [TYPE] → 4-FRAMES → F3:|OTHER|→ 1-TOOL FRAME

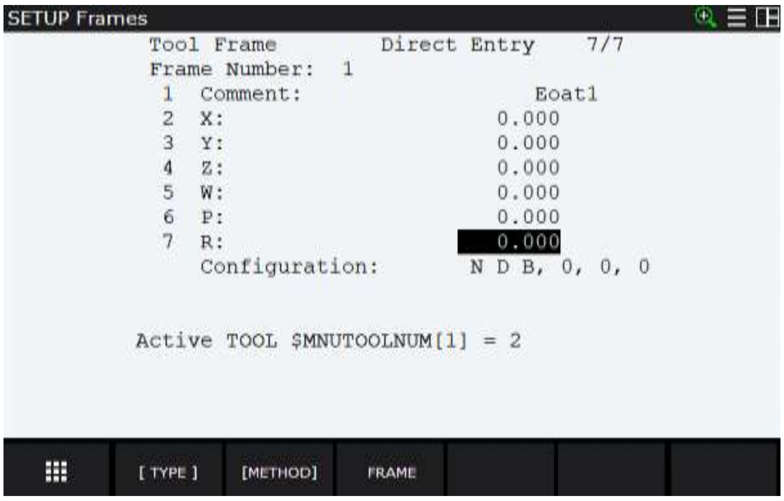
When found there, you can see 10 boxes which are made up of a number, the location coordinates and a name. To declare the tool we want, we have to position the cursor of the teach pendant in a box and press DETAIL, then choose a method shown below to declare the origin of that tool.

2.5.4.1 Direct entry method

This method is used when the coordinates and specific orientation of the tool are already perfectly known (Navarro Sanz, 2009), thanks to the fact that this method allows these data to be placed manually without the need to move the robot (see figure 2.7). To do so, go to the following address and enter all the position data of the TCP of your tools according to what is requested:

F2: [DETAIL] → F2: [METHOD] → 6-DIRECT ENTRY

Figure 2.7 TP screen using the direct entry method for tools



```
Tool Frame          Direct Entry    7/7
Frame Number:  1
 1  Comment:                    Eoat1
 2  X:                      0.000
 3  Y:                      0.000
 4  Z:                      0.000
 5  W:                      0.000
 6  P:                      0.000
 7  R:                      0.000
    Configuration:         N D B, 0, 0, 0

Active TOOL $MNUTOOLNUM[1] = 2
```

Source: Modified image obtained from ROBOGUIDE software

2.5.4.2 Three points method

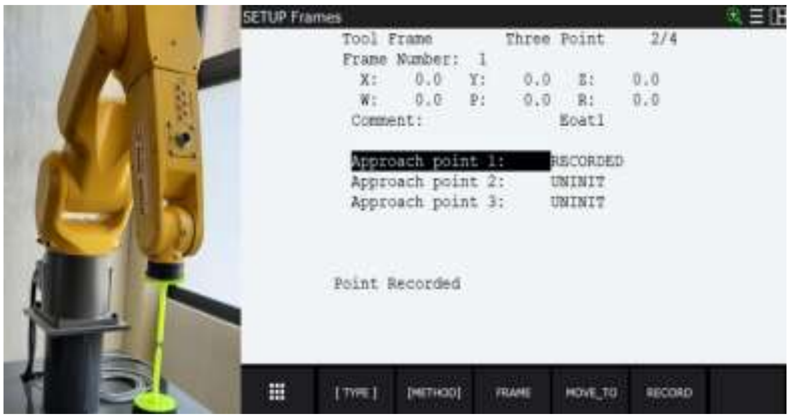
This method consists of recording three points by means of a real reference to create a coordinate system at the point of attack of some tool. To do this, we must move the robot to that reference as close and centered as possible to then record a point there, later it is done twice more, but orienting the robot differently without moving away from the reference, until the UNINIT legends in the teach pendant change to USED. The address to choose this method is: is:

F2: [DETAIL] → F2: [METHOD] → 1-THREE POINTS

Steps to declare a tool by the 3-point method:

1) The first step is to place the robot in any position where the tool is about to touch our reference and that it is as close and centered as possible (see figure 2.8).

Figure 2.8 Step 1 of the three points method for declaring tools



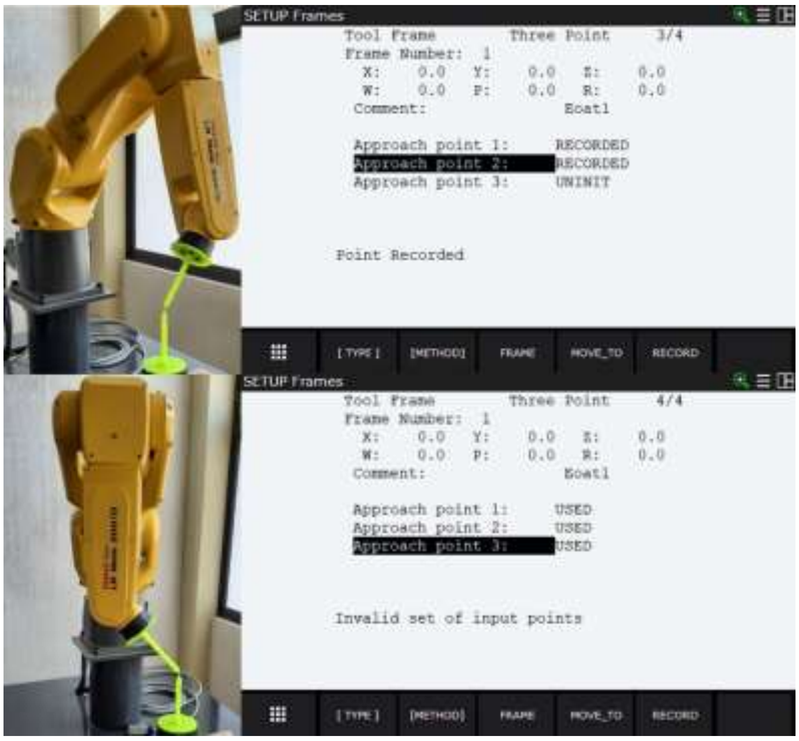
```
Tool Frame          Three Point      2/4
Frame Number:  1
  X:    0.0   Y:    0.0   Z:    0.0
  W:    0.0   P:    0.0   R:    0.0
 Comment:                Eoat1

 Approach point 1:       RECORDED
 Approach point 2:       UNINIT
 Approach point 3:       UNINIT


Point Recorded
```

Source: Elaborated by the work team

2) After that we position the Teach Pendant cursor at APPROACH POINT 1 and record that point. To record that position just press the key combination SHITF + RECORD at the same time. To know if what we did before was correct, the legend RECORDED will have to appear on the screen instead of UNINIT (see figure 2.8).

3) To conclude with this method, the following is to repeat step 1 and 2 two more times, positioning our robot in different postures and selecting the point to record with the cursor dates in the TP. When we are finished we can see that the legends RECORDED will change to USED, in that case we will have finished, and our tool will be ready to use (see figure 2.9).

Figure 2.9 Step 2 and 3 of the three points method for declaring tools



```
Tool Frame          Three Point      3/4
Frame Number:  1
  X:     0.0   Y:     0.0   Z:     0.0
  W:     0.0   P:     0.0   R:     0.0
 Comment:                  Eoat1

 Approach point 1:      RECORDED
 Approach point 2:      RECORDED
 Approach point 3:      UNINIT


Point Recorded
```

```
Tool Frame          Three Point      4/4
Frame Number:  1
  X:     0.0   Y:     0.0   Z:     0.0
  W:     0.0   P:     0.0   R:     0.0
 Comment:                  Eoat1

 Approach point 1:      USED
 Approach point 2:      USED
 Approach point 3:      USED


Invalid set of input points
```

Source: Elaborated by the work team

Nota: To position a robot to a point that we recorded earlier because we accidentally moved the robot or for some other reason, you have to position the TP cursor to the point to which we want to move it and press the key combination SHIFT + MOVE_TO

2.5.4.3 Six point method

This method is practically the same as the 3-point method, with the difference that we place an additional point that gives the origin to our coordinate system, and two more points to define the orientation that this system will have, it can be through XY or XZ. The address to choose this method is:

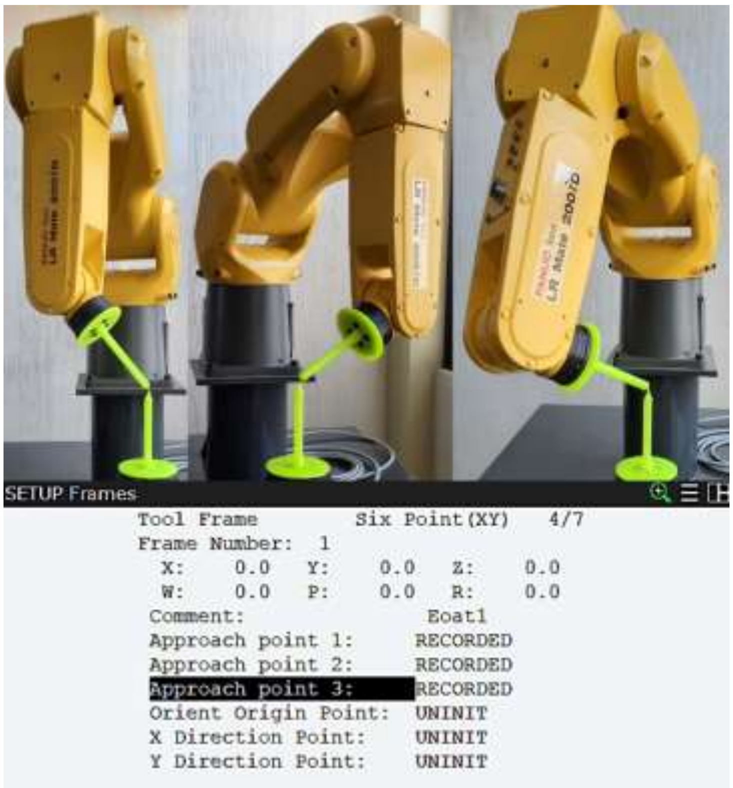
F2: [DETAIL] → F2: [METHOD] → 2/3 SIX POINT (XY o XZ)

Steps to declare a tool by the 6-point method:

1) At first we will have to record the first three reference points as in the previous method, again considering that the robot postures are different (see in figure 2.10).

2) Subsequently, we have to record a point to which it is going to be oriented to the origin, that is, we must consider what is the coordinate system that we want to give to the tool so that this new point is referenced to it (see figure 2.11).

Figure 2.10 Step 1 of the six point method for declaring tools

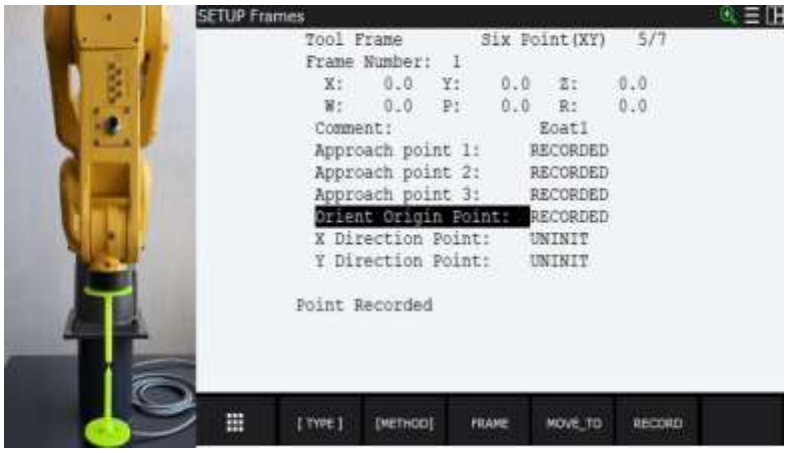


```
Tool Frame          Six Point(XY)    4/7
Frame Number:  1
  X:     0.0   Y:     0.0   Z:     0.0
  W:     0.0   P:     0.0   R:     0.0
 Comment:                Eoat1
 Approach point 1:      RECORDED
 Approach point 2:      RECORDED
 Approach point 3:      RECORDED
 Orient Origin Point:   UNINIT
 X Direction Point:     UNINIT
 Y Direction Point:     UNINIT
```

Source: Elaborated by the work team

3) Now, depending on the method you chose (XY or XZ), you have to move the robot first to where we want our X axis and record that point. Then, we return to ORIENT ORIGIN POINT and press SHITF + MOVE_TO at the same time, to move the robot to the position that we are indicating. Then the robot moves again, but now oriented to the "Z" or "Y" axis that we want to declare and we also save it (see figure 2.12).

Figure 2.11 Step 2 of the six-point method for declaring tools

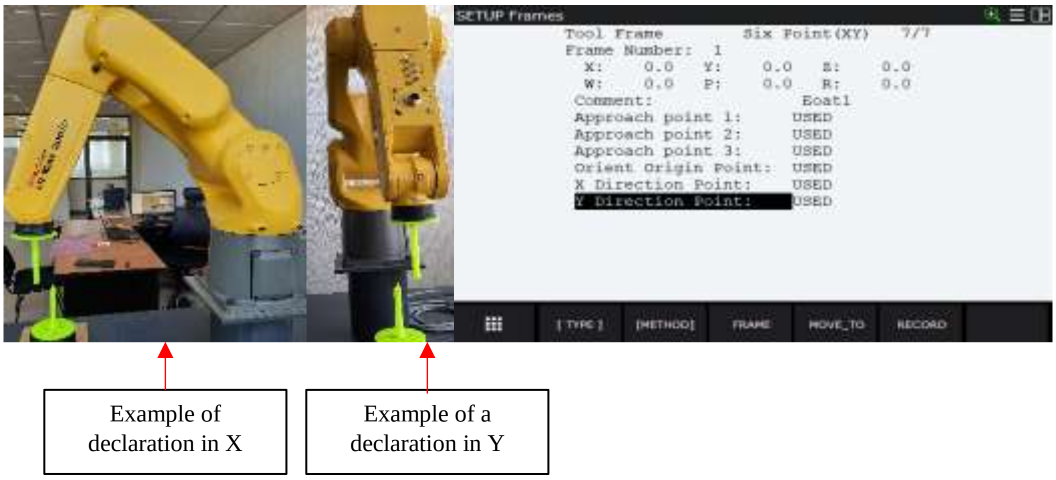


```
Tool Frame          Six Point(XY)    5/7
Frame Number:  1
  X:     0.0   Y:     0.0   Z:     0.0
  W:     0.0   P:     0.0   R:     0.0
 Comment:                Eoat1
 Approach point 1:      RECORDED
 Approach point 2:      RECORDED
 Approach point 3:      RECORDED
 Orient Origin Point:   RECORDED
 X Direction Point:     UNINIT
 Y Direction Point:     UNINIT

Point Recorded
```

Source: Elaborated by the work team

4) Finally, we will see that the RECORDED legends will change to USED, and the tool will be declared and ready to be used. The most common way to check how accurate the new system is, is to activate the tool by changing the robot coordinates to Tool coordinates and pressing only the rotation keys from the Teach Pendant to make it move on that system. If when moving the robot, the tool is in the same starting position, the system has a high accuracy.

Figure 2.12 Step 3 of the six-point method for declaring tools



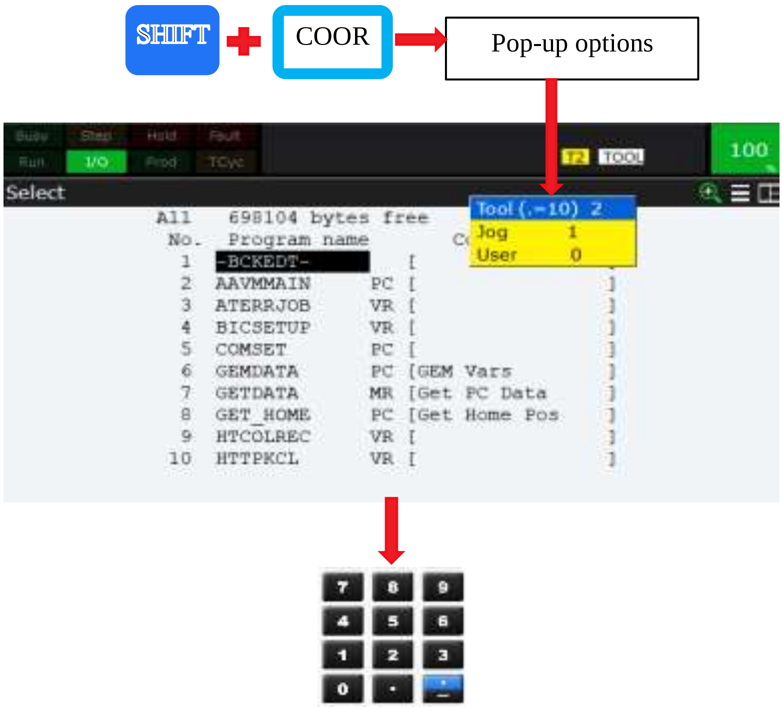
```
SETUP Frames
        Tool Frame           Six Point(XY)    7/7
        Frame Number:   1
          X:     0.0    Y:     0.0    Z:     0.0
          W:     0.0    P:     0.0    R:     0.0
         Comment:                   Eoat1
         Approach point 1:         USED
         Approach point 2:         USED
         Approach point 3:         USED
         Orient Origin Point:      USED
         X Direction Point:        USED
         Y Direction Point:        USED
```

Source: Elaborated by the work team

2.5.5 Activation of a declared tool

When a tool has been declared, it can be selected from the Teach Pendant by pressing the SHIFT + COORD key at the same time, when a pop-up tab appears, place the number of the tool that was declared from the beginning, press ENTER, and it will be ready (see figure 2.13). Another way to do this is to press the F5: SENTID key in the SETUP Frames window, enter the tool number and press ENTER.

Figure 2.13 Activation of declared tools



```
Select
      All    698104 bytes free          Tool (.-10)  2
      No.   Program name            Co  Jog          1
       1   -BCKEDT-          [          User         0
       2   AAVMMAIN      PC  [                       ]
       3   ATERRJOB      VR  [                       ]
       4   BICSETUP      VR  [                       ]
       5   COMSET        PC  [                       ]
       6   GEMDATA       PC  [GEM Vars               ]
       7   GETDATA       MR  [Get PC Data            ]
       8   GET_HOME      PC  [Get Home Pos           ]
       9   HTCOLREC      VR  [                       ]
      10   HTTPKCL       VR  [                       ]
```

Source: Elaborated by the work team

2.6 Declaration of additional reference systems (User frame)

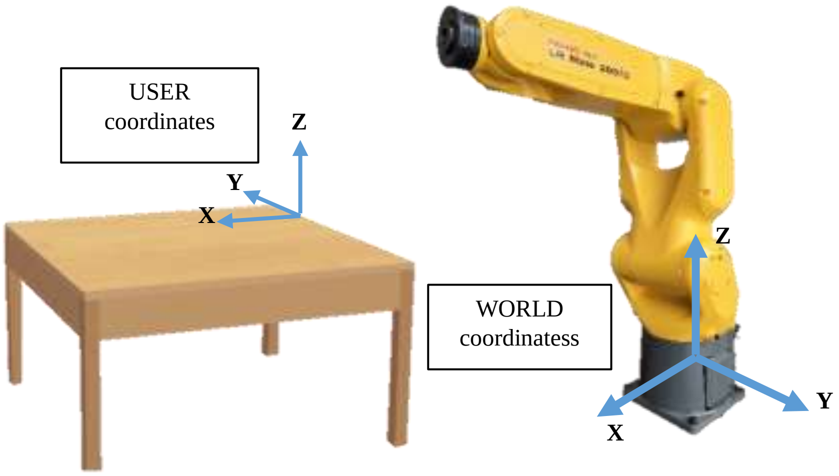
2.6.1 User frame coordinate systems

The user frame is a three-dimensional, cartesian reference system where the user can configure its orientation and position anywhere in space within reach of the robot (see figure 2.14). Same in which, when performing a TP program, all the positions that are recorded will move and take the configuration of this system as a reference and can only be changed internally (FANUC Robotics Iberic S. L., 2018).

To declare a User frame it is necessary to move and orient our TCP to the base of the system we want, in this way we can move the robot, as well as record positions that are guided based on the reference system. Also, only one user frame can be selected at a time, or it can be called programmatically. If no user frame is configured, WORLD coordinates will be taken by default. There are 3 methods to declare a User frame:

- 3 points method
- 4 points method
- Incoming address method

Figure 2.14 Difference between USER and WORLD coordinates



Source: Elaborated by the work team

2.6.2 Declare a USER frame

The robot has a list of frames where all the Users declared in the system are shown, specifically, this version of the robot and controller allows up to 9 user systems to be declared. To get to that list, follow the address:

MENU → SET UP → F1: [TYPE] → 4-FRAMES → F3:|OTHER|→ 3-USER FRAME

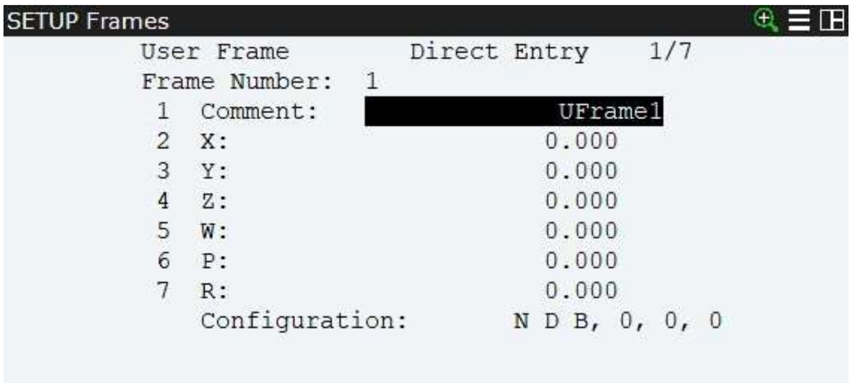
Being there, you can see 9 boxes which are made up of a number, the location coordinates and a name. To declare the user systems that we need, we have to position the cursor of the teach pendant in some box and press DETAIL, later the name of the system can be configured, and a method shown below is chosen.

2.6.2.1 Direct entry method

This method is used when the coordinates and specific orientation of the new reference system are already perfectly known, thanks to the fact that this method allows these data to be placed manually without the need to move the robot (see figure 2.15). To do so, go to the following address and enter all the data according to what is requested.

F2: [DETAIL] → F2: [METHOD] →3-DIRECT ENTRY

Figure 2.15 TP screen using the direct entry method for Users



```
SETUP Frames
          User Frame         Direct Entry    1/7
          Frame Number:  1
           1  Comment:                   UFrame1
           2  X:                           0.000
           3  Y:                           0.000
           4  Z:                           0.000
           5  W:                           0.000
           6  P:                           0.000
           7  R:                           0.000
              Configuration:        N D B, 0, 0, 0
```

Source: Modified image obtained from ROBOGUIDE software

2.6.2.2 Three points method

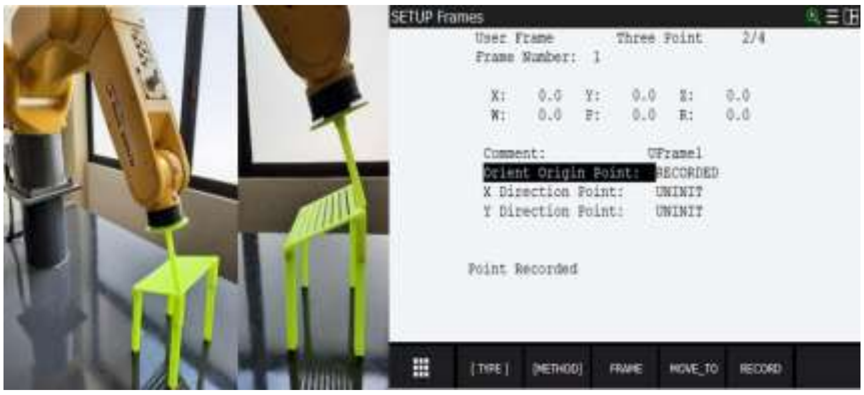
This method consists of recording three points by means of a real reference (it can be the base of the reference system that we want), this to create a coordinate system from which our robot will be guided. To do this, we must move the robot and position it on the reference base as close and centered as possible and then record a point there, to finish this step is done twice more, but determining the orientation of the system until the UNINIT legends in the iPendant change to USED.

The address to choose this method is:

F2: [DETAIL] → F2: [METHOD] → 1-THREE POINTS

1) We place the cursor on “Orient Origin Point” and orient our TCP to the place where our new reference system will begin to orient itself, once there, press SHIFT+RECORD to record the position (see figure 2.16). Here you will change the UNINIT legend to RECORDED.

Figure 2.16 Step 1 of the three points method for declaring USER frames

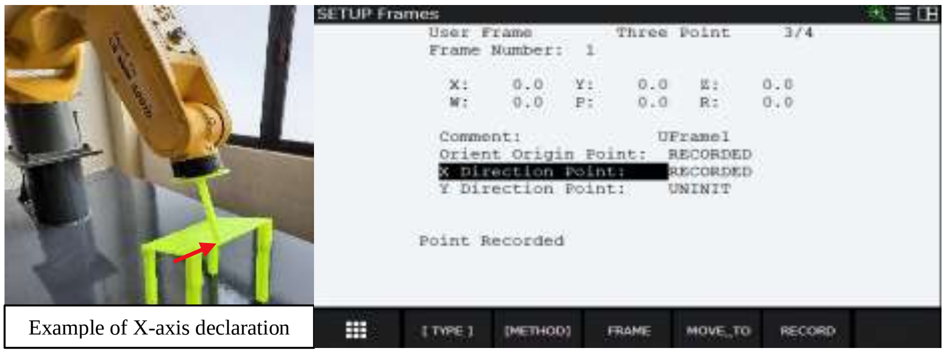


```
SETUP Frames
     User Frame          Three Point     2/4
     Frame Number:  1

       X:    0.0   Y:    0.0   Z:    0.0
       W:    0.0   P:    0.0   R:    0.0

      Comment:                UFrame1
      Orient Origin Point:    RECORDED
      X Direction Point:      UNINIT
      Y Direction Point:      UNINIT

    Point Recorded
```

Source: Elaborated by the work team

2) We place the TP cursor on the “X Direction Point” line and move the robot just a little to indicate the new x axis of the system, once this is done, record the position (see figure 2.17).

Figure 2.17 Step 2 of the three points method for declaring USER frames

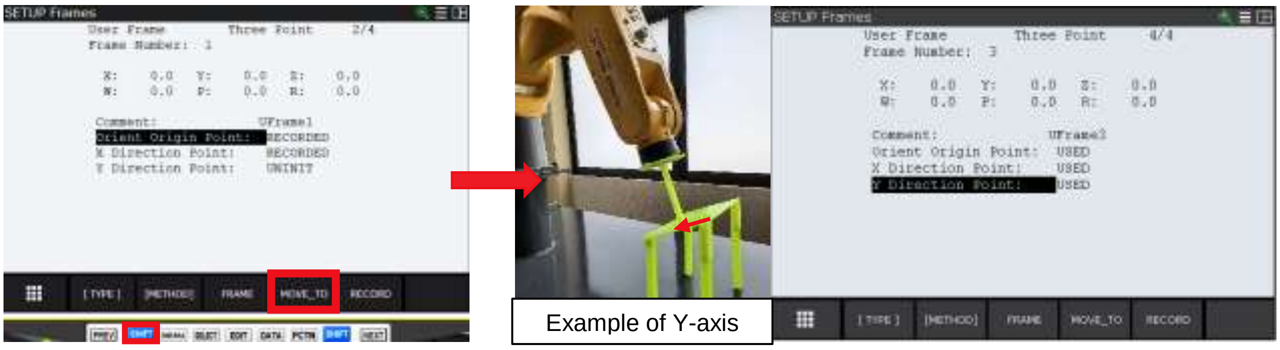


```
SETUP Frames
        User Frame          Three Point      3/4
        Frame Number:   1

          X:     0.0    Y:     0.0    Z:     0.0
          W:     0.0    P:     0.0    R:     0.0

         Comment:                    UFrame1
         Orient Origin Point:   RECORDED
         X Direction Point:     RECORDED
         Y Direction Point:     UNINIT


       Point Recorded
```

Source: Elaborated by the work team

3) Finally, place the cursor on the "Orient Origin Point" line and press SHIFT+MOVE TO to return to the system origin. Then relocate the cursor to "Y Direction Point" and do the same procedure as in step two, defining the Y axis this time (see figure 2.18).

Figure 2.18 Step 3 of the three points method for declaring USER frames



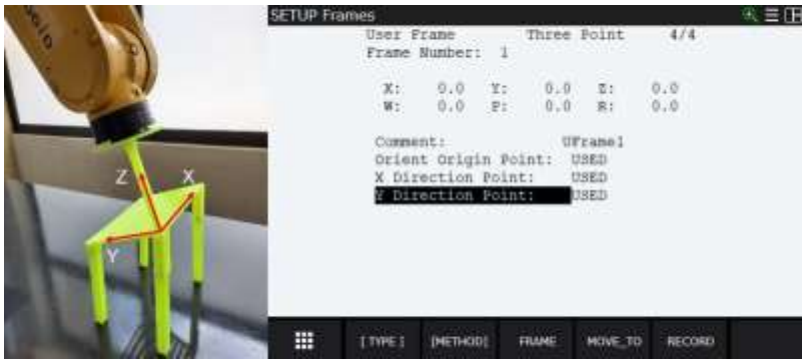
```
SETUP Frames
        User Frame          Three Point      2/4
        Frame Number:   1

          X:     0.0    Y:     0.0    Z:     0.0
          W:     0.0    P:     0.0    R:     0.0

         Comment:                    UFrame1
         Orient Origin Point:   RECORDED
         X Direction Point:     RECORDED
         Y Direction Point:     UNINIT
```

```
SETUP Frames
        User Frame          Three Point      4/4
        Frame Number:   3

          X:     0.0    Y:     0.0    Z:     0.0
          W:     0.0    P:     0.0    R:     0.0

         Comment:                    UFrame3
         Orient Origin Point:   USED
         X Direction Point:     USED
         Y Direction Point:     USED
```

Source: Elaborated by the work team

4) When you are done with the process, the RECORDER lines will change to USED and the new reference frame will be ready to use (see figure 2.19). It is worth mentioning that the final coordinates of the new system can be seen at the top of the User frame.

Figure 2.19 Step 4 of the three points method for declaring USER frames



```
SETUP Frames
        User Frame          Three Point      4/4
        Frame Number:   1

          X:     0.0    Y:     0.0    Z:     0.0
          W:     0.0    P:     0.0    R:     0.0

         Comment:                    UFrame1
         Orient Origin Point:   USED
         X Direction Point:     USED
         Y Direction Point:     USED
```

Source: Elaborated by the work team

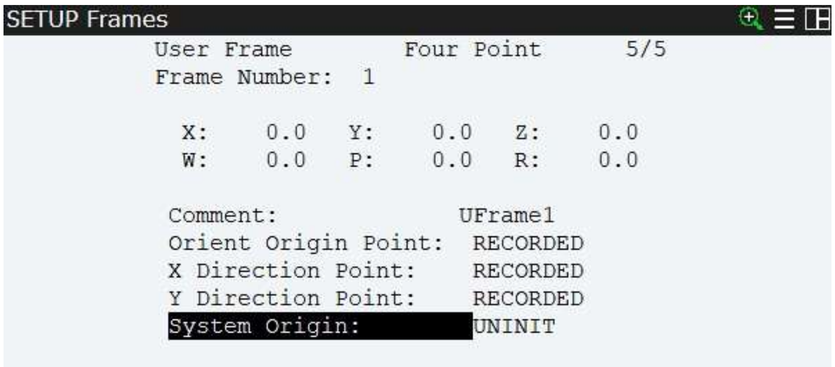
2.6.2.3 Four points method

This method follows the first three steps of the previous method and adds an extra point where we have to define the origin of our system. An example of its use is that the reference system can be oriented in the center of a worktable (for greater comfort) and the fourth point would serve to position the origin of that system in one of its vertices. The address to choose this method is:

F2: [DETAIL] → F2: [METHOD] → 2-FOUR POINTS

1) We follow the first three steps of the 3-point method, until we have the first three lines on the screen, with the legend RECORDER (see figure 2.20)

Figure 2.20 Step 1 of the four points method for declaring USER frames



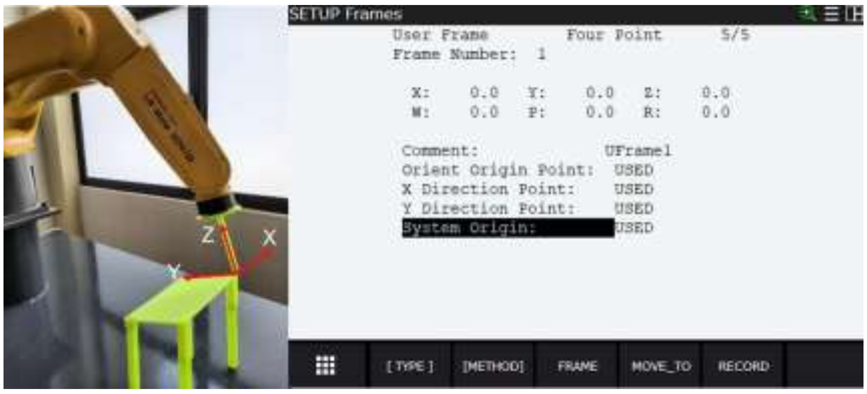
SETUP Frames

```
User Frame          Four Point       5/5
Frame Number:   1

  X:     0.0   Y:     0.0   Z:     0.0
  W:     0.0   P:     0.0   R:     0.0

 Comment:                UFrame1
 Orient Origin Point:    RECORDED
 X Direction Point:      RECORDED
 Y Direction Point:      RECORDED
 System Origin:          UNINIT
```

Source: Modified image obtained from ROBOGUIDE software

2) Finally, and referring to figure 2.19 for the orientation of the system, place the robot at the point where you want the origin of that system and record it (see figure 2.21). This will cause the orientation to move directly to that point and the method lines will change to USED.

Figure 2.21 Step 2 of the four points method for declaring USER frames



SETUP Frames

```
User Frame          Four Point       5/5
Frame Number:   1

  X:     0.0   Y:     0.0   Z:     0.0
  W:     0.0   P:     0.0   R:     0.0

 Comment:                UFrame1
 Orient Origin Point:    USED
 X Direction Point:      USED
 Y Direction Point:      USED
 System Origin:          USED
```

Source: Elaborated by the work team

2.6.3 Activating a declared User frame

When a user frame has been declared, it can be selected from the Teach Pendant by pressing the SHIFT + COORD key at the same time, when a pop-up tab appears, select the USER option with the cursor and place the number of the frame that was declared from the start, press ENTER and you're good to go. Another way to do it is by pressing the F5: SENTID key in the SETUP USER Frames window, enter the reference system number and press ENTER.

2.7 Payload declaration

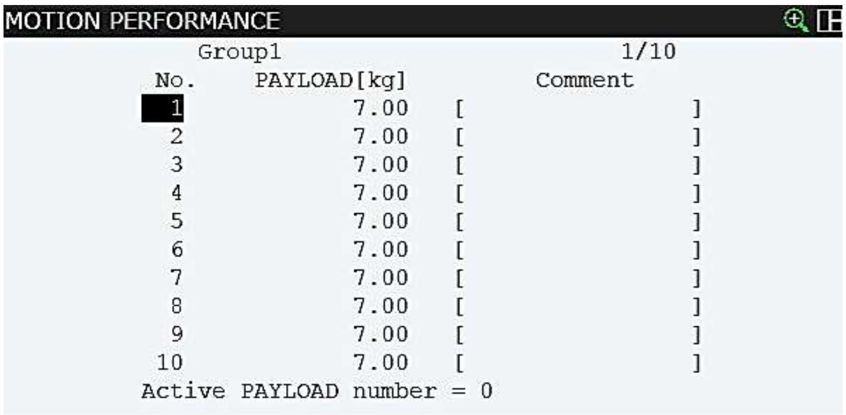
The payload or payload is a way of informing the robot about the weight it is carrying on the last extremity of the arm. This weight depends on the tool to be installed, the workpiece to be moved, etc. This version of the robot contains memory to declare up to 10 payloads and the reasons why they are performed are listed below (FANUC Robotics Ibérica S. L., 2018):

- Improves the performance and accuracy of the robot.
- Collision detection works much better.
- Improve cycle times.
- Reduces vibration.

To get to the main screen of declared payloads, go to the address:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 7-MOTION

Figure 2.22 Main screen of available payloads



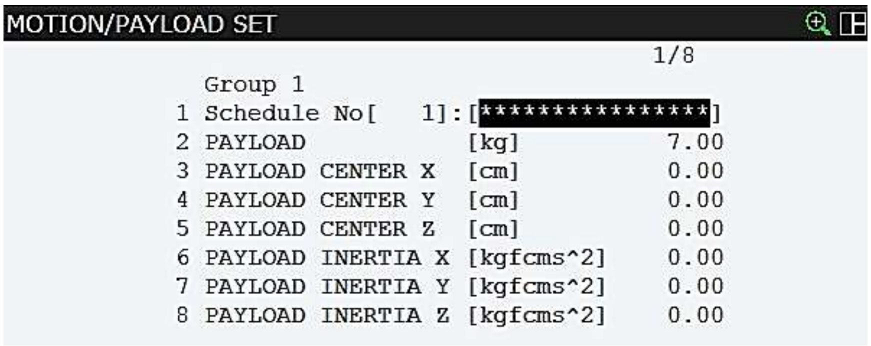
```
MOTION PERFORMANCE
          Group1                     1/10
       No.    PAYLOAD[kg]        Comment
        1           7.00    [                 ]
        2           7.00    [                 ]
        3           7.00    [                 ]
        4           7.00    [                 ]
        5           7.00    [                 ]
        6           7.00    [                 ]
        7           7.00    [                 ]
        8           7.00    [                 ]
        9           7.00    [                 ]
       10           7.00    [                 ]
      Active PAYLOAD number = 0
```

Source: Modified image obtained from ROBOGUIDE software

2.7.1 Manual entry

Following the context of figure 2.22 and depending on the payload you want to register, choose an option by positioning the TP cursor on any available section and press DETAIL. Once inside, we have to enter all the data requested (see figure 2.23), among them are: the weight in Kg, the center of gravity in cm with respect to the original TCP of the robot (the flange), as well as well as the inertia X, Y and Z in Kgfcms^2, as shown below.

Figure 2.23 Internal details of a Payload

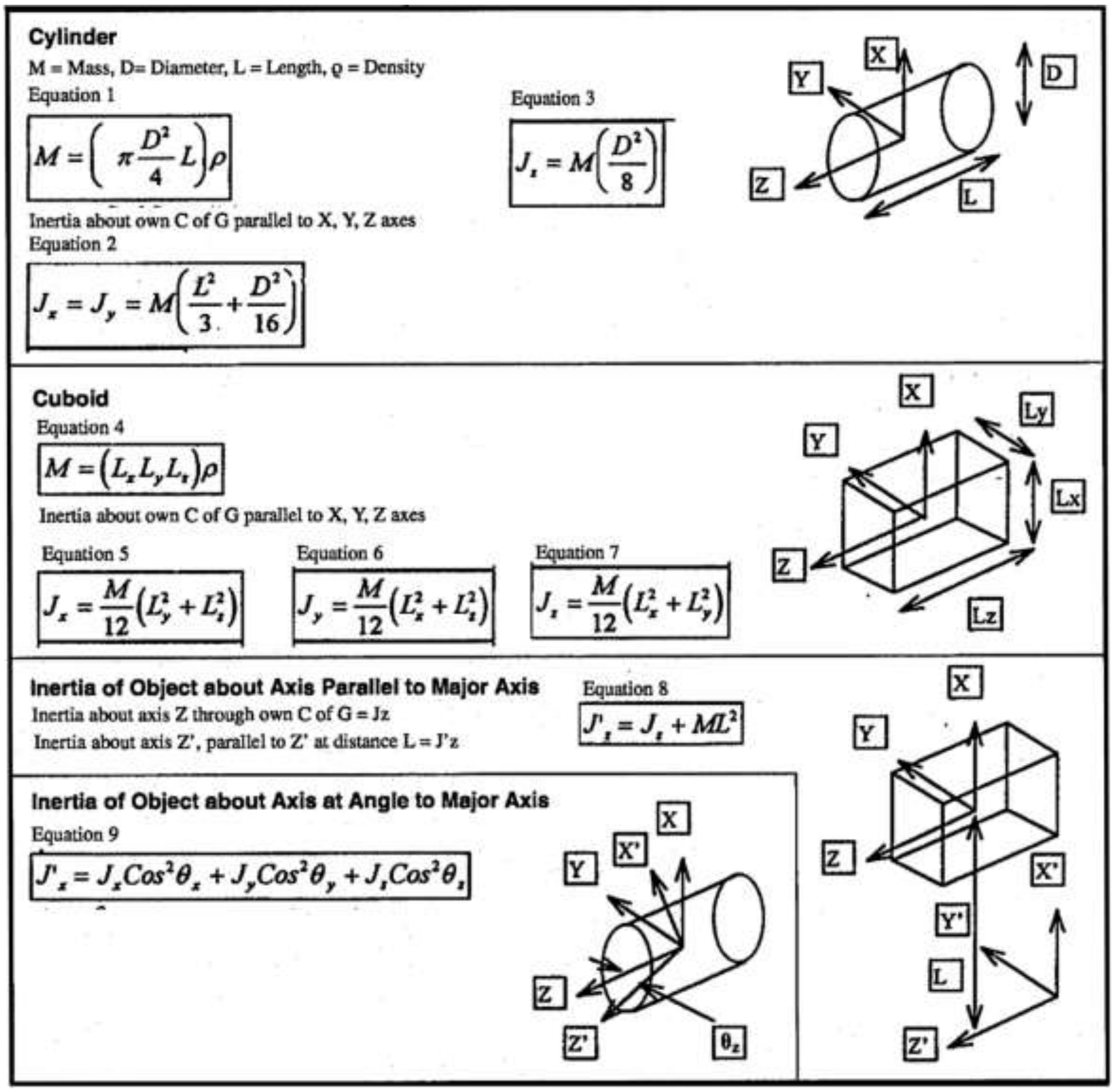


```
MOTION/PAYLOAD SET
                                         1/8
          Group 1
        1 Schedule No[   1]:[****************]
        2 PAYLOAD            [kg]          7.00
        3 PAYLOAD CENTER X   [cm]          0.00
        4 PAYLOAD CENTER Y   [cm]          0.00
        5 PAYLOAD CENTER Z   [cm]          0.00
        6 PAYLOAD INERTIA X  [kgfcms^2]    0.00
        7 PAYLOAD INERTIA Y  [kgfcms^2]    0.00
        8 PAYLOAD INERTIA Z  [kgfcms^2]    0.00
```

Source: Modified image obtained from ROBOGUIDE software.

If you do not know how to determine the inertia of a work object, you can be guided by the following calculations regarding the shape of the object (see in figure 2.24):

Figure 2.24 Calculations to obtain Payload data



Cylinder

M = Mass, D= Diameter, L = Length, ρ = Density

Equation 1

$$M = \left(\pi \frac{D^2}{4} L \right) \rho$$

Inertia about own C of G parallel to X, Y, Z axes

Equation 2

$$J_x = J_y = M\left(\frac{L^2}{3} + \frac{D^2}{16} \right)$$

Equation 3

$$J_z = M\left(\frac{D^2}{8} \right)$$

Cuboid

Equation 4

$$M = \left(L_x L_y L_z \right) \rho$$

Inertia about own C of G parallel to X, Y, Z axes

Equation 5

$$J_x = \frac{M}{12}\left(L_y^2 + L_z^2 \right)$$

Equation 6

$$J_y = \frac{M}{12}\left(L_x^2 + L_z^2 \right)$$

Equation 7

$$J_z = \frac{M}{12}\left(L_x^2 + L_y^2 \right)$$

Inertia of Object about Axis Parallel to Major Axis

Inertia about axis Z through own C of G = Jz

Inertia about axis Z', parallel to Z' at distance L = J'z

Equation 8

$$J'_z = J_z + ML^2$$

Inertia of Object about Axis at Angle to Major Axis

Equation 9

$$J'_x = J_x Cos^2\theta_x + J_y Cos^2\theta_y + J_z Cos^2\theta_z$$

Source: (FANUC Robotics Ibérica S. L., 2018)

During data entry, the TP will warn you if these are acceptable or not, so you must constantly confirm this entry by pressing the F4-YES or F5-NO key when the question “Path and Cycletime will change. Set it? Depending on the case, if the system decides that the data is not acceptable, it will have to change it. Otherwise, if you have finished, press the PREV key.

If you have additional loads in the physical axis of the robots 1 or 3 go to section 2.7.3 ARMLOAD, if not, after creating, configuring or modifying your payload, do a cold start by turning the robot controller off and on again. robot to restart it with the new inserted data.

Up to this point you have already declared your payload and you can call it from the programming panel (see section 2.7.5).

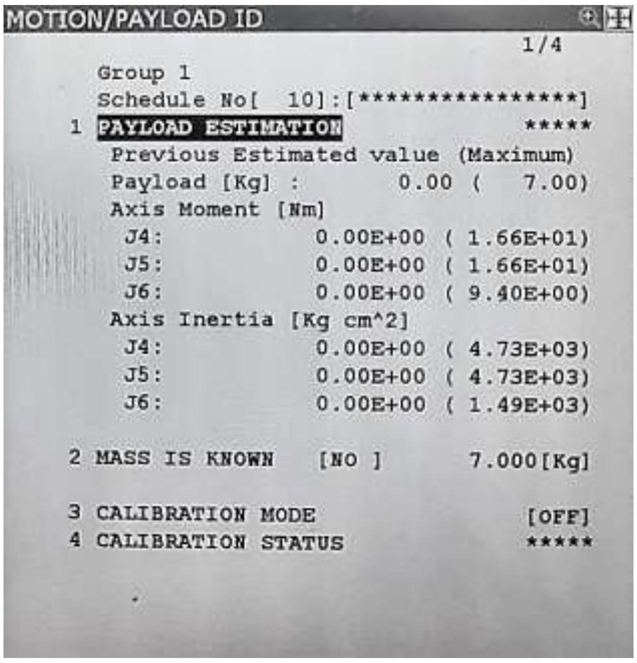
Payload ident

This method automatically calculates the necessary data to declare a payload, here the robot performs a maneuver at low speed and another at high speed only with axes 5 and 6 to determine the parameters of the load (this only works with the automatic mode of the controller). To do so follow the address:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 7-MOTION

After that, we choose the payload we want, positioning the TP cursor on it and then pressing NEXT → F2; IDENT, where you will get the screen of figure 2.25.

Figure 2.25 Payload estimation



```
MOTION/PAYLOAD ID
                                          1/4
    Group 1
    Schedule No[  10]:[****************]
  1 PAYLOAD ESTIMATION                  *****
     Previous Estimated value (Maximum)
     Payload [Kg] :        0.00 (   7.00)
     Axis Moment [Nm]
      J4:             0.00E+00 ( 1.66E+01)
      J5:             0.00E+00 ( 1.66E+01)
      J6:             0.00E+00 ( 9.40E+00)
     Axis Inertia [Kg cm^2]
      J4:             0.00E+00 ( 4.73E+03)
      J5:             0.00E+00 ( 4.73E+03)
      J6:             0.00E+00 ( 1.49E+03)

  2 MASS IS KNOWN   [NO ]       7.000[Kg]

  3 CALIBRATION MODE                [OFF]
  4 CALIBRATION STATUS              *****
```

Source: Elaborated by the work team

Some recommendations before proceeding are as follows:

- You can enter a name for the payload.
- In item number two “MASS IS KNOWN”, if you know the weight of the tool that the robot is loading, change the NO constant to YES and enter the weight in kilograms, otherwise, do not change anything.
- In point three CALIBRATION MODE, unless the robot is factory calibrated, avoid changing the constant OFF.

Now, if you want to change the details of the operation, press NEXT → F4: DETAIL, this will take you to the screen in figure 2.26. In that case you can change the following settings:

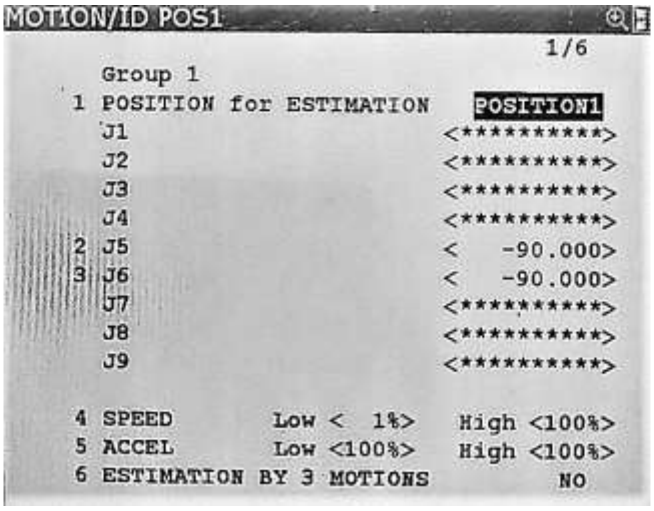
1. You can change the position in degrees to which the robot will move by placing the cursor on axis 5 or 6.

a. Use SHIFT+ RECORD (activating the TP) if you want to save a new position.
b. Use SHIFT + MOVE TO (activating the TP) if you want to move to a set position.
c. Press F2: POS.1/ POS.2 to set the position one or two which is in the system.
d. Press DEFAULT to return to the default values offered by the TP.

2. You can manipulate the speed at which the robot will move, both at high speed and at low speed. If you want to do it, put the cursor on the value you want to change and do it.

Note: Before pressing PREV and returning to the main screen, make sure that the robot is in position 1, to do so press F2: POS.1 to appear (If this option does not appear, ignore the note). Later, with the deadman switch activated and after pressing the RESET key, press SHIFT+ MOVE TO and continue with the procedure.

Figure 2.26 Operation details



```
MOTION/ID POS1
                                             1/6
     Group 1
   1 POSITION for ESTIMATION           POSITION1
     J1                              <**********>
     J2                              <**********>
     J3                              <**********>
     J4                              <**********>
   2 J5                              <   -90.000>
   3 J6                              <   -90.000>
     J7                              <**********>
     J8                              <**********>
     J9                              <**********>

   4 SPEED           Low <  1%>    High <100%>
   5 ACCEL           Low <100%>    High <100%>
   6 ESTIMATION BY 3 MOTIONS                NO
```

Source: Elaborated by the work team

Once the changes have been made, press PREV to return to the screen in figure 2.25 and follow the steps below:

A. Put the TP in its OFF state and if it is not, put the controller in automatic mode.
B. Press SHIFT + RESET or just RESET to clear warnings.
C. Press F4: EXEC and at the Robot moves and estimates prompt. Ready?, press F4: [YES].
D. The robot will move in two maneuvers (one fast and one slow) and the values of moment and inertia of the axes will change. Once done, press F5: APPLY.
E. Lastly, verify that the payload values have changed by pressing PREV→F3: DETAIL and check.

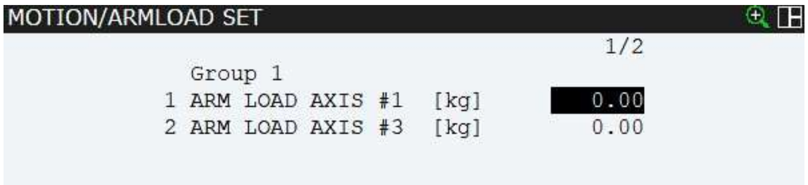
At this point you have already declared your payload and can call it from the programming panel (see section 2.7.5).

2.7.2 Armload

The ARMLOAD, like the previous points, is a way to inform the robot about a load that it must consider (see figure 2.27). In this case, an additional load is reported for axis 1 or axis 3 of the robot. To do this, go to the following location:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 7-MOTION → F4: ARMLOAD

Figure 2.27 Armload set



```
MOTION/ARMLOAD SET
                                           1/2
        Group 1
      1 ARM LOAD AXIS #1   [kg]           0.00
      2 ARM LOAD AXIS #3   [kg]           0.00
```

Source: Imagen obtenida del software ROBOGUIDE

Once there, enter the weight of the loads you need to declare and confirm the data by pressing F4-YES. The software will ask you to reboot the controller, so you will have to do a cold boot by powering off and on the robot controller, so that when it fully reboots, your paylod along with your ARMLOAD will be declared and ready to be called.

2.7.3 Payload check

This option in the TP exists so that we can make quick revisions or to check the adjustment of an already declared payload. To get to it, follow the address:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 8-PayLoadCheck

Once you are at the above address, you will have two options to click and review your payload:

- F4: PLSET: It works to check the setting of a payload. Once pressed, a confirmation question will appear to move the robot, in this case press F3: YES.
- F5: OVLD: Works to check for an overload, so accept all confirmations.

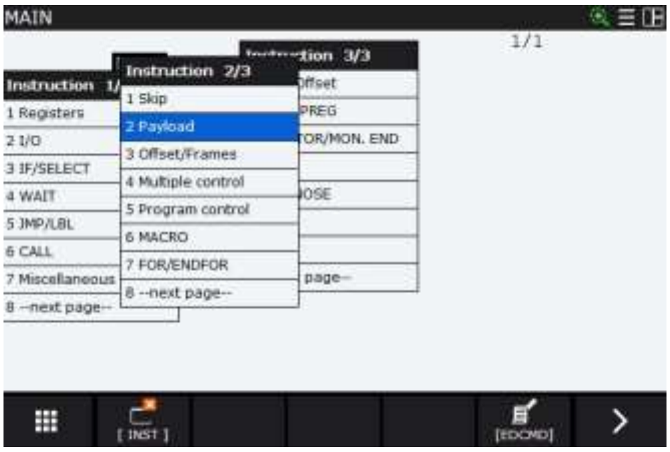
NOTE: For the options provided by the payload check to work properly, check that the robot controller is in its automatic state, that its TP is in the OFF state and that a payload has already been selected. To select it press F1: [TYPE] → 7-Motion → F5: SETIND → Select your payload.

2.7.4 Activating a payload

In order to use a payload, it must first have been declared. Once that is done, we can call it from the programming tree (see figure 2.28), to do this within a program you can press:

NEXT → INST → 8-NEXT PAGE → 2-PAYLOAD → ENTER → ENTER

Figure 2.28 Activating a payload from a program

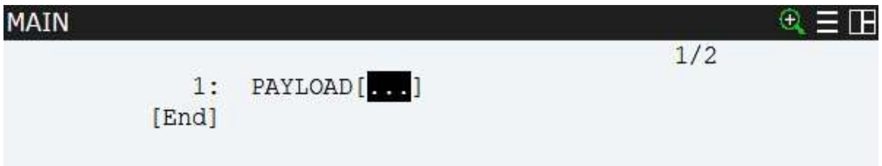


Source: Modified image obtained from ROBOGUIDE software

When the instruction is created, we position ourselves with the TP cursor inside the brackets of it and select the option that has been declared by inserting the paylod number with the following options (see figure 2.29).

- DIRECT → You can enter the payload number directly.
- INDIRECT → You can place the payload number through registers.
- [LIST] →A physical list of available payloads will appear, here choose the one that best suits you.

Figure 2.29 Selection of the payload to use



```
   1:  PAYLOAD[...]
  [End]
```

Source: Modified image obtained from ROBOGUIDE software

2.8 Limits

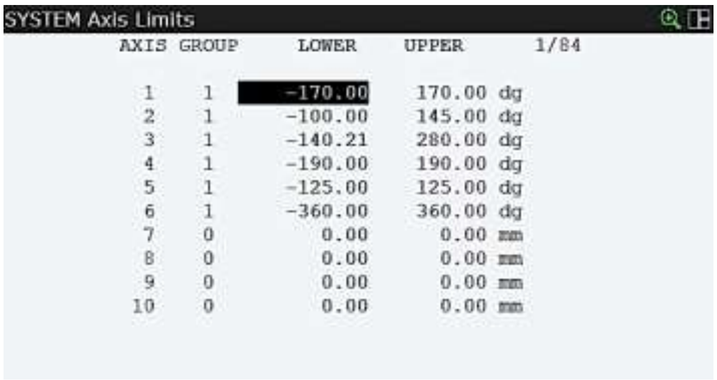
A limit is important for a robot, since they are accesses that each axis has and that allow it to move a certain distance within the work zone. FANUC Robotics Ibérica tells us that there are three types of limits: software, electrical and mechanical. The last type always depends on the robot or its axes to be configured, the others are described below.

2.8.1 Axis limits

The Axis limits or axis limit, are an option that the software gives us to configure the number of degrees to which the axis box is allowed to move (see figure 2.31). When we reach the limit, the robot stops and sends the warning "JOG-13 Strock limit (G:1 A:2)" to warn us that the stroke limit has been reached, in this case, it is only enough to move the axis inversely. To configure the limits we have to follow the address:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 5-AXIS LIMITS

Figure 2.30 Standard configuration for axis limit on the FANUC LR Mate 200iD robot



```
SYSTEM Axis Limits
      AXIS GROUP     LOWER     UPPER      1/84

        1    1      -170.00    170.00 dg
        2    1      -100.00    145.00 dg
        3    1      -140.21    280.00 dg
        4    1      -190.00    190.00 dg
        5    1      -125.00    125.00 dg
        6    1      -360.00    360.00 dg
        7    0         0.00      0.00 mm
        8    0         0.00      0.00 mm
        9    0         0.00      0.00 mm
       10    0         0.00      0.00 mm
```

Source: Modified image obtained from ROBOGUIDE software

Once the values we need are configured, the robot will ask us to restart the power cycle, so we must turn it off and turn it on, so the changes will be made.

2.8.2 Electric limits

The electrical limits, on the other hand, are limits that, when activated, the selected axis is impossible to move and will only be achieved by pressing SHIFT+RESET or searching among system variables. This type of limit is optional and to do so we go to:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 4-RELEASE

Once there, we position the cursor on the axis to which we want to apply an electrical limit and press F2:RELEASE**.**

References

Fanuc. (2018). *Robot industrial FANUC LRMate 200iD*. https://www.fanuc.eu/es/es/robots/página-filtro-robots/serie-lrmate/lrmate-200-id

FANUC América Corporation. (2016). *Serie LR Mate _ Robots ligeros y compactos _ FANUC América*. https://www.fanucamerica.com/products/robots/series/lr-mate
FANUC Robotics Ibérica S. L. (2018). ROBÓTICA PROGRAMACIÓN FANUC. *SEAT*. http://www.infoplc.net/files/descargas/fanuc_robotica/infoPLC_net_ManualFANUCR-J3iB.pdf

Navarro Sanz, I. (2009). Programación de un proceso de paletización y posicionamiento flexible en un robot Fanuc. *Universitat Politécnica de Catalunya*. http://hdl.handle.net/2099.1/8147

Pinto Hernández, J. de. (2016). *Programación de robot Kuka para paletizado*. http://hdl.handle.net/10016/24236

Ponsa Asensio, P. (2003). Estudio sobre robots Fanuc. *Universitat Politécnica de Catalunya*, 139. http://hdl.handle.net/2117/185131

Chapter 3 Programming and creating trajectories

Abstract

A trajectory is considered as a finite series of points in space through which the robot must pass when executing a specific task and which have a starting point and an end point. The problem lies in the control needed to force the robot to follow such a trajectory considering different characteristics, having to resort to programming. Speaking particularly of the LR Mate 200iD version of the robot and its R-30Ib Plus controller, programming is done first through the physical recording of the points that make up the trajectory. Subsequently, both a command editor and instructions that allow specific commands to be given according to the function being sought are used and, finally, the execution of the final program is carried out. This makes the main objective of the chapter to focus on describing the steps to carry out a program, generating the trajectories and determining the function of the instructions provided by the robot controller.

Path, Program, Instructions

3.1 Introduction

La Trajectory planning is an important part in the programming of industrial robots, this is due to the fact that it consists in the generation of the control parameters necessary for the robot to execute a specific task, depending on the conceptual foundations of a manipulator robot (Sierra Bueno & Martinez Angel, 2002). Starting from the previous point, it is necessary to determine the control that the robot will have to carry out a correct execution, necessary control for the different existing applications in the industry (Reyes Cortés, 2011). This in turn generated programming, which is a tool that uses different instructions, which are responsible for controlling different functions in the robot to achieve all of the above. Looking from another perspective, mathematically a model is obtained to address this control in a broader way, carrying out the most appropriate study (Saltarén Pazmiño et al., 2000), in this case dividing it into study blocks, which makes it a time-consuming process. and complex.

This chapter aims to develop a description of how a trajectory is made and programmed with the Fanuc LR Mate 200iD robot using the interface provided by the R-30iB Plus controller. This by determining technical data and instructions so that the content displayed can be replicated by a user.

Creating a trajectory

A trajectory is a locus made up of several points in space through which a physical body (in this case the robot) must pass when performing a movement. For this to be possible, it is necessary for the robot to move from an initial point to an end point, added to this, the technological advance allowed the trajectories to be increasingly complex (Saltarén Pazmiño et al., 2000).

3.1.1 Creation of a program

A program is a series of instructions that control the operation of the robot, as well as the equipment that is associated with it (Guerrero Ávila & Llama Leal, 2017), however, the program in which those instructions will be saved must be created. To do it physically in the robot, the teach pendant is activated to its ON state and the SELECT keys are pressed, giving a screen which will contain all the programs created and to be created. After that, the F2:CREATE key is pressed, we will obtain a screen where the name for the program to be developed will be placed. In that case, the TP gives us table 3.1, which contains the options to write that name through the function keys F1-F5.

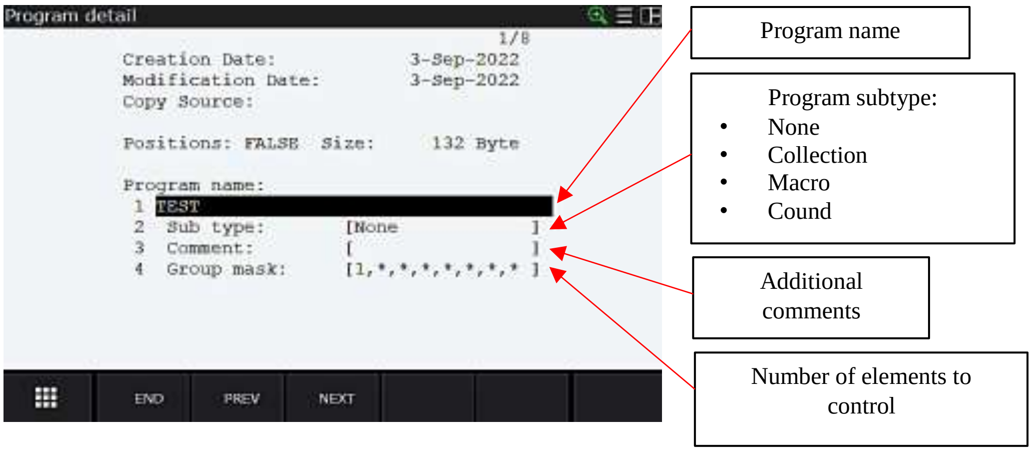
Table 3.1 Options for developing a name for the program to be created

Option	F1	F2	F3	F4	F5
Words (Default words)	PRG	MAIN	SUB	TEST	
Uppercase	ABCDEF	GHIJKL	MNOPQR	STUVWX	YZ_@*.
Lowercase	abcdef	ghijkl	mnopqr	stuvwx	Yz_@*.
Extra options	OVRWRT	INSERT	CLEAR		

Source: (FANUC Robotics Ibérica S. L., 2018)

Later, when you write the name, it has to be validated by pressing the ENTER key and then F2:DETAIL. A window will appear with options such as: subtype, comments and number of elements to control (see figure 3.1). Unless an option is already being considered, it is advisable to press F1:END

Figure 3.1 Additional information obtained when developing a program



```
Program detail
                                              1/8
      Creation Date:               3-Sep-2022
      Modification Date:           3-Sep-2022
      Copy Source:

      Positions: FALSE  Size:       132 Byte

      Program name:
        1 TEST
        2   Sub type:            [None             ]
        3   Comment:             [                 ]
        4   Group mask:          [1,*,*,*,*,*,*,* ]
```

Source: Modified image obtained from ROBOGUIDE software

NOTE: In "Group mask", the group of axes or movements that will be controlled is defined, that is, each number one represents the systems to be used and each asterisk (*) represents systems that are not used. If there is no one in this section, move instructions cannot be inserted in the programming.

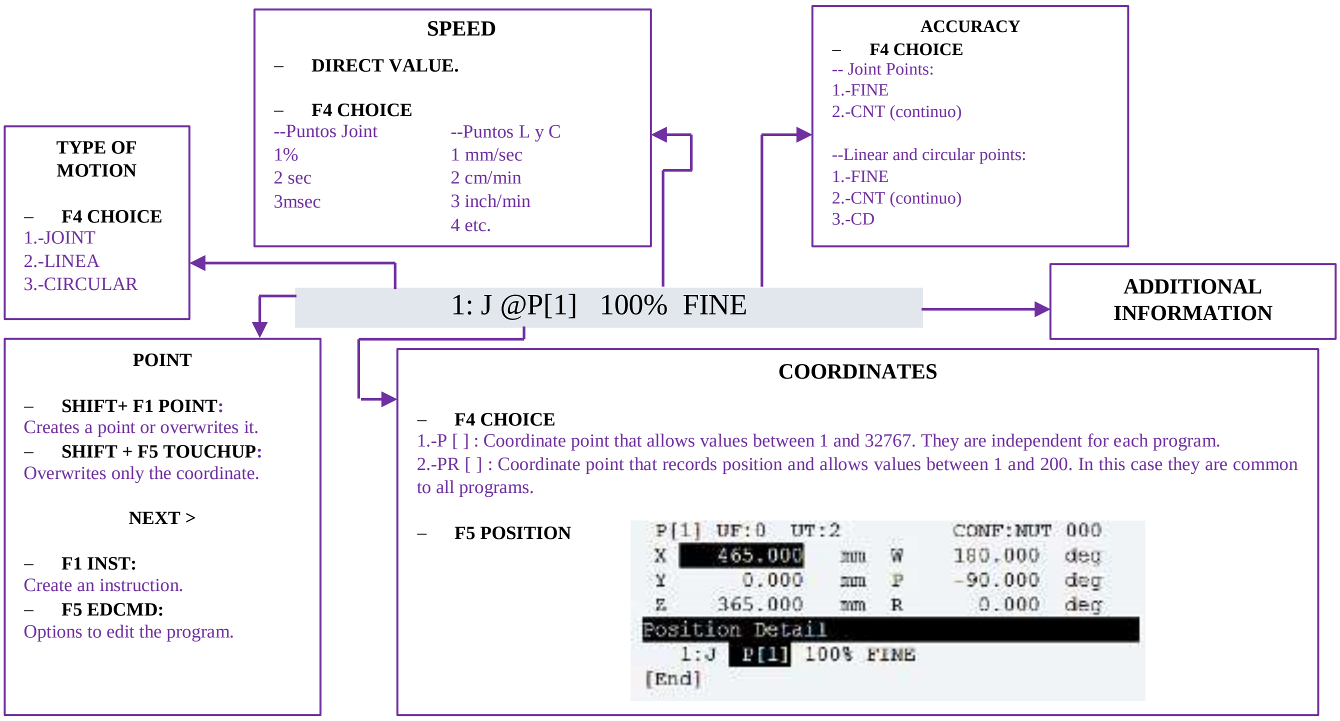
Finally, by pressing END, the TP will take you to the main programming tree screen where you can start creating toolpaths through point engraving.

3.1.2 Creation of points

The creation of points is simple, when you have moved the robot to the position or location you need, from the teach pendant you can press any of the following keys or combination of keys:

– F1: POINT = Shows the options of points that can be created, and you must choose one.
– SHIFT+F1: POINT at the same time = Generates a standard point (see figure 3.2).

Figure 3.2 Content of a standard point



1: J @P[1]   100% FINE

SPEED

– **DIRECT VALUE.**

– **F4 CHOICE**

--Puntos Joint	--Puntos L y C
1%	1 mm/sec
2 sec	2 cm/min
3msec	3 inch/min
	4 etc.

ACCURACY

– **F4 CHOICE**

-- Joint Points:
1.-FINE
2.-CNT (continuo)

--Linear and circular points:
1.-FINE
2.-CNT (continuo)
3.-CD

TYPE OF MOTION

– **F4 CHOICE**
1.-JOINT
2.-LINEA
3.-CIRCULAR

ADDITIONAL INFORMATION

POINT

– **SHIFT+ F1 POINT**:
Creates a point or overwrites it.
– **SHIFT + F5 TOUCHUP**:
Overwrites only the coordinate.

NEXT >

– **F1 INST:**
Create an instruction.
– **F5 EDCMD:**
Options to edit the program.

COORDINATES

– **F4 CHOICE**
1.-P [ ] : Coordinate point that allows values between 1 and 32767. They are independent for each program.
2.-PR [ ] : Coordinate point that records position and allows values between 1 and 200. In this case they are common to all programs.

– **F5 POSITION**

```
P[1]  UF:0  UT:2           CONF:NUT  000
X     465.000   mm   W     180.000   deg
Y       0.000   mm   P     -90.000   deg
Z     365.000   mm   R       0.000   deg
Position Detail
   1:J  P[1]  100%  FINE
[End]
```

Source: Elaborated by the work team

Note: Each of the characters shown in figure 3.2 can be used or configured depending on the application it is intended for.

3.1.3 Types of programming points and movements

The R-30iB Plus version of the controller allows you to select different points and movements according to the function that is needed, where the movements are generated point by point developing different shapes (see figure 3.3), which can be modified with respect to their speed. and accuracy.

Type of points

- P[n] = They are normal points; they work only for the programs in which they are recorded.
- PR[n] = They are points that, when recorded, work for any program globally.

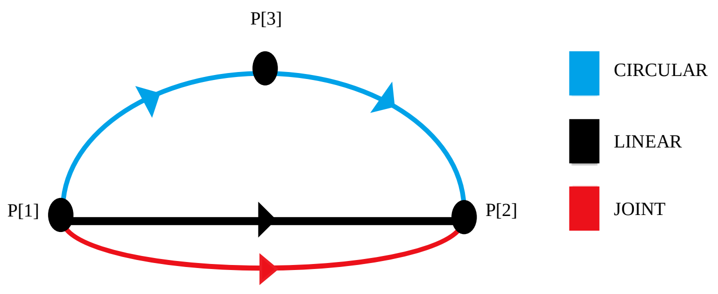
Types of point-to-point movements

- Joint (J): This is a joint movement where each of the robot's axes is used to reach a position. It is commonly used in free spaces, that is, workspaces where there are no obstacles or when the robot returns to its mechanical zero, so rapid movements can be made. In addition, with this option the robot autonomously seeks to perform the movement without forcing any of its axes (see figure 3.3, red path).

- Linear (L): This type of movement forces the robot to follow a straight line that goes from an initial position to a final position (see figure 3.3, black trajectory). They are mainly used in movements that require precision or for movements in which the work area has multiple obstacles that can cause a collision, so they will always be slow. Not to mention that, with these types of movements, the robot's axes are forced to follow the trajectory by struggling.

- Circular (C): Unlike the previous movements, this one requires an additional point that serves as a reference to complement the circular path through them. To declare the movement, the robot will have to be positioned at P[1], which will function as an attack point and later P[3] followed by P[2] are recorded, whose position will determine the dimensions, as well as the shape of the movement. the circumference or arc (see figure 3.3, blue path).

Figure 3.3 Types of point-to-point movement



Source: Elaborated by the work team

3.1.4 Speeds and accuracies

Speed

The speed is expressed according to the type of movement (FANUC Robotics Ibérica S. L., 2018). To choose the type of speed we must position ourselves in the speed option at the point at which we want to change it, and then press CHOICE. The options to choose from are the following:

- Joint
  - Value in % of the maximum speed
  - Value in seconds (An arrival time must be imposed)

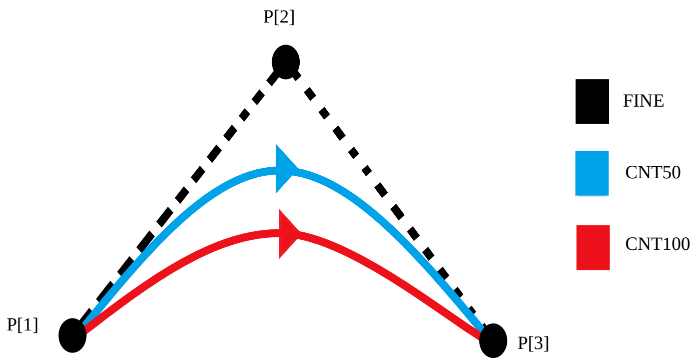
- L or C
    - Value in mm/s
    - Value in cm/min
    - Value in seconds (An arrival time must be imposed)

Accuracy

Accuracy in simple words is how a robot gets to a point or finishes a move. For this robot there are three types of accuracies:

- FINE: It is a type of exact precision, where the robot will reach the specified point regardless of other aspects and thus increasing the cycle time but developing a safer program depending on the application.
- CNT (Continuous): It is a precision in which the robot does not specifically arrive at a point, but rather gets too close to it, reducing the cycle time. Here the percentage of continuity while it is lower, the precision will be higher (see figure 3.4).

Figure 3.4 Point-to-point accuracies

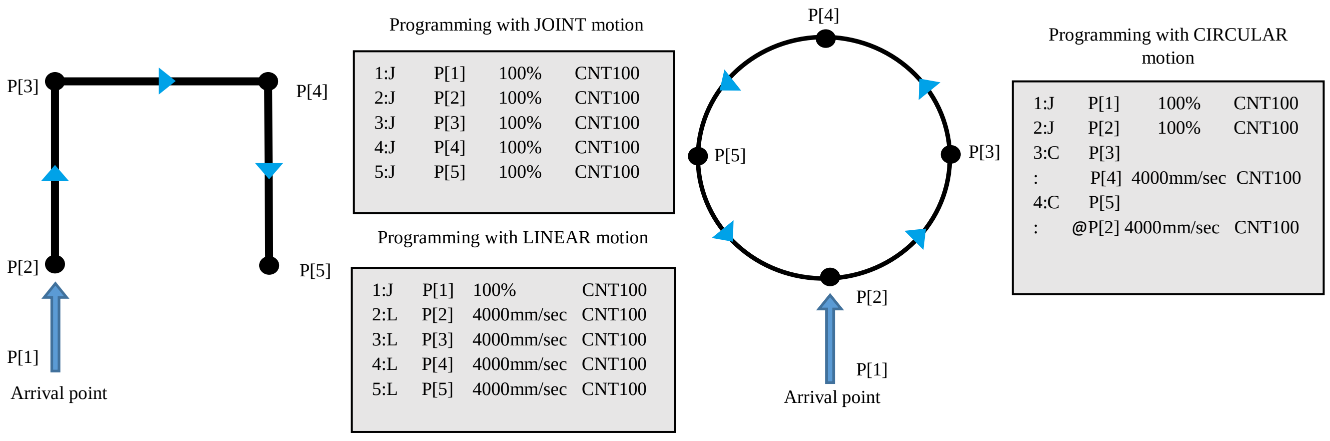


Source: Elaborated by the work team

3.1.4.1 Examples of trajectories according to the recording of points

Next, figure 3.5 shows some examples of the use of the three most common types of movement and how to record them so that the final movement of the robot is correct.

Figure 3.5 Examples of trajectories



Programming with JOINT motion

```
1:J    P[1]    100%    CNT100
2:J    P[2]    100%    CNT100
3:J    P[3]    100%    CNT100
4:J    P[4]    100%    CNT100
5:J    P[5]    100%    CNT100
```

Programming with LINEAR motion

```
1:J   P[1]  100%          CNT100
2:L   P[2]  4000mm/sec    CNT100
3:L   P[3]  4000mm/sec    CNT100
4:L   P[4]  4000mm/sec    CNT100
5:L   P[5]  4000mm/sec    CNT100
```

Programming with CIRCULAR motion

```
1:J    P[1]     100%      CNT100
2:J    P[2]     100%      CNT100
3:C    P[3]
:       P[4]  4000mm/sec  CNT100
4:C    P[5]
:     @P[2] 4000mm/sec    CNT100
```

Source: Elaborated by the work team

3.1.5 Execution of programs

The execution of a program will always assume that you already have a program itself, either to start a process or for testing. In any case, two types of executions can be used:

Execution in manual mode (Using the Deadman Switch)

– Using the SHIFT + FWD keys: This method will cause the program to be executed descendingly, that is, the program will be read and executed according to the numerical order in which it was written.

– Using the SHIFT + BWD keys: Unlike FWD, this method will execute the program in an ascending manner, in such a way that a kind of backtracking will be observed. This method can be used to revalidate moves or roll back actions.

– Using STEP and SHIFT +FWD/BWD: This option allows you to manipulate the execution of a program, making the movements step by step (reading line by line in the main program and stopping at each one).

Execution in automatic mode

For this method, the following steps must be followed.

1) Press RESET to remove warnings.

2) TP in OFF state.

3) Set the TP and the controller to automatic mode.
a. Select the automatic mode of the controller by turning the main key.
b. On the TP press: Menu → Next → System →Config → Option Local / Remote = Local.

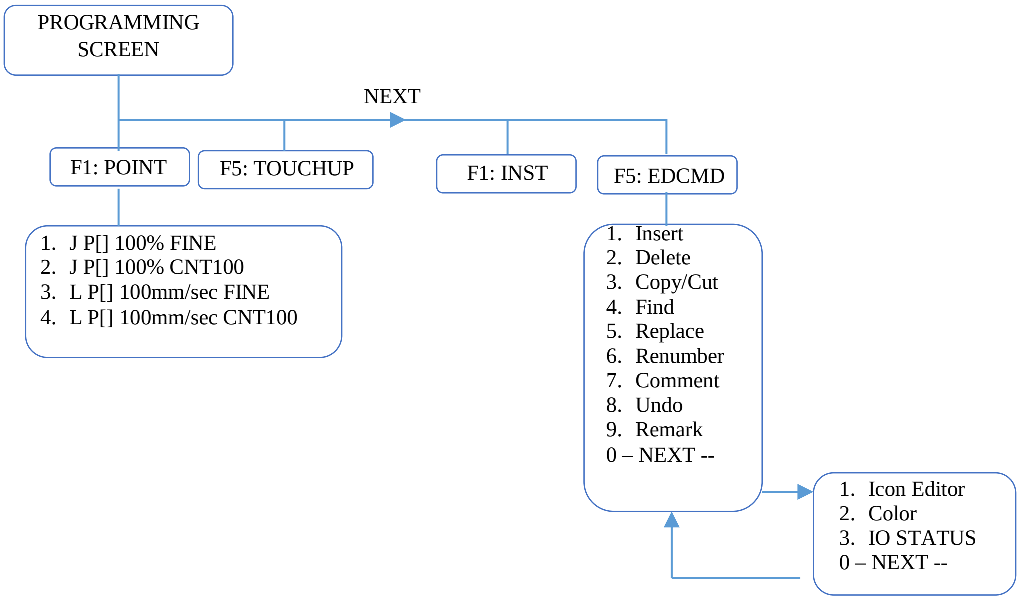
4) Press FCNT → Abort All → Select → Choose the program to execute.

5) Press the CYCLE START button on the controller as many times as desired.

3.2 Programming

Some of the options offered by teach pendan focused on editing and programming are shown in figure 3.6.

Figure 3.6 Diagram of the options offered by the TP with respect to programming.



Source: Elaborated by the work team

3.2.1. Positions

Record a standard position/point

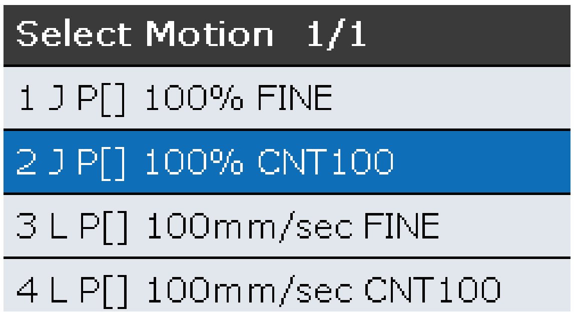
To record a default or standard position, press SHIFT + F1: POINT on the TP once the robot is located. This will generate a position referenced to a point, which has the following structure:

```
1: J P[1] 100% FINE
```

Code 3.1: Standard point

If you want to change the default structure, press F1: POINT. When the position options shown in figure 3.7 appear, choose the one that suits you best.

Figure 3.7 Options of points to record

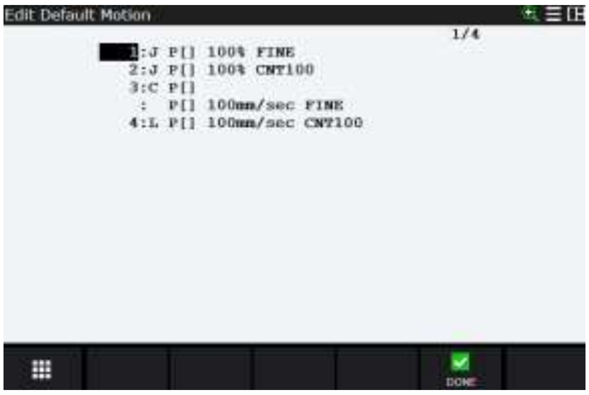


```
Select Motion  1/1
1 J P[] 100% FINE
2 J P[] 100% CNT100
3 L P[] 100mm/sec FINE
4 L P[] 100mm/sec CNT100
```

Source: Image obtained from Roboguide software

If none of the options convince you, press F1: POINT and then F1: ED_DEF. Here you can modify the standard positions as you like (see figure 3.8), just position the cursor on the part you want to change and press F4: CHOICE. When you are finished press F5: DONE.

Figure 3.8 Standard point modification



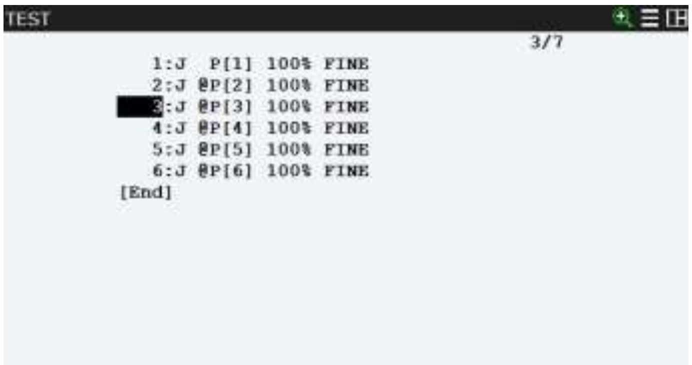
```
Edit Default Motion
                                   1/4
        1:J P[] 100% FINE
        2:J P[] 100% CNT100
        3:C P[]
         :  P[] 100mm/sec FINE
        4:L P[] 100mm/sec CNT100
```

Source: Image obtained from Roboguide software

Configure a standard position/position

There are two ways to set the position of a recorded point, overwrite it or change it manually. To overwrite it move the robot to the new position and place the TP cursor on the point you want to change, then press SHIFT + F5: TOUCHUP and you will see the message "Position has been recorded to [*]" (see Figure 3.9).

Figure 3.9 Using the TOUCHP function

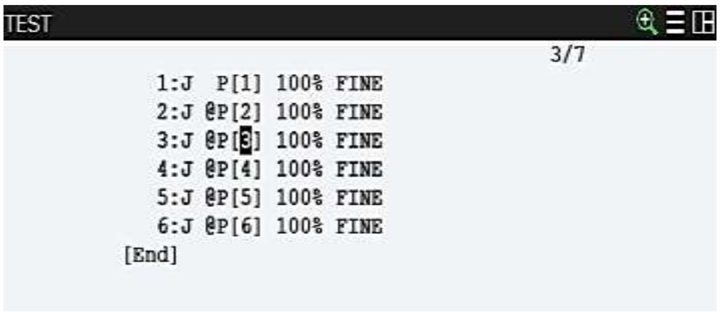


```
TEST
                                          3/7
     1:J  P[1] 100% FINE
     2:J @P[2] 100% FINE
     3:J @P[3] 100% FINE
     4:J @P[4] 100% FINE
     5:J @P[5] 100% FINE
     6:J @P[6] 100% FINE
    [End]
```

Source: Image obtained from Roboguide software

Otherwise, change the position manually by placing the TP cursor on the number inside the brackets of the point to be set (see figure 3.10) and press F5: POSITION.

Figure 3.10 Location of the manual positional setting

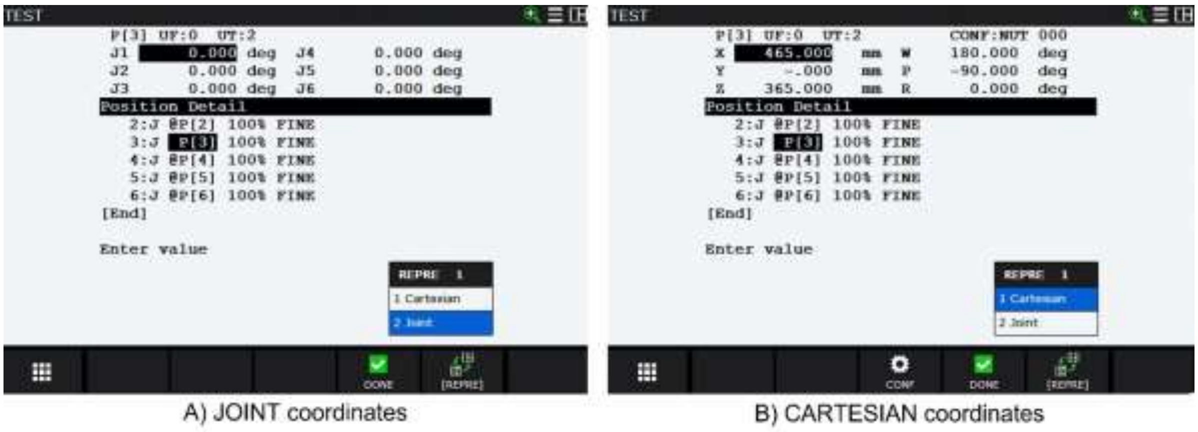


```
TEST
                                          3/7
     1:J  P[1] 100% FINE
     2:J @P[2] 100% FINE
     3:J @P[3] 100% FINE
     4:J @P[4] 100% FINE
     5:J @P[5] 100% FINE
     6:J @P[6] 100% FINE
    [End]
```

Source: Image obtained from Roboguide software

Once inside the point, change the type of coordinates by pressing F5: REPRE and select the desired coordinates, Joint or Cartesian (see figure 3.11). When finished press F4: DONE.

Figure 3.11 A) Position of the point in Joint coordinates, B) Position of the point in World coordinates



```
TEST
   P[3] UF:0  UT:2
   J1      0.000 deg  J4     0.000 deg
   J2      0.000 deg  J5     0.000 deg
   J3      0.000 deg  J6     0.000 deg
  Position Detail
     2:J @P[2] 100% FINE
     3:J  P[3] 100% FINE
     4:J @P[4] 100% FINE
     5:J @P[5] 100% FINE
     6:J @P[6] 100% FINE
  [End]

  Enter value
```

A) JOINT coordinates

```
TEST
   P[3] UF:0  UT:2              CONF:NUT 000
   X    465.000   mm  W    180.000  deg
   Y      -.000   mm  P    -90.000  deg
   Z    365.000   mm  R      0.000  deg
  Position Detail
     2:J @P[2] 100% FINE
     3:J  P[3] 100% FINE
     4:J @P[4] 100% FINE
     5:J @P[5] 100% FINE
     6:J @P[6] 100% FINE
  [End]

  Enter value
```

B) CARTESIAN coordinates

Source: Image obtained from Roboguide software

3.2.2 Command Editor "EDCMD"

The command editor appears when pressing the F5: EDCMD key within the programming screen (see Figure 3.6). This gives us several options that we can use to modify a program, these options and their function are as follows:

Insert

To add lines within a program we must place the TP cursor in the space where we want to insert them, as shown in code 3.2:

```
 1:  J P[1] 100% FINE
 2:  L P[2] 4000mm/s FINE
```

Code 3.2: Cursor location for the "Insert" command

Now press 1.Insert and when "How many line to insert?" appears, enter the number of lines to insert (in this example 2) and press ENTER. You will get the function shown in code 3.3.

```
1: J P[1] 100% FINE
2:
3:
4: L P[2] 4000mm/s FINE
```

Code 3.3: "Insert" command applied

Delete

It works for deleting programming lines where, before pressing 2.Delete, you must first place the TP cursor on the line to be deleted or on the beginning of a group of lines to be deleted (see in code 3.4).

```
1: J P[1] 100% FINE
2: L P[2] 4000mm/s FINE
3: L P[3] 4000mm/s FINE
4: L P[4] 4000mm/s FINE
5: L P[5] 4000mm/s FINE
```

Code 3.4: Cursor location for the "Delete" command

Then press 2.Delete. If you only want to delete a single line press F4: YES, otherwise, using the up and down arrows of the TP, mark the other lines as in code 3.5 and press F4: YES.

```
  1: J P[1] 100% FINE
  2: L P[2] 4000mm/s FINE
  3: L P[3] 4000mm/s FINE
  4: L P[4] 4000mm/s FINE
  5: L P[5] 4000mm/s FINE
```

Code 3.5: Cursor location for the "Delete" command

At the end you will have something similar to code 3.6:

```
1:  J P[1] 100% FINE
2:  L P[5] 4000mm/s FINE
```

Code 3.6: "Delete" command applied

Copy/Cut

Press 3.Copy/Cut, then place the TP cursor on the line to be edited or on the beginning of a group of lines as in code 3.2. If you want to copy more than one line, use the up and down arrows of the TP to select them, otherwise proceed with the following.

After selecting them, you can use F2: COPY (only to copy lines) or F3: CUT (to relocate the lines by removing them from the program and placing them elsewhere by pressing F5: PASTE). Then, position the cursor again on the line where the new points will be copied or moved, press F5: PASTE and select one of the following options.

- F2: LOGIC = Pastes only the logic of the copied point (or line) without position or identification (see code 3.7).

```
3: L P[ ... ] 4000mm/s FINE
```

Code 3.7: “Copy/Cut” command applied using the F2: LOGIC

- F3: POSID = Pastes or relocates programming lines in the most accurate way, respecting the logic, identification and position of the point (see in code 3.8).

```
3: L P[2] 4000mm/s FINE
```

Code 3.8: “Copy/Cut” command applied using the F3: POSID

- F4: POSITION =Puts the point respecting the logic and the position, but changes the identification number generating a new one (see code 3.9), following the numerical order of the programming.

```
3: L P[3] 4000mm/s FINE
```

Code 3.9: “Copy/Cut” command applied using the F4: POSITION

- F5: CANCEL = Deletes the copy or cut memory, so in case you want to copy or cut programming lines, you have to select again those lines and then paste them in their new location, in addition to choosing one of the options presented up to this point.
- NEXT+F1: R-LOGIC = Pastes the logic of the lines in reverse order without position or identification, generating a code similar to code 3.7.
- NEXT+F2: R-POSID/ F3: RM-POSID = Pastes lines in reverse order respecting their configuration, generating a code similar to code 3.8.
- NEXT+F4: R-POSID / F5: RM-POSID = Pastes lines in reverse order changing their identification, generating a code similar to code 3.9.

Find

Allows you to find instructions within the programming (the types of instructions will be discussed later in this chapter). To find them, just press 4. Find and depending on the instruction, you will be prompted for specifications in any case you will have to press F4: NEXT to find it or F5:EXIT to exit.

Replace

It allows to replace instructions within the programming (the types of instructions will be discussed in this chapter). In this case, it only works for move instructions that have characteristics in common, for example, replace the JMP LBL [1] tag by JMP LBL [2]. To do this we must place the cursor on the instruction we want to replace (see in code 3.10).

```
 1: J P[1] 100% FINE
 2: L P[3] 4000mm/s FINE
 3: JMP LBL [2]
```

Code 3.10: Cursor location for the “Replace” command

Once that is done press F5: EDCMD→ 5. Replace → search for the instruction → enter the number of the instruction to be replaced, this will cause the matches to appear in the program, as can be seen in code 3.11.

```
 1: J P[1] 100% FINE
 2: L P[3] 4000mm/s FINE
3: JMP LBL [2]

  3:      JMB LBL [2]
```

Code 3.11: “Replace” command applied

Press ENTER → Enter the number of the instruction to be replaced → ENTER. Next, the question "Replace OK?" will appear. And we will have 4 options to choose from:

- F2: ALL = The instruction is modified globally.
- F3: YES = Accept the initial question.
- F2: NEXT = Go to the next line.
- F2: EXIT = Exit the function.

Renumber

This option renumbers all the points in the program in ascending order, to do this click:

F5: EDCMD→ 6.Renumber → F4:YES/F5:NO

Comment

The "comment" function removes or displays comments related to registers (R[ ]) or inputs and outputs (I/O). To activate it, press:

F5: EDCMD→ 7. Comment.

Undo

This option reverses an action, in other words, it cancels actions that were performed by mistake or for some other reason, returning to a previous state or condition. It can be performed by pressing:

F5: EDCMD→ 8. Undo → F4:YES/F5:NO

Remark

Causes a normal programming line to become a comment and, therefore, when executing a program, the TP does not identify that line. To do this, position the cursor at the point to remark and press F5:EDCMD→ 9. Remark → F4: REMARK / F5: UNREMARK (in case you need to remove the function).

```
1:  J P[1] 100% FINE
2:  //L P[2] 4000mm/s FINE
```

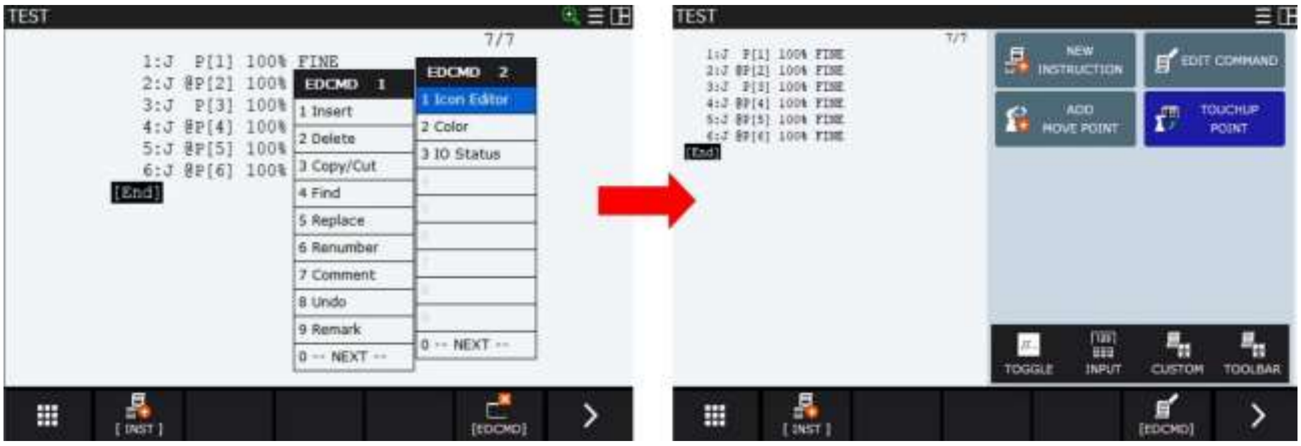
Code 3.11: “Remark” command applied

Icon Editor

It is a function that allows us to use a screen with icons to program in a more intuitive way (see Figure 3.12). From this screen we can manipulate all the functions mentioned above, as well as record points, add instructions, etc. To get this screen click:

F5: EDCMD→ 0. Next → 1. Icon Editor → ENTER

Figure 3.12 Activating or deactivating the icon editor



```
1:J  P[1] 100% FINE
2:J @P[2] 100% FINE
3:J  P[3] 100% FINE
4:J @P[4] 100% FINE
5:J @P[5] 100% FINE
6:J @P[6] 100% FINE
[End]
```

Source: Modified image obtained from ROBOGUIDE software.

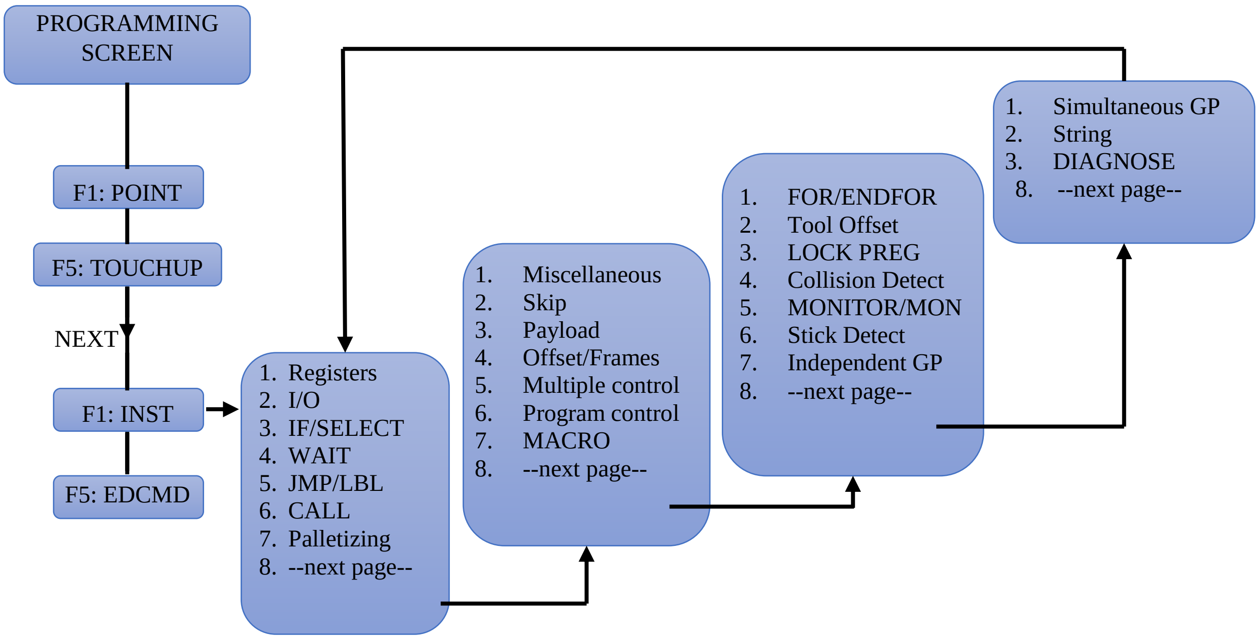
To remove it, press again:

F5: EDCMD→ F4: EXIT ICON

3.2.3 TP instructions

The options offered by the TP focused on the addition of instructions for a program can be subdivided into categories containing similarities (see Figure 3.13). The most important instructions that this version of the robot allows us to use can be seen below:

Figure 3.13 Diagram of instructions that can be called by the TP during programming



Source: Elaborated by the work team

3.2.3.1 Instructions with registers and position registers

Registers are global variables that can be accessed by all programs created and to be created. There are two types of variables, registers and position registers. The "R[ ]" registers are data or real values (32 bits) or integers, while the "PR[ ]" position registers can contain joint, Cartesian or matrix coordinate points.

- **Registers**

This robot model contains up to 200 configurable records and we can see a general list, as well as the content of each of the elements in it by clicking on the TP at the following address.

DATA→ F1:[TYPE] → 1-Registers

To insert an entry into a program, you can press the following key address from the programming screen. Now, when we use a register as an instruction, we must place the identification and the value of the register. The latter we can do directly or indirectly as you choose:

NEXT→ F1: [INST] → 1-Registers→ Seleccione una opción a su conveniencia → 1 -R[ ]

- Directly: The value we give to the register is stored directly in the specified register, as can be seen in Equation 3.1.

$$R[1] = 10 \ \rightarrow \ the\ value \text{ "10" } is\ stored\ in\ R[1] \tag{3.1}$$

- Indirectly: Here, the value we give to the record is stored according to the content of that same record. For example, if for us

$$R[R[1]] = 20 \ \rightarrow \ pero\ en\ DATA\ R[1] = 10 \ \rightarrow \ the\ value\ is\ stored\ in\ R[10] \tag{3.2}$$

Therefore, the final result will be

$$R[10] = 20 \tag{3.3}$$

When we call a register from an instruction, we are given a choice of arithmetic operations and some statements to obtain a result and store it in that register or in a different register. The way to use these tools follows the following format:

$$R[n] = [value] \quad o \quad R[n] = [value]\ [operator]\ [value] \tag{3.4}$$

Where the following options can be chosen:

- Operations
  - Add “+”
  - Subtraction " - "
  - Multiplication " * "
  - Division " / "
  - Integer division " DIV "
  - Remainder of a division " MOD "
- Value
  - Register R[n]
  - Constant
  - E/S
  - TIMER […]
  - Etc.

- **Position registers**

This robot model contains up to 100 configurable position registers and we can see a general list as well as the content of each of the elements in it by clicking on the TP.

DATA→ F1:[TYPE] → 2-Position Reg

To insert a position register in a program, press from the programming screen.

NEXT→ F1: [INST] → 1-Registers→ Seleccione una opción a su conveniencia → 2-PR [ ]

Now, when we use a register as an instruction, we must place the identification and the position of the register through a dot. The latter we can do directly or indirectly as you choose:

- Directly: The point we record in the position register is stored directly in the specified register, i.e., if for us.

$$PR[1] = P\ [1] \ \rightarrow\ point\ P[1]\ is\ stored\ in\ PR[1]. \qquad (3.5)$$

- Indirectly: Here, the point that we record in the position register is saved according to the register content that is specified. For example, if for us

$$PR\big[R[1]\big] = P\ [2] \rightarrow but\ in\ DATA\ R[1] = 10\ \rightarrow\ the\ point\ is\ stored\ in\ PR[10]. \qquad (3.6)$$

Therefore, the final result will be

$$PR\ [10] = P[2] \qquad (3.7)$$

When we call a position register from an instruction in a program, we are given a choice of arithmetic operations and some statements to obtain a result and store it in the same or a different register. The way to use these tools follows the following format:

$$PR[n] = [point]\ \ o\ \ \ PR[n] = [point]\ [operator]\ [point] \qquad (3.8)$$

Where the following options can be chosen:

- Operator
  - Add “ + “
  - Sustraction “ – “

- Point
  - Position P[n].
  - Position register PR[n].
  - Current position of the robot in degrees axis by axis JPOS.
  - Current position of the robot in Cartesian LPOS.
  - User frame UFRAME [ ]/ UTOOL [ ].

- **Item-by-item position registers**

A different way to make a position record is by entering values element by element, depending on the coordinates and orientation of the point. To insert an item-by-item position record in a program, press from the programming screen.

NEXT→ F1: [INST] → 1-Registers→ Select an option at your convenience → 3-PR[i, j]

Like the other records, they can be entered directly or indirectly.

- Directly: Here the coordinates or positions of the record are defined directly by the values inside, e.g., if for us:

$$PR[1,2] = 200 \rightarrow \mathit{the\ Y\ coordinate\ of\ } PR[1] = 200 \tag{3.9}$$

- Indirectly: The coordinates or positions of the record are affected by the contents of the records within it, i.e., if for us:

$$PR[R[1], R[2]] = 200, \mathit{but\ } R[1] = 3 \mathit{\ and\ } R[2] = 4 \rightarrow \mathit{the\ value\ is\ stored\ in\ } PR[3] \tag{3.10}$$

- And remembering that R[2]=4 , then:

$$\mathit{The\ position\ } W_3 \mathit{\ de\ } PR[3] = 200 \tag{3.11}$$

To understand the position in coordinates and orientation for this type of records according to equation 3.11, there is the following table.

Table 3.2 Definition of coordinates and orientations in position registers

	Coordinates			**Orientations**		
Indicators	**PR[… ,1]**	**PR[… ,2]**	**PR[… ,3]**	**PR[… ,4]**	**PR[… ,5]**	**PR[… ,6]**
PR[1, …]	X_1	Y_1	Z_1	W_1	P_1	R_1
PR[2, …]	X_2	Y_2	Z_2	W_2	P_2	R_2
PR[n, …]	X_n	Y_n	Z_n	W_n	P_n	R_n

Source: (FANUC Robotics Ibérica S. L., 2018)

When we call a position register element by element from an instruction in a program, we are given a choice of arithmetic operations and some statements to obtain a result and store it in the same register or a different one. The way to use these tools follows the format:

$$PR[\,i,j\,] = [value] \quad o \quad PR[\,i,j\,] = [value]\ [operator]\ [value] \tag{3.12}$$

Where the following options can be chosen:

- Operations
  - Addition " + "
  - Subtraction " - "
  - Multiplication " * "
  - Division " / "
  - Integer division " DIV "
  - Remainder of a division " MOD "
- Value
  - Register R[n]
  - Constant
  - I/O
  - TIMER […]
  - Etc.

Note: If you made a position record of a single element, that does not mean that the position record will be completely defined, so, if you seek to define that record completely, you will have to perform the same operation now with each of the missing elements. At the end, you can go to DATA→ F1:[TYPE] → 2-Position Reg and identify if the record is complete (this can be identified if the records you save are equal to R).

- **String registers**

There are different reasons why we use string registers, for example, they are used to generate robot states during a program or to write modular code. In other words, they are used as variables to call programs, among many other options. The robot controller contains up to 25 string registers and they are located in:

DATA→ F1:[TYPE] → 3-String Registers

To insert a string record into a program, press from the programming screen.

NEXT→ F1: [INST] → 1-Registers→ Seleccione una opción a su conveniencia → 4-SR[ ]

Now, when we use a string register as an instruction in a program, we must place the identification and the value it will have. This value can only be obtained from other registers and not from constants (this is done from the DATA address). This can be done directly or indirectly as you choose:

- Directly: The value we give to the string register is stored directly in the specified register and only works through other registers, e.g., if for us:

$$SR[1] = R[1] \;\rightarrow\; but\; R[1] = 2 \;\rightarrow\; the\; final\; value\; is\; SR[1] = 2 \qquad (3.13)$$

O

$$SR[1] = SR[2] \;\rightarrow\; he\; current\; value\; of\; SR[2]\; is\; stored\; as\; SR[1]. \qquad (3.14)$$

- Indirectly: Here, the value we give to the string register is stored according to the content of of the normal register. For example, if for us

$$SR[1] = SR[R[1]] \;\rightarrow\; but\; R[1] = 2 \;\rightarrow\; SR[R[1]]\; takes\; the\; value\; of\; SR[2] \qquad (3.15)$$

If SR[2]=3, therefore, the result would be

$$SR[1] = 3 \qquad (3.16)$$

On the other hand, if:

$$\mathrm{SR[R[1]] = SR[1] \rightarrow\; but\; R[1] = 2 \;\rightarrow\; SR[R[1]]\; takes\; only\; the\; place\; of\, SR[2].} \qquad (3.17)$$

Assuming that SR[1]=4, therefore, the final result would be

$$SR[2] = 4 \qquad (3.18)$$

When we call a string register from an instruction, we are given a choice of arithmetic operations and some statements to obtain a result and store it in that register or a different register. The way to use these tools follows the following format:

$$SR[n] = [value]\; or\; SR[n] = [value]\; [operator]\; [value]. \qquad (3.19)$$

Where you can perform calculations or operations with the following elements:

.

- Operations
  - Addition " + "
  - Subtraction " - "
  - Multiplication " * "
  - Division " / "
  - Integer division " DIV "
  - Remainder of a division " MOD "

- Value
  - SR[ ] string register.
  - Register R[ ].
  - Argument register AR[ ].

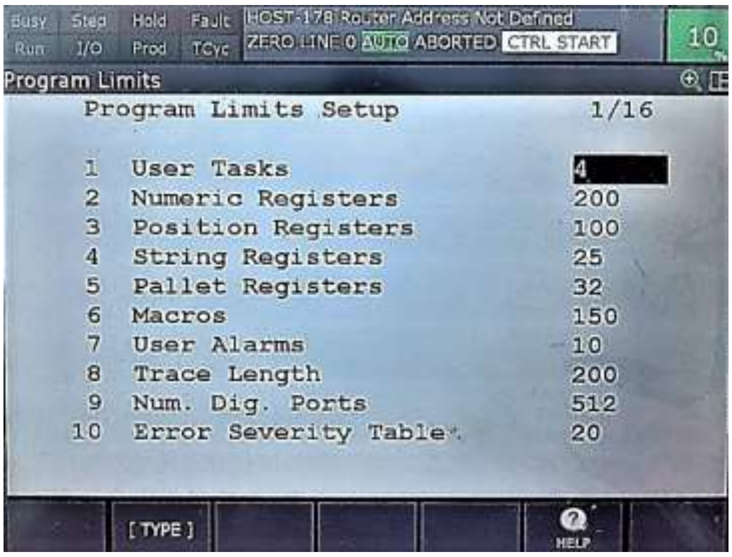
- **Configuration of the number of registers in the TP**

Depending on the number of registers we need to access, the amount can be modified or adapted from a controlled start. To do this, start the robot with the TP in its ON state and press and hold PREV+NEXT while powering up, this will cause us to access the BMON memory.

Once there, choose option 3. Controlled Start and wait for the screen to load completely (see Figure 3.14). After that, go to the following address and manually manipulate the parameters you need:

MENU→ 0-NEXT → 1-PROGRAM SETUP

Figure 3.14 Configuration screen for programming limits



```
HOST-178 Router Address Not Defined
ZERO LINE 0 AUTO ABORTED CTRL START
Program Limits
     Program Limits Setup             1/16

     1  User Tasks                    4
     2  Numeric Registers             200
     3  Position Registers            100
     4  String Registers              25
     5  Pallet Registers              32
     6  Macros                        150
     7  User Alarms                   10
     8  Trace Length                  200
     9  Num. Dig. Ports               512
    10  Error Severity Table          20
```

Source: Elaborated by the work team

To terminate and exit the controlled start, whether or not you have manipulated the programming parameters, press the FCTN→ 1-START (COLD) key and wait (approx. 30 sec) until the PT start screen appears and you can continue normally.

3.2.3.2 Input and output instructions

If you want to use inputs and outputs by means of instructions in a program, the easiest way is to go to the address:

F1: [INST]→ 2--I/O → Select some of the options below

Output instructions

- Digital outputs "DO[n]" or robot outputs "RO[n]".

$$\boldsymbol{DO}[\boldsymbol{n}] = [\boldsymbol{value}] \quad \boldsymbol{o} \quad \boldsymbol{RO}[\boldsymbol{n}] = [\boldsymbol{value}] \tag{3.20}$$

Where the value can be:

1. On/Off
2. Press → Time in seconds between 0.1 to 25.5 s
3. R[ ] → Take values from 0 to 1, where 0 is off and any number followed by 1 is On.
4. (...) → Additional options

- Group outputs “GO[n]”

$$\mathrm{GO}[\mathrm{n}] = [\mathrm{value}] \tag{3.21}$$

Where the value can be:

1. A constant
2. R[ ] → Take decimal values limited to "2n" for n grouped outputs.
3. AR[ ] → Take decimal values limited to "2n" for n grouped outputs.
4. (...) → Additional options

- Analog outputs "AO[n]".

$$\boldsymbol{AO}[\boldsymbol{n}] = [\boldsymbol{value}] \tag{3.22}$$

Where the value can be:

1. A constant
2. R[ ] → Takes values between 0 and 2000 where, 2000 is the maximum voltage at the output.
3. AR[ ] → Takes values between 0 and 2000 where, 2000 is the maximum voltage at the output.
4. (...) → Additional options

Input instructions

The entries are made by means of a register where the following characteristics must be followed:

- R[ ]=DI[ ] → The register will mark 1 (for digital inputs ON) and 0 (for digital inputs OFF).
- R[ ]=RI[ ] → The register will mark 1 (for digital inputs ON) and 0 (for digital inputs OFF).
- R[ ]=GI[ ] → The register will have a decimal value in binary code obtained from a group of digital outputs. digital outputs.
- R[ ]=AI[ ] → The register will have a value between 0 and 2000 depending on the value of the voltage applied to the analog input AI[ ]. applied to the analog input AI[n].

3.2.3.3 Unconditional jump instructions

- **LBL[n] and JMP/LBL [n] instructions**

This instruction is divided into two parts, LBL and JMP LBL. The first is a tag or mark that serves as the destination of the jump. On the other hand, JMP LBL allows conditional and unconditional jumping depending on when they are used or where the tag is located in a program. To insert a tag press:

F1:[INST]→ 5-JMP/LBL → 1-JMP LBL[ ] / 2-LBL[ ]

An example of unconditional jump with labels, is through a loop as in the one presented in code 3.12. There LBL [n] is placed at the beginning of a cycle and JMP LBL [n] where it ends, this will cause the actions that are within them to repeat indefinitely making jumps between them.

```
1: LBL[1]
2: J P[1] 100% FINE
3: J P[2] 100% FINE
4: J P[3] 100% FINE
5: JMP LBL[1]
```

Code 3.12: Unconditional jump with labels

- **CALL instructions**

The CALL instruction allows you to execute an external program by exiting the main program and returning when finished, or you can terminate the current program. This instruction is largely used as a complement with other condition instructions. To use it, press :

F1:[INST]→ 6-CALL→ 1-CALL program / 2-END

Depending on the position in which it is placed, 2-END can make a conditional or unconditional jump to the end of the program, that is to say, when we place this instruction automatically the program that is being executed will stop or terminate.

On the other hand, 1-CALL program, when called, will proceed to execute the program that was called, pausing momentarily the main program. Once it is finished with that subtask, it will return to the CALL instruction and continue with the main execution. Depending on the type of program you wish to call, you will be given the following options:

- F1: PROGRAM = Calls another TP program in the robot database.
- F2: MACRO = Calls only the programs that have been configured as MACRO.
- F3: COLLECT = You can call a program from your Karel programming.
- F4: INDIRECT = Calls a program from an SR[n] string register.
- F5: STRINGS = Calls a program from the script, i.e., if you know the name of the program you need, just type it and the robot will look it up automatically.
- NEXT = It works as a search engine where we are allowed to write the name of the program we need according to the previous options.

3.2.3.4 Conditional jump instructions

These instructions depend on whether the conditions set by them are correct or not, to perform a programming line jump during the execution of a program. The address for them is:

F1:[INST]→ 3-IF/SELECT

- **IF instructions**

Depending on whether the conditions are true, the IF statement will make a jump to an external program or to a label within the main program, whichever the case, the statement is operated following the format:

$$\boldsymbol{IF}\ [\,\boldsymbol{Value\ 1}\,][\,\boldsymbol{Operator}\,][\,\boldsymbol{Value\ 2}\,],[\,\boldsymbol{Jump}\,] \qquad (3.23)$$

Where:

- Value 1
  - Register R[n]
  - Digital I/O DI[n]/DO[n]
  - Robot I/O RI[n]/RO[n]
  - Group I/O GI[n]/GO[n]

- Operator
  - Equal " = "
  - Different from " <> "
  - Less than "<"
  - Less than or equal to "<= "
  - Greater than " > "
  - Greater than or equal to" >= "

- Value 2
  - Register R[n]
  - Digital I/O DI[n]/DO[n]
  - Robot I/O RI[n]/RO[n]
  - Group I/O GI[n]/GO[n]

- Jump
  - JMP LBL[ ]
  - CALL program
  - CALL program ()
  - AND
  - OR

An example of the application of the IF instruction can be seen in code 3.13, here we show a program with an IF condition that works with registers, which tells us that as long as the register R[1] is less than or equal to 5, a loop will be generated. This program will continue looping until the condition of the instruction is no longer fulfilled, in this case, when R[1] takes a value greater than 5. Then the same instruction, if it is not fulfilled, will read the programming line that precedes it, terminating the execution of the program.

```
1: LBL[1]
2: J P[1] 100% FINE
3: J P[2] 100% FINE
4: J P[3] 100% FINE
5: J P[4] 100% FINE
6: J P[5] 100% FINE
7: R[1]=R[1]+1
8: IF R[1]<=5, JMP LBL[1]
9: END
```

Code 3.13: Application of the IF instruction

- **SELECT instructions**

The SELECT statement depends on a condition for one or more jumps to be performed, resulting from the value of a register where all the values inside are checked. To place this instruction press :

F1:[INST]→ 3-IF/SELECT→ 8- -next page- - → 4-SELECT R[ ]=…

The format that this instruction manipulates is as follows:

$$\begin{aligned}\boldsymbol{SELECT\ R[n]} &= [\boldsymbol{Value}], [\boldsymbol{Jump}]\\ &= [\boldsymbol{Value}], [\boldsymbol{Jump}]\\ &= [\boldsymbol{Value}], [\boldsymbol{Jump}]\\ &= \boldsymbol{ELSE}\ [\boldsymbol{Jump}]\end{aligned} \tag{3.24}$$

Where:

- n = A whole number

- Value
  - R[n]
  - Constant
  - AR[n]
  - SR[n]

- Jump
  - JMP LBL [ ]
  - CALL program
  - CALL program (con parámetro adicional)

An example of the application of the Select instruction is in Code 1.14, where it depends entirely on the value of a register to determine which subprogram will be executed.

```
 4: J P[3] 100% FINE
 5: J P[4] 100% FINE
 6: J P[5] 100% FINE
 7: SELECT R[1]=1 CALL PROGRAM_1
 8:          =2 CALL PROGRAM_2
 9:          =3 CALL PROGRAM_3
10:          ELSE, JMP LBL[1]
11: END
```

Code 3.14: Application of the SELECT instruction

3.2.3.5 Wait instructions

This type of instruction delays the execution of a program depending on the type of WAIT we are using, there are time WAITs and condition WAITs. To find them in the programming screen follow the address:

F1:[INST]→ 4-WAIT

- **Instructions “Wait” for time**

Time WAITs delay the execution of a program by a specific time. The time will always be taken in seconds, and you can choose a value between 0.00s as minimum and 327.67s as maximum. Its working format is:

$$WAIT\ [Time] \tag{3.25}$$

Where the time can be entered:

- Directly (Seconds): The time is placed directly taking values between 0 and 327.67s.
- Indirectly(R[n]): The time is placed by means of a register, here the time is equivalent to the content of that register.

- **Instructions “Wait” for condition**

This instruction delays the execution of a program until the declared conditions are true or until a TIMEOUT time period has elapsed. Its working format is as follows:

$$\boldsymbol{WAIT}\ [\,\boldsymbol{Value}][\,\boldsymbol{Operator}\,][\boldsymbol{Value\ 2}][\boldsymbol{Time}] \tag{3.26}$$

Where:

- Value and value 2 Register R[n]
  - Digital I/O and robot DO/DI[n]/DO[n]
  - Constant (only for Value 2)
  - ON/OFF (only for Value 2)

- Operador
  - Equal “ = “
  - Not equal to “ <> “
  - Etc.

- Time
  - <Forever> → Wait until the condition is true.
  - Timeout-LBL[ ] → Wait for a specified time until the condition is true, if the condition is not met after that time, it will go directly to the declared label.
  - AND /OR → Increments the conditions of the instruction.

Note: You can change the Timeout time from a program, to do this press F1:[INST] → 7-Miscellaneous → 7-Parametername → 1-$...=... → F4:[CHOICE] → 2-$WAITTMOUT→ again F4:[CHOICE] → 2-Constant and set the new value.

If you want to change the value from variables press: MENU→0-NEXT→6-SYSTEM→2-VARIABLES → $WAITTMOUT and change the value.

3.2.3.6 Miscellaneous instructions

The miscellaneous instructions are instructions that function to control the execution of a program, particularly focused on production, here we will find tools such as:

- **Ualm[ ]**

The user alarm is an instruction where the user has the control to place the robot in alarm state, causing the robot to stop and, in addition, the message appears on the TP as if it were a common warning, to continue with the execution of the program it is necessary to press RESET and re-execute the program. This version contains up to 10 spaces to configure alarms, as well as allowing a maximum of 29 characters for each alarm. To use this instruction press:

F1: [INST] → 8--next page-- → 1-Miscellaneous → 2-UALM [ ]

This will display the following format

$$\boldsymbol{UALM} = [\,\boldsymbol{value}\,] \tag{3.27}$$

Where the value can be

- Direct = Constants or a choice through a list by pressing F5.
- Indirect = Through a record.

Note: The content of the user warnings can be changed from MENU→0-NEXT→6-SYSTEM→2-VARIABLES → $UALRM_MSG and changing the value.

Or you can press the SET UP → F1:[TYPE] → 9-User Alarm key.

- **Timer [ ]**

Timers are instructions that count the time in seconds of a program, task or subtask. The instruction can start, stop or reset the timer and there can be correlation between 2 or more programs, that is, we can start a timer in a main program and stop or reset it in another one. This version allows to use up to 20 timers and to use them you must press:

F1:[INST] → 8--next page-- → 1-Miscellaneous → 3-TIMER[ ]

This will display the following format:

$$\boldsymbol{TIMER}\,[\,\boldsymbol{value}\,] = [\,\boldsymbol{option}\,] \tag{3.28}$$

Where

- Value = Constants/ A choice through a list by pressing F5/ A register.
- Option= START/ STOP/ RESET/ other options.

Note: To display the list of timers, as well as their current count, you can press from your PT the STATUS→F1:[TYPE]→7-Prg Timer key.

- **Override [ ]**

This instruction allows you to modify the speed of the movements that the robot will make during the program. This action will also be reflected from the TP in the upper right corner where you can see the change in the speed percentage. To use this instruction click:

F1:[INST] → 8--next page-- → 1-Miscellaneous → 4-OVERRIDE

This will display the following format

$$\boldsymbol{OVERRIDE} \;= [\,\boldsymbol{X}\,] \tag{3.29}$$

Where X equals Speed override or A constant value between 0 and 100%.

- **Remark**

This instruction allows the user to place comments or notes anywhere in a program, without having any impact on the program. When the note is placed, the message will be highlighted and will contain an exclamation mark at the beginning, as in the following example:

![Comment containing between 1 to 32 characters]

- **Message**

Message is an instruction that allows to display a message on the main screen of the TP when it is activated when executing a program. The message is specified by the user and must contain between 1 to 23 characters, if there are no characters inside the instruction, it will be activated anyway, but the message will appear empty. The format that follows is:

MESSAGE [Message between 1 and 23 characters]

NOTE: When the MESSAGE instruction is activated, the message will occupy the entire screen of the TP, leaving the programming screen completely and will remain that way, even if the program has finished executing. If you wish to return to the programming screen at the end of or during program execution, you can press the EDIT key on the TP, or you can press the SELECT + ENTER key.

- **Parameter name [ ]**

This instruction modifies the current value of some system variables, specifically variables whose value depends on a number. To use this instruction press:

F1:[INST] → 8--next page-- → 1-Miscellaneous → 7-Parameter

There are two types of formats that can be used here:

$$\$[\boldsymbol{variable}] = [\boldsymbol{value}]\ \boldsymbol{or}\ [\boldsymbol{value}] = \$[\boldsymbol{variable}]. \quad (3.30)$$

Where

- Variable = Depends on the system and the choice you require.
- Valor= R[ ]/Constant/AR[ ]/PR[ ](For the first format).
  R[ ]/PR[ ] (For the second format).

Note: For variables using boolean values, 1 will be taken as a true value and 0 as a false value.

- **MAX_SPEED**

It declares or controls the maximum speed depending on the movement of the robot, these can be:

- JOINT_MAX_SPEED[…]=… To control the movements of each joint.
- LINEAR_MAX_SPEED=[…] To control the linear movements.

- **Multi-Ing Remark**

Unlike the Remark instruction, this instruction allows you to place much larger comments or notes in the range of 1 to 240 characters. Likewise, regardless of the content of the comment, it will not affect the execution of the program, however, when writing a larger comment, more lines of programming will be required.

```
1: [Comment containing between 1 to 240 characters]
2: [            Continuation of the commentary          ]
3: …                                                     .                                   .
```

3.2.3.7 Offsets and Frames instructions

These instructions contain positional information, where, depending on the type of instruction used, is the content that it will have or can be modified:

- Offsets: Contains information measured in distance or degrees, for a displacement.
- Frames: Contains information of a frame to use or of the frame used in some register position.

- **OFFSET Condition**

The offset condition is a reference instruction that only works with position registers, from which it obtains the offset information used in that register when the OFFSET command is activated. If a UFRAME is used as an item, when the OFFSET command is activated, the information that will be received from the position register will use that User Frame and not from its offset. You can use this instruction by pressing :

F1:[INST] → 8—next page-- → 4-Offset/Frames→ 1-OFFSET CONDITION

This will display the following format

$$\boldsymbol{OFFSET\ CONDITION\ PR}[\ldots], \boldsymbol{item} \quad (3.31)$$

Where:

- PR[...]= Can be entered directly or indirectly via a register. Or, a mask group can be used, if the system is configured for it.
- Item = Can be a UFRAME which is used when the OFFSET command is activated or it can be empty.

– **UFRAME_NUM= / UTOOL_NUM=**

This instruction allows to set directly in the programming, the number of user or tool frame being used or to be used. If the frame number is equal to zero, the default coordinate system will be used (WORLD coordinates on the base for the user frame and WORD coordinates on the robot plate for the user tool). You can use this instruction by pressing :

F1:[INST] → 8—next page-- → 3-Offset/Frames→ 2- UFRAME_NUM=/3 UTOOL_NUM=

This will display the following format

$$\boldsymbol{UFRAME\_NUM} = [\ldots] \quad \boldsymbol{o} \quad \boldsymbol{UTOOL\_NUM} = [\ldots] \qquad (3.32)$$

Where:

– [...]= It can be a constant, R[ ] or an AR[ ]. Whatever the case, you must select a number between 0 and 9 (for User frame) or 0 to 10 (for User tool), where 0 will take the default coordinate system.

3.2.3.8 Program control and multiple control instructions

As mentioned in the name, these are instructions that directly interfere with the execution of a program as soon as they are activated. In total there are five of the most used ones and they will be defined below.

– **PAUSE**

It suspends the execution of a program, due to the following characteristics:

- The timers used continue to be incremented.
- Any motion that has started before the instruction will continue until completed.
- All instructions in progress continue until completed, except for CALL instructions, which will be suspended until the program resumes.

You can use this instruction by pressing:

F1:[INST] → 8--next page-- → 6-Program control→ 1-PAUSE

– **ABORT**

When activated, the abort instruction terminates a program in its entirety and cancels any ongoing or paused motion. There is no way to resume a program that has been aborted, so it will have to be reset. You can use this instruction by pressing:

F1:[INST] → 8--next page-- → 6-Program control→ 2-ABORT

– **ERROR_PROG**

This instruction stores the name of the described program in the system variable $ERROR_PROG, where its use depends on the configuration in the system. You can use this instruction by pressing :

F1:[INST] → 8--next page-- →6-Program control→ 3-ERROR_PROG

– **RESUME_PROG**

This is an instruction that resumes a previously paused program. When activated, the instruction stores the name of the described program in the system variable $RESUME_PROG, where its use depends on the configuration in the system and the installation of the J924 element. You can use this instruction by pressing :

F1:[INST] → 8--next page-- → 6-Program control→ 4-RESUME_PROG

- **RUN**

Depending on the system configuration, the multiple control has only the "RUN" instruction, which allows to perform more than one task at the same time. It starts the execution of the specified program when the instruction is called and follows the execution of the main program, in other words, it allows to launch one or more programs in the background without stopping the execution of the original program as long as different mask groups are used. You can use this instruction by pressing:

F1:[INST] → 8--next page-- → 5-Multiple control→ 1-RUN

3.2.3.9 Condition Instructions

These instructions allow to specify conditions, to develop movements, to make small jumps within a program, etc. All this, following the specifications declared from the beginning, among them are:

- **SKIP**

It sets the condition directly or through operators and depends on whether the answer is true or not, to perform a jump over an additional move instruction,. This instruction follows the following characteristics:

- If the instruction detects a signal, the additional move instruction in the programming line below it is not considered and a jump to the next line is performed.
- If the instruction does not detect a signal, the program continues reading the next programming line and performs the additional move instruction.
- This condition only works if the programming line below it contains an additional move instruction, otherwise the program will avoid the execution of this instruction and continue executing the program normally.
-

In which, the format used is as follows:

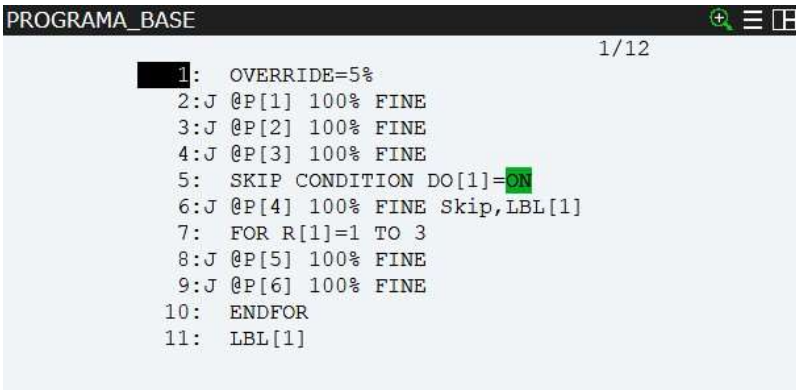
$$\boldsymbol{SKIP\ CONDITION}\ [\boldsymbol{Variable}][\boldsymbol{Operator}][\boldsymbol{Value}] \tag{3.33}$$

Where:

- Variable = They can be records or system variables.
- Operator = These are common operators: equal to, different from, greater than, less than, etc.
- Value = They can be records or constants.

Example: Figure 3.15 shows an example of the use of the SKIP instruction, where if the condition in line 5 is true, the reading of the program will continue in line 7, making a jump. However, if the condition is false, line 6 is read. Since this line contains an additional movement instruction (in this case a label), the jump will be made directly to line 11 and the reading will continue as normal.

Figure 3.15 Use of SKIP and FOR/ENDFOR instruction



```
PROGRAMA_BASE
                                          1/12
     1:  OVERRIDE=5%
     2:J @P[1] 100% FINE
     3:J @P[2] 100% FINE
     4:J @P[3] 100% FINE
     5:  SKIP CONDITION DO[1]=ON
     6:J @P[4] 100% FINE Skip,LBL[1]
     7:  FOR R[1]=1 TO 3
     8:J @P[5] 100% FINE
     9:J @P[6] 100% FINE
    10:  ENDFOR
    11:  LBL[1]
```

Source: Image obtained from Roboguide software

– **FOR/ENDFOR**

The FOR statement allows us to loop through a program as long as a numerical condition is met with registers or constants, while ENDFOR terminates the statement (see Figure 3.15). You can use this instruction by pressing:

F1:[INST] → 8--next page-- → 8--next page--→ 1-FOR/ENDFOR

The format used is as follows:

$$\begin{aligned}&\boldsymbol{FOR\ R[n] = [\ Value\ 1\ ]\ TO\ \ [Value\ 2]}\\&\boldsymbol{ENDFOR}\end{aligned} \tag{3.34}$$

Where:

- R[n]= Is a record number from 1 to 200 (available records).
- Value1 = Can be registers or constants. The initial value of the register is defined here, where the value zero refers to an additional execution of the created loop.
- Value2 = Can be registers or constants. Here you define the final value that the register will have and therefore, the number of times the loop will be executed.

– **UFRAME [ ]= / UTOOL [ ]=**

This instruction allows the user frame or tool to be defined by the contents of a position register. You can use this instruction by pressing :

F1:[INST] → 8—next page-- → 4-Offset/Frames→ 4- UFRAME[ ] /5-UTOOL_NUM[ ]

The format to be used is as follows:

$$\boldsymbol{UFRAME}[\ldots] = \cdots \quad \boldsymbol{o} \quad \boldsymbol{UTOOL}[\ldots] = \cdots \tag{3.35}$$

Where:

- [...]= It can be a direct or indirect value between 0 to 9 (for user frames) and 0 to 10 (for user tools) where, if the value is equal to zero, the default values will be taken.
- ... = A position register is placed.

– **PAYLOAD**

Specifies the payload condition to be used in some section of the program or in the whole program, where the payload number is placed through a list, directly or indirectly. You can use this instruction by pressing:

F1:[INST] → 8--next page-- → 3-Payload

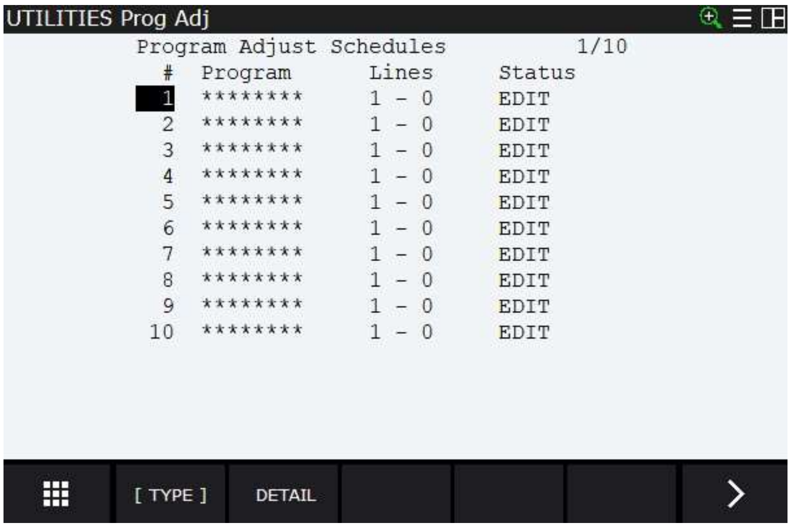
3.2.4 Advanced functions

3.2.4.1 Prog Adjust

This tool allows us to adjust the information of a program (modifying specific points) while the robot is executing them, and they are stored in cards. This version of the controller can contain up to ten program setting cards (see Figure 3.16). The address to use this function is:

MENÚ → 1-UTILITIES → 5-PROG ADJUST

Figure 3.16 Screen of the available schedules to adjust programs



```
Program Adjust Schedules          1/10
  #  Program      Lines     Status
  1  ********     1 - 0     EDIT
  2  ********     1 - 0     EDIT
  3  ********     1 - 0     EDIT
  4  ********     1 - 0     EDIT
  5  ********     1 - 0     EDIT
  6  ********     1 - 0     EDIT
  7  ********     1 - 0     EDIT
  8  ********     1 - 0     EDIT
  9  ********     1 - 0     EDIT
 10  ********     1 - 0     EDIT
```

Source: Image obtained from Roboguide software

Note: If you already have a program open in the TP, pressing the above address will not take you to the list of programs, but to the screen for setting that program. You can return to the screen in figure 3.14 if you wish, by pressing PREV.

In the screen shown in Figure 3.16, you can see that the cards consist mainly of the name of the program, the programming lines where the points to be set are located and the status (enabled and not enabled), in addition to the following options that allow you to modify, delete or copy cards:

– F2: DETAIL= Allows you to set the adjustment parameters for a program.
– NEXT+F1: COPY= Copies the parameters from one schedule to another. To use it, position the cursor on the schedule you want to copy and when it says "Enter schedule number to copy:" enter the number of the schedule where the new information will be sent and press ENTER.
– NEXT+F2: CLR_ADJ=Delete all the position and velocity settings of the cell, but do not delete the name of the cell, the relative offset, nor the start and end line you are working with.
– NEXT+F3: CLR_ALL=Delete the entire plate.

Subsequently, to make the settings of a program, place the cursor on one of the available cells and press DETAIL, where you will obtain the screen shown in Figure 3.17. The different available options offered by the TP for adjusting parameters such as units of measurement, etc., are shown:

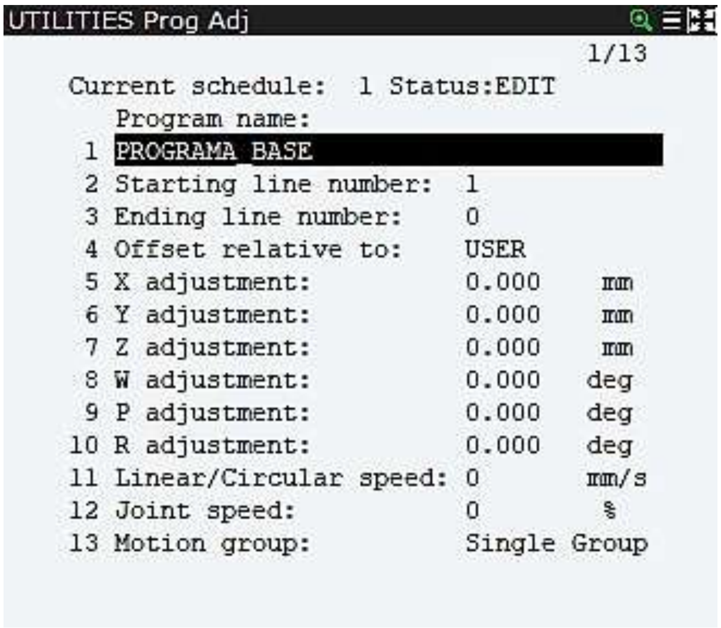
– F2: UNITS = Changes the units of measure mm to inches (in) or vice versa.
– F3: SCHED= Allows the user to switch from one grid to another.
– F4: ENABLE= Activates the settings made.
– F5: DISABLE= Disables the settings made.
– NEXT+F5: SAVE= Saves all data collected up to the moment of pressing it.

Now, you can be guided by the following series of steps to develop an adjustment on a program:

1) Position the TP cursor on item number one and enter the name of the program to be set. To do this you can press the F4:[CHOICE] button and you will get a physical list of the available programs.
2) In "Starting line number:" and "Ending line number:" place the starting and ending line where the points you want to adjust in your program are located.
3) Identify and place in the fourth item the relative offset, i.e. the coordinate system on which the grid will be based to adjust the position of the points of step 2. This can be USER or TOOL, considering that this system you use will be the same as the one used to elaborate your original program.

4) Place the new X, Y, Z, W, P, R. settings, depending on your needs, in their corresponding items. You can change the units by pressing F2: UNITS.
5) If necessary, you can change the speed parameters for circular, linear or joint movements, as well as the movement group. To do so, go to item 11-13 and do so. Note that a value of zero means that there will be no changes in items 11-12 and the "Single Group" motion group is the default (see Figure 3.17).
6) Once the changes are completed, press F4: ENABLE and your settings are done. Similarly, in case you want to revert the settings already made, you can press F5: DISABLE.

Figure 3.17 Program settings screen for a schedule



```
UTILITIES Prog Adj
                                          1/13
   Current schedule:  1 Status:EDIT
      Program name:
    1 PROGRAMA BASE
    2 Starting line number:  1
    3 Ending line number:    0
    4 Offset relative to:    USER
    5 X adjustment:          0.000    mm
    6 Y adjustment:          0.000    mm
    7 Z adjustment:          0.000    mm
    8 W adjustment:          0.000   deg
    9 P adjustment:          0.000   deg
   10 R adjustment:          0.000   deg
   11 Linear/Circular speed: 0       mm/s
   12 Joint speed:           0        %
   13 Motion group:          Single Group
```

Source: Image obtained from Roboguide software

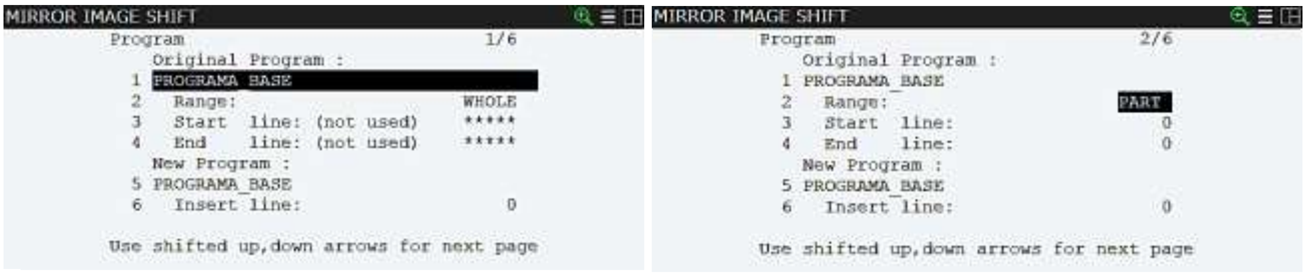
Note: If the setting fails, the TP will show the line number that generated the error and you will have to correct it. Therefore, it is recommended that you disable the card, try to correct the error and re-enable it.

3.2.4.2 Program Shift

Unlike the Prog Adjust, the program shift allows to correct a trajectory, to duplicate only a part of it or to duplicate it completely (see figure 3.18). This type of shift can be done in parallel or with rotation. The address to find this utility is:

MENÚ → 1-UTILITIES → 6-PROGRAM SHIFT

Figure 3.18 Program Shift screen, a) For a whole program b) For a part of a program



```
MIRROR IMAGE SHIFT
   Program                             1/6
      Original Program :
    1 PROGRAMA BASE
    2   Range:                         WHOLE
    3   Start  line: (not used)        *****
    4   End    line: (not used)        *****
      New Program :
    5 PROGRAMA_BASE
    6   Insert line:                       0

   Use shifted up,down arrows for next page
```

a)

```
MIRROR IMAGE SHIFT
   Program                             2/6
      Original Program :
    1 PROGRAMA_BASE
    2   Range:                          PART
    3   Start  line:                       0
    4   End    line:                       0
      New Program :
    5 PROGRAMA_BASE
    6   Insert line:                       0

   Use shifted up,down arrows for next page
```

b)

Source: Image obtained from Roboguide software

The following are the operations that you have to fill in to perform a trajectory displacement (see figure 3.18):

- Original Program: Enter the name of the program from which you will correct or duplicate its trajectory. To do this you can press the F4: [CHOICE] button and you will get a physical list of the available programs.

- Range: Define the working range, WHOLE is for using the whole program path (see figure 3.18a) and PART is for using only a part of it (see figure 3.18b).

a. Start line: Indicate the first line of the path to be corrected or copied (only for the PART range).
b. End line: Indicate the last line of the path to be corrected or copied (Only for the PART range).

- New Program: If you want to make a complete copy or only a part of a path, as well as to develop a new program, press the ENTER key and modify the name. Otherwise, to correct the original path, do not modify this option,

- Insert line: Here insert the number of the program line where the new displaced path will be placed (only works for the PART range to make corrections and not copies).

Once you have the above data, press the SHIFT+UPPER/LOWER ARROW key and depending on the type of scrolling you want, use some combination of keys to record points.

- F3: REFER = to use position records or place the number of some point that is located in the path of the original program.

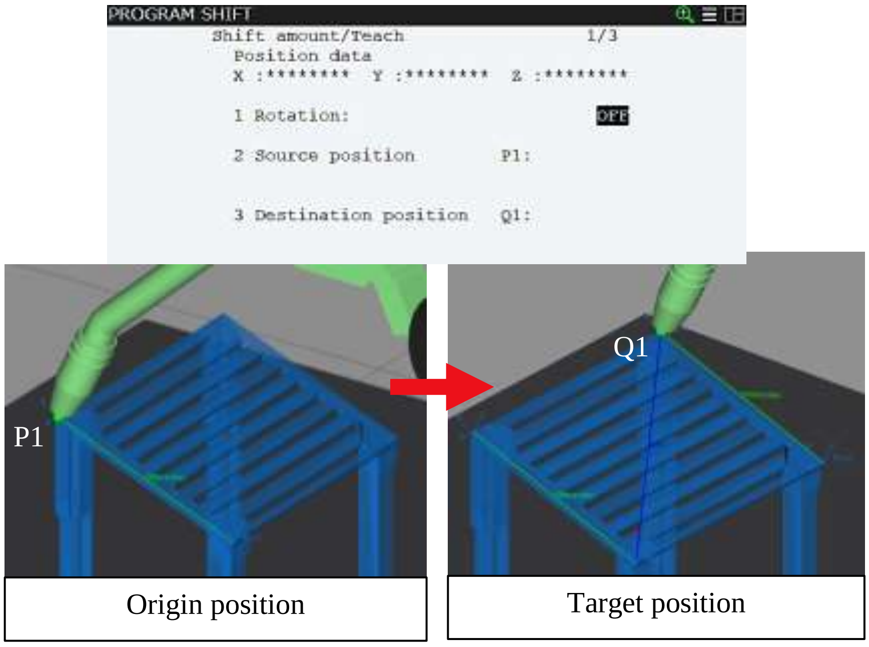
- SHIFT+F5: RECORD= To record a point directly from the robot.

- NEXT+F2: DIRECT =To add the position manually using the default WORLD coordinate system (to return to the previous screen you can press NEXT+TEACH).

For parallel displacements (see Figure 3.19):

1. In the "Rotation" item, leave the default option OFF.
2. In P1 place the point that will serve as a reference for the displacement of the trajectory.
3. In Q1 place the point that will serve as destination and to which our trajectory will move.

Figure 3.19 Displacement of a trajectory without applied rotation



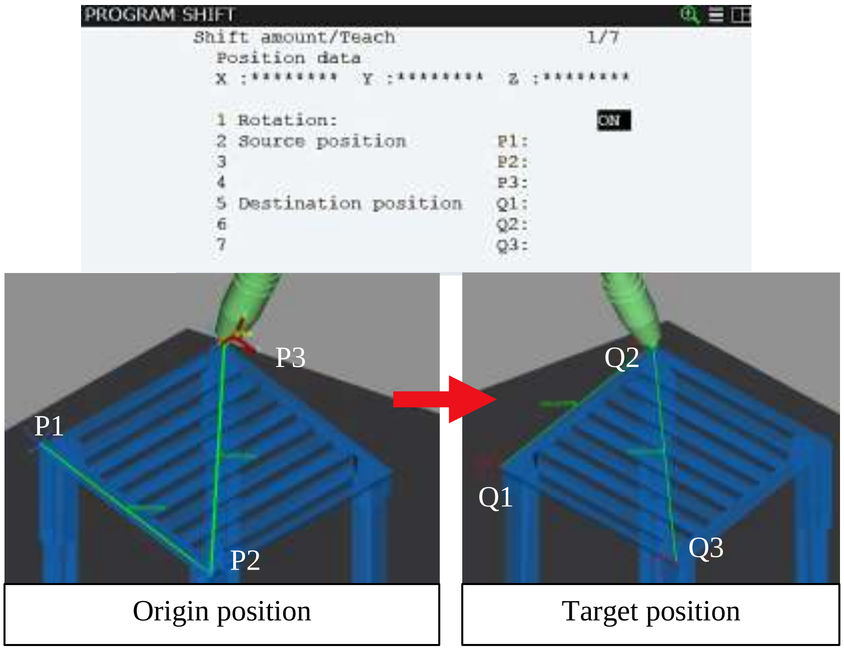
```
PROGRAM SHIFT
  Shift amount/Teach                         1/3
    Position data
     X :********   Y :********   Z :********

    1 Rotation:                             OFF

    2 Source position        P1:

    3 Destination position   Q1:
```

Source: Image obtained from Roboguide software

For displacements with rotation (see Figure 3.20):

1. In the "Rotation" item change the option to ON.
2. In P1-P3, place the points that will serve as reference for the displacement of the trajectory.
3. In Q1-Q3 place the points that will serve as destination and to which our trajectory will move.

Figure 3.20 Displacement of a trajectory applying rotation



```
PROGRAM SHIFT
        Shift amount/Teach                     1/7
          Position data
          X :********   Y :********   Z :********

          1 Rotation:                              ON
          2 Source position          P1:
          3                          P2:
          4                          P3:
          5 Destination position     Q1:
          6                          Q2:
          7                          Q3:
```

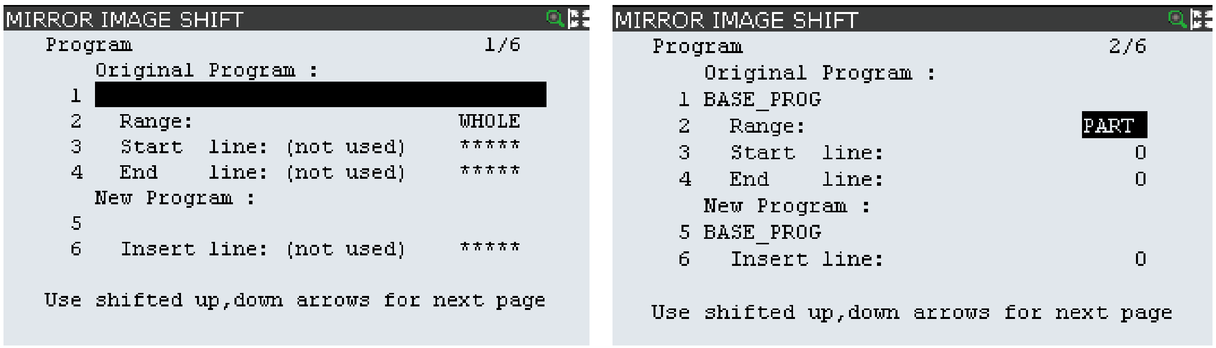
Source: Image obtained from Roboguide software

3.2.4.3 Mirror Image Shift

This utility, in the same way as Program shift, allows to duplicate the whole path of a program or only a part of it, with the difference that the new path generated is symmetrical to the points that were selected in the process. The address to find this utility is:

MENÚ → 1-UTILITIES → 7-MIRROR IMAGE SHIFT

Figure 3.21 Mirror Image Shift screen, a) For a whole program b) For a part of a program



```
MIRROR IMAGE SHIFT
   Program                                1/6
      Original Program :
    1
    2    Range:                         WHOLE
    3    Start   line: (not used)       *****
    4    End     line: (not used)       *****
      New Program :
    5
    6    Insert line: (not used)        *****

   Use shifted up,down arrows for next page
```

a)

```
MIRROR IMAGE SHIFT
   Program                                2/6
      Original Program :
    1 BASE_PROG
    2   Range:                           PART
    3   Start  line:                        0
    4   End    line:                        0
      New Program :
    5 BASE_PROG
    6   Insert line:                        0

   Use shifted up,down arrows for next page
```

b)

Source: Image obtained from Roboguide software

The following are the operations that you have to fill in to perform a path symmetry (see figure 3.21):

– Original Program: Enter the name of the program from which you will correct or duplicate its trajectory. To do this you can press the F4: [CHOICE] button and you will get a physical list of the available programs.

– Range: Define the working range, WHOLE is for using the whole program path (see figure 3.21a) and PART is for using only a part of it (see figure 3.21b).

a. Start line: Indicate the first line of the path to be corrected or copied (only for the PART range).

b. End line: Indicate the last line of the path to be corrected or copied (Only for the PART range).

– New Program: If you want to make a complete copy or only a part of a path, as well as to develop a new program, press the ENTER key and modify the name. Otherwise, to correct the original path, do not modify this option,

– Insert line: Here insert the number of the program line where the new displaced path will be placed (only works for the PART range to make corrections and not copies).

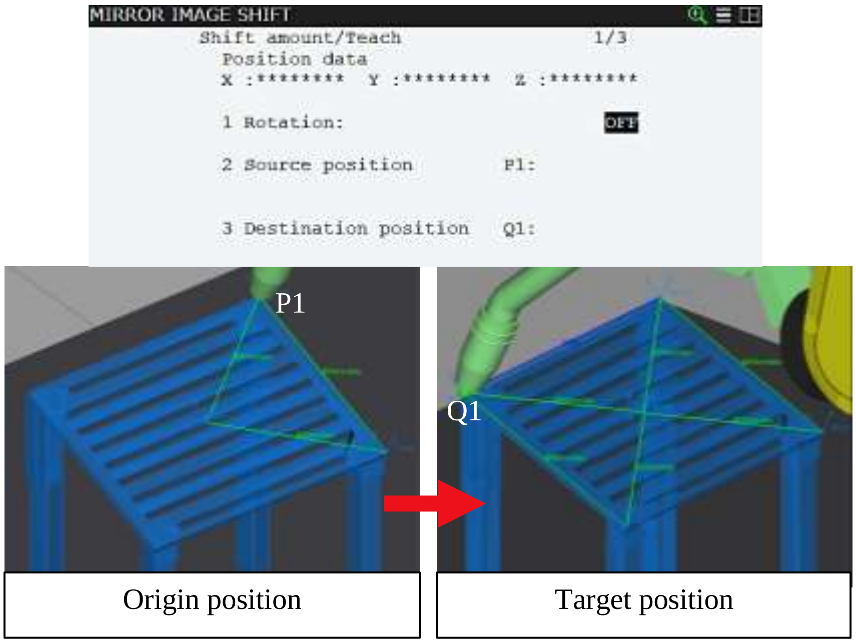
Once you have the above data, press the SHIFT+UPPER/LOWER ARROW key and depending on the type of scrolling you want, use some combination of keys to record points.

– F3: REFER = to use position records or place the number of some point that is located in the path of the original program.
– SHIFT+F5: RECORD= To record a point directly from the robot.
– NEXT+F2: DIRECT =To add the position manually using the default WORLD coordinate system (to return to the previous screen you can press NEXT+TEACH).

For symmetries parallel to the original trajectory (see Figure 3.22):

1. In the item "Rotation" leave the default option OFF.
2. In P1 place the point that will serve as reference for the path symmetry.
3. In Q1 place the point that will serve as the destination where the path symmetry will be created.

Figure 3.22 Parallel displacement of a trajectory



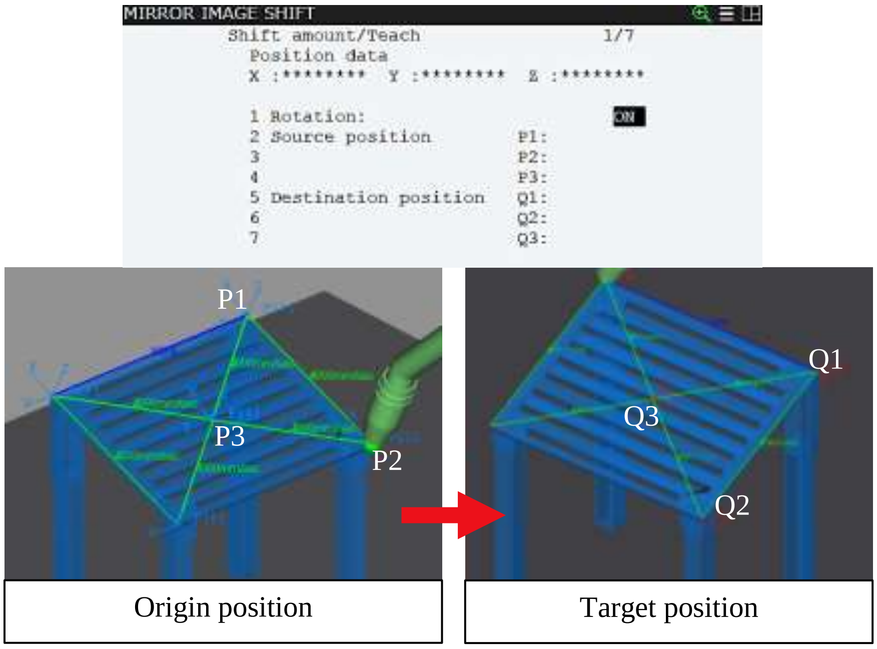
```
MIRROR IMAGE SHIFT
      Shift amount/Teach                 1/3
        Position data
        X :********  Y :********  Z :********

        1 Rotation:                      OFF

        2 Source position       P1:

        3 Destination position  Q1:
```

Source: Image obtained from Roboguide software

For displacements with rotation (see Figure 3.23):

1. In the "Rotation" item change the option to ON.
2. In P1-P3, place the points that will serve as reference to create the path symmetry.
3. In Q1-Q3, place the points that will serve as the destination where the path symmetry will be created.

Figure 3.23 Displacement of a trajectory applying rotation



```
MIRROR IMAGE SHIFT
         Shift amount/Teach                 1/7
           Position data
           X :********  Y :********  Z :********

           1 Rotation:                         ON
           2 Source position          P1:
           3                          P2:
           4                          P3:
           5 Destination position     Q1:
           6                          Q2:
           7                          Q3:
```

Origin position

Target position

Source: Image obtained from Roboguide software

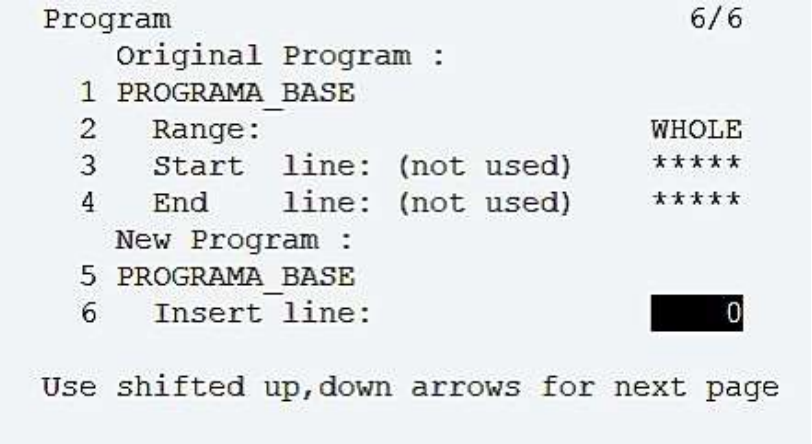
3.2.4.4 Tool offset and User offsets

It is possible to modify the tools and user frames that were used in a program, altering or not, the points that are found within it. The address to find this utility is:

MENÚ → 1-UTILITIES → 8-TOOL OFFSET/ 9-FRAME OFFSETSHIFT

The configuration screen for both cases is as follows (see Figure 3.24):

Figure 3.24 Configuration screen for changing offsets in a program



```
Program                                   6/6
    Original Program :
  1 PROGRAMA_BASE
  2   Range:                             WHOLE
  3   Start  line: (not used)            *****
  4   End    line: (not used)            *****
    New Program :
  5 PROGRAMA_BASE
  6   Insert line:                           0

Use shifted up,down arrows for next page
```

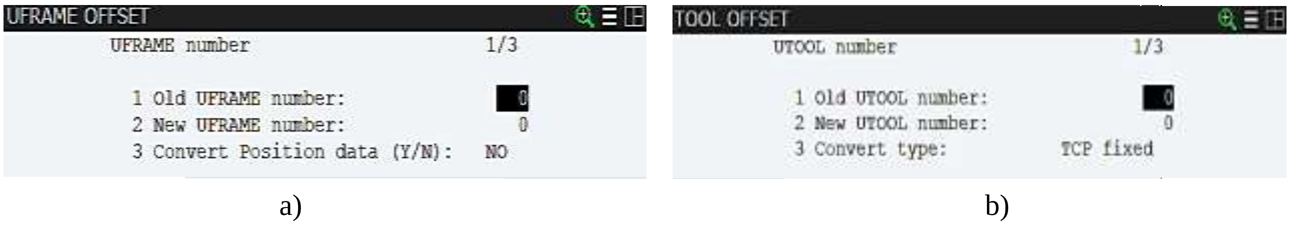
Source: Image obtained from Roboguide software

Where in:

- Original Program: Enter the name of the program.
- Range: Define the working range, WHOLE or PART.
- New Program: If you want to change the offset of the whole program or only a part of it, as well as to develop a new program, press the ENTER key and modify the name. Otherwise, to correct the offset of the original program, do not do it.
- Insert line: Insert the number of the line where the lines with the new offset will be placed.

To conclude the offset change press the SHIFT+UPPER/LOWER ARROW key, this will take you to the next screen (see figure 3.25):

Figure 3.25 Configuration screen for changing offsets in a program UFRAME/UTOOL



```
UFRAME OFFSET
        UFRAME number                   1/3

          1 Old UFRAME number:            0
          2 New UFRAME number:            0
          3 Convert Position data (Y/N):  NO
```

```
TOOL OFFSET
        UTOOL number                    1/3

          1 Old UTOOL number:             0
          2 New UTOOL number:             0
          3 Convert type:          TCP fixed
```

a) b)

Source: Image obtained from Roboguide software

Here you must enter the following data:

- Old UTool/Uframe number = The original user tool or frame number is entered.
- New UTool/Uframe number=New user tool or frame number is set.
- Covert = "TCP Fixed / NO " Changes the UTool or UFrame without changing the program dimensions to preserve the original program paths.
  "Robot fixed / YES" Changes the UTool or UFrame also changing the program dimensions and generating possible collisions.

References

FANUC Robotics Ibérica S. L. (2018). ROBÓTICA PROGRAMACIÓN FANUC. *SEAT*. http://www.infoplc.net/files/descargas/fanuc_robotica/infoPLC_net_ManualFANUCR-J3iB.pdf

Guerrero Ávila, Y., & Llama Leal, M. A. (2017). Contribución al Estudio de la Selección de Parámetros de la Estación de Soldadura Robotizada Fanuc 100iC. *Tecnológico Nacional de México*. https://rinacional.tecnm.mx/handle/TecNM/610

Reyes Cortés, R. (2011). Robótica. Control de Robots Manipuladores - 1ra Edición. *AlfaOmega*, *Primera Ed*, 1–592.

Saltarén Pazmiño, R. J., Azorin Poveda, J. M., Miguel, A. K., & Sabater Navarro, J. M. (2000). Prácticas de Robótica utilizando Matlab. In *Práctica 5: Planificación de Trayectorias* (pp. 1–23). https://nbio.umh.es/libro-practicas-de-robotica-utilizando-matlab/

Sierra Bueno, D. A., & Martínez Angel, R. (2002). Planeación de trayectorias. *Revista UIS Ingenierías, Vol 1*(No. 2), 31–34. https://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/2509

Chapter 4 I/O control and MACRO programs

Abstract

For industrial robotics, the use of inputs and outputs has become something of daily use, so much so that they began to be categorized according to the application to be performed, for example, there are digital, analog, group, user, etc. inputs and outputs. Therefore, understanding how they work, how they are configured and how to control each one of them is vital to perform a more complete programming with the robot. On the other hand, macro-commands can be considered as mini programs, which contain in their programming code, a series of simple instructions with the purpose of executing specific tasks. In addition, each macro can be assigned in a list called Manual Fnct, which is a list that allows to perform specific tasks as a pre-maintenance program, executing this type of programs. As can be seen, the focus of this chapter is on learning the different types of I/O found in the Fanuc LR Mate 200iD robotic system, as well as the macro commands by identifying their basic characteristics.

I/O, Mini program, Assignment

4.1 Introduction

The inputs and outputs (I/O) in robotics, are an ingredient that over time has become basic to learn in any of its branches by its integration to the robot software as can be seen in the work of (Pinto Hernández, 2016). On the one hand, we have that the inputs are the data acquired by the robot after an action and, on the other hand, the outputs are the commands that are sent when the data obtained from the inputs are processed.

In that case, most robotic applications in industry use them and that is why it is important to know them. Starting from that point, the I/O can be classified depending on the data they manipulate, for example, there are digital I/O, which take voltage values at high and low levels, among other classifications. Moving away from the above, there are also macro-commands or macro programs, which are small programs, which include simple instructions to execute them continuously (FANUC Robotics Ibérica S. L., 2018) which makes it interesting for maintenance applications or to be called from a main program.

Chapter 4 aims to provide a technical-basic description of the different I/O found in the Fanuc LR Mate 200iD robotic system, using tools that allow a simpler understanding. In the same way, it is intended to describe the macro commands, how they are used, how they are created and what possible applications they can have.

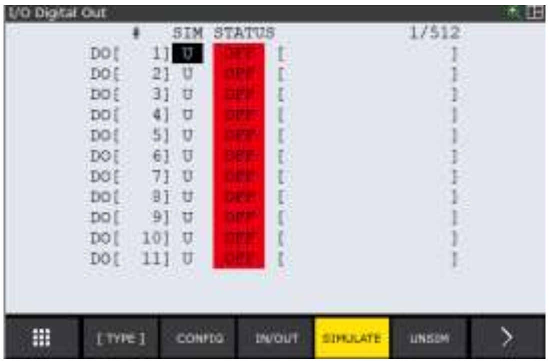
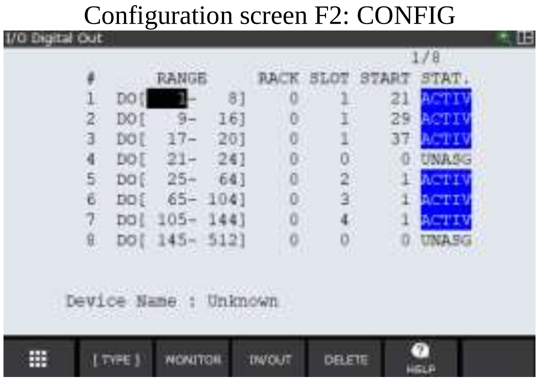
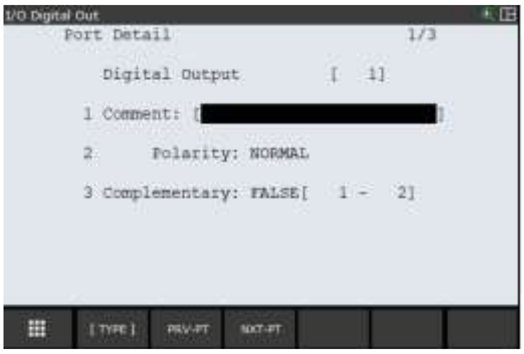
4.2 I/O control

The Fanuc LR Mate 200iD version of the robot contains 6 different types of inputs and outputs, which can be used, assigned, or configured. These I/O are explained in more detail below.

4.2.1 Digital I/O “DI/DO”

These are digital inputs and outputs that can be configured from the TP and controlled 0 and 24V signals. From the monitoring screen for this type of I/O we can see the status of the 512 available, as well as the different options and other screens to perform some configuration (see table 4.1). The location of these can be found in: MENÚ → 5-I/O → 3-DIGITAL

Table 4.1 Configuration screens for digital I/O and their description

Configuration screens for digital I/O	Description
I/O monitor main screen  I/O Digital Out # SIM STATUS 1/512 DO[1] U OFF [ ] DO[2] U OFF [ ] DO[3] U OFF [ ] DO[4] U OFF [ ] DO[5] U OFF [ ] DO[6] U OFF [ ] DO[7] U OFF [ ] DO[8] U OFF [ ] DO[9] U OFF [ ] DO[10] U OFF [ ] DO[11] U OFF [ ] 	– # = Defines the number of I/O to be used or configured. – SIM= Indicates the simulation status of the I/O. – F4:SIMULATE→ I/O is being simulated at software level (without generating physical voltages). – F5:UNSIM→ I/O is forced by generating a physical voltage. – STATUS=Declares the status of each I/O. – F4:ON – F5:OFF
Configuration screen F2: CONFIG  I/O Digital Out 1/8 # RANGE RACK SLOT START STAT. 1 DO[1- 8] 0 1 21 ACTIV 2 DO[9- 16] 0 1 29 ACTIV 3 DO[17- 20] 0 1 37 ACTIV 4 DO[21- 24] 0 0 0 UNASG 5 DO[25- 64] 0 2 1 ACTIV 6 DO[65- 104] 0 3 1 ACTIV 7 DO[105- 144] 0 4 1 ACTIV 8 DO[145- 512] 0 0 0 UNASG Device Name : Unknown 	– RANGE =Shows the number of I/O that are in a signal range. – RACK=The physical location where the I/O board or modules are mounted. – SLOT=The space in the RACK where the I/O module is connected. – START=The port used from the board or I/O module. – STAT=Indicates the status of the digital signal. – ACTIV → The assignment is valid. – UNASG→ The assignment has not been made. – INVAL→ The assignment is invalid. – PEND→ The assignment is valid but not active.
Port details screen NEXT→ DETAIL  I/O Digital Out Port Detail 1/3 Digital Output [1] 1 Comment: [ ] 2 Polarity: NORMAL 3 Complementary: FALSE[1 - 2] 	– Comment: Allows to declare a comment for the I/O. – Polarity: Sets the polarity of the I/O. – F4:INVERSE→ ON=0V and OFF=24V – F5:NORMAL→ ON=24V and OFF=0V – Complementary: Associates two outputs at the same time. – F4: TRUE →For DO[1]=ON, DO[2]=OFF For DO[1]=OFF, DO[2]=ON – F5: FALSE →The outputs are independent. – F4: PRV-PT/F5:NXT-PT→Pass or return to the next I/O.

Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

As you can see, Table 4.1 lists the locations and parameters that make up each configuration screen for the digital I/O, whose condition can be changed. Here you can use the F3:IN/OUT function key to work specifically with inputs or outputs, and note that after any configuration, you must do a cold start, so you must turn the robot controller off and on again for the system to start with the new variables.

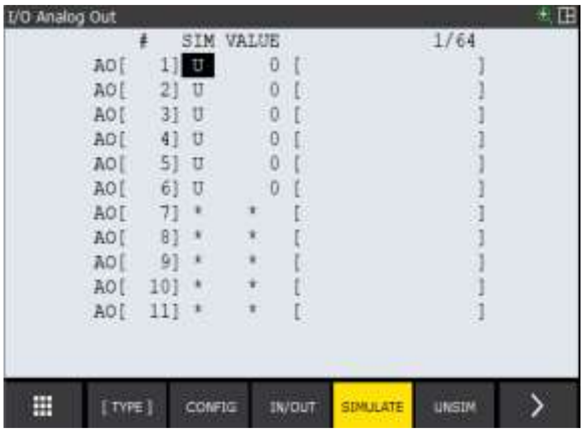
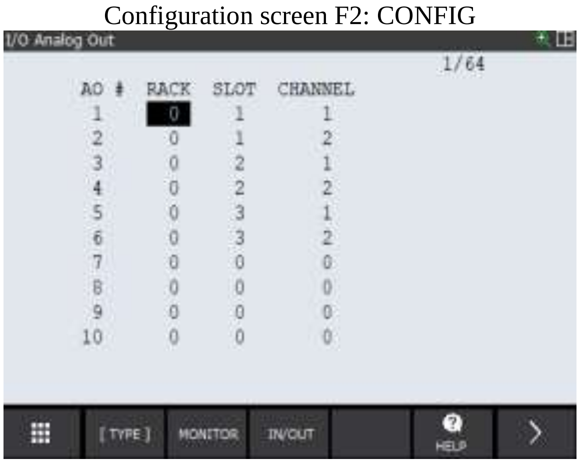
Note: You can extend the total number of inputs and outputs allowed in the system if you wish. To do so, go to the variable $MAX_DIG_PRT (approx. in item 330) and change the value depending on your needs (min 15, max 8192).

You can also perform the above action by a controlled start in MENU → Program Setup, find the variable and change it manually.

4.2.2 Analog I/O "AI/AO"

This type of I/O is used in cases where analog elements (sensors or actuators) are used. The list location for these outputs can be found under: MENU → 5-I/O → 4-ANALOG. (See in table 4.2).

Table 4.2 Configuration screens for analog I/O and their description

<table>
<tr><th>Analog I/O configuration screens</th><th>Description</th></tr>
<tr><td>I/O monitor main screen

I/O Analog Out
SIM VALUE 1/64
AO[1] U 0 [ ]
AO[2] U 0 [ ]
AO[3] U 0 [ ]
AO[4] U 0 [ ]
AO[5] U 0 [ ]
AO[6] U 0 [ ]
AO[7] * * [ ]
AO[8] * * [ ]
AO[9] * * [ ]
AO[10] * * [ ]
AO[11] * * [ ]
 </td><td>– # = Defines the number of I/O to be used or configured.
– SIM= Indicates the simulation status of the I/O.
  – F4:SIMULATE→ I/O is being simulated at software level (without generating physical voltages).
  – F5:UNSIM→ I/O is forced by generating a physical voltage.
– STATUS=Declares the status of each I/O.
  – F4:ON
  – F5:OFF
– F3: IN/OUT → Change inputs to outputs and vice versa.</td></tr>
<tr><td>Configuration screen F2: CONFIG

I/O Analog Out
1/64
AO # RACK SLOT CHANNEL
1 0 1 1
2 0 1 2
3 0 2 1
4 0 2 2
5 0 3 1
6 0 3 2
7 0 0 0
8 0 0 0
9 0 0 0
10 0 0 0
 </td><td>– AO#/AI# = Defines the number of I/O to be used or configured.
– RACK=This is the physical location where the I/O board(s) or module(s) is mounted.
– SLOT=This is the space in the RACK where the I/O module is connected.
– CHANEL= This item is the physical location of the port on the I/O board.

– NEXT+F2: VERIFY → Verify if the assignment is valid.
– F3: IN/OUT → Change inputs to outputs and vice versa.</td></tr>
</table>

Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

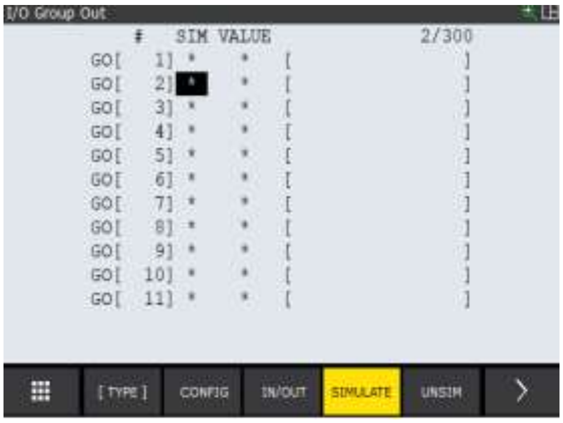
Note: after any configuration, you must do a cold start, so you must turn the robot controller off and on again for the system to start with the new variables.

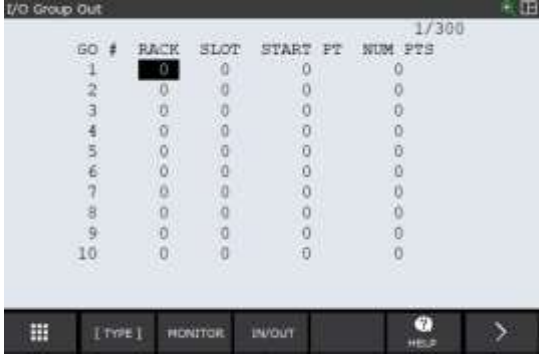
4.2.3 Group I/O "GI/GO"

These I/O are used to activate several input or output signals at once and not sequentially like the others. In addition, the signals are controlled entirely with a decimal encoded binary number format.

The direction to find the list of this type of I/O is by pressing the keys of the teach pendant MENU → 5-I/O → 5-GROUP, to obtain the different screens presented in Table 4.3, which are accompanied by a brief description of each of the parts that make them up.

Table 4.3 Configuration screens for group I/O and their description

<table>
<tr><th>Group I/O configuration screens</th><th>Description</th></tr>
<tr><td>I/O monitor main screen

I/O Group Out
SIM VALUE 2/300
GO[1] * * [ ]
GO[2] * * [ ]
GO[3] * * [ ]
GO[4] * * [ ]
GO[5] * * [ ]
GO[6] * * [ ]
GO[7] * * [ ]
GO[8] * * [ ]
GO[9] * * [ ]
GO[10] * * [ ]
GO[11] * * [ ]
 </td><td>– # = Defines the number of I/O to be used or configured.
– SIM= Indicates the simulation status of the I/O.
  – F4:SIMULATE→ I/O is being simulated at software level (without generating physical voltages).
  – F5:UNSIM→ I/O is forced by generating a physical voltage.
– STATUS=Declares the status of each I/O.
  – F4:ON
  – F5:OFF
– F3: IN/OUT → Change inputs to outputs and vice versa.</td></tr>
</table>

Configuration screen F2: CONFIG 	– GO#/GI# = Defines the number of I/O to be used or configured. – RACK=This is the physical location where the I/O board(s) or module(s) is mounted. – SLOT=The RACK space where the I/O module is connected. – START PT= Port number in the sequence of ports on the I/O module. – NUM PTS=Number of points. – NEXT+F2: VERIFY → Verify if the assignment is valid. – F3: IN/OUT → Change inputs to outputs and vice versa.

```
I/O Group Out
                                      1/300
      GO #   RACK   SLOT   START PT   NUM PTS
        1      0      0       0          0
        2      0      0       0          0
        3      0      0       0          0
        4      0      0       0          0
        5      0      0       0          0
        6      0      0       0          0
        7      0      0       0          0
        8      0      0       0          0
        9      0      0       0          0
       10      0      0       0          0
```

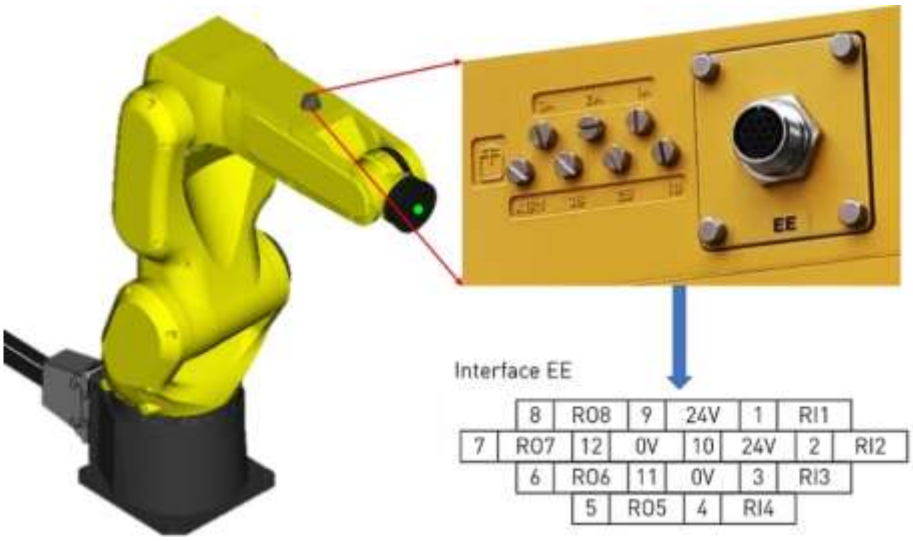
Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

Note: After any configuration, you must do a cold start, so you must turn the robot controller off and on again for the system to start with the new variables.

4.2.4 Robot I/O “RI/RO”

The robot has I/O between the controller and the robot, which send signals to the end effector (EE) connector located on axis 3 (see Figure 4.2). They are mainly used with tools mounted on the robot flange. Here the state of the outputs can be changed from the PT and the number of them depends on the number of axes in the system, in this case, this version of the robot has 8 I/Os.

Figure 4.1 FANUC LR MATE 200ID robot EE connector



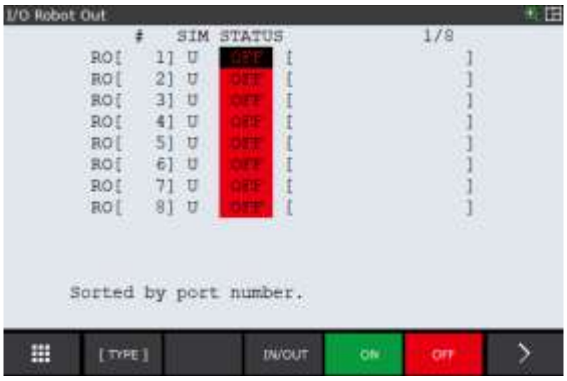
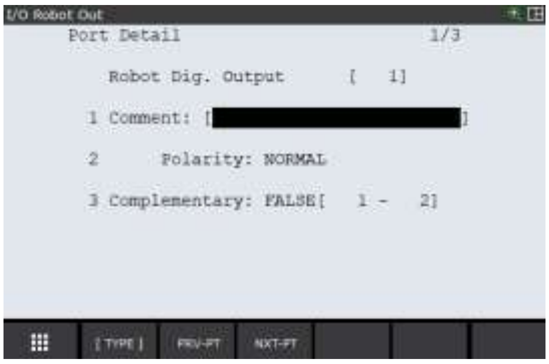
Interface EE

	8	RO8	9	24V	1	RI1	
7	RO7	12	0V	10	24V	2	RI2
	6	RO6	11	0V	3	RI3	
		5	RO5	4	RI4		

Source: Elaborated by the work team

Below is a description of the parameters that can be configured for this type of output (see table 4.4). Note that after any configuration, you must do a cold start by turning the robot controller off and on again for the system to start with the new variables.

Table 4.4 Configuration screens for robot I/O and their description

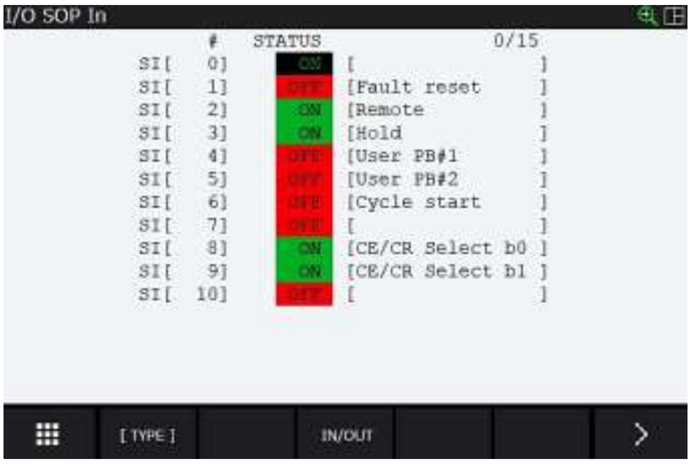
Robot I/O configuration screens	Description
I/O monitor main screen  I/O Robot Out # SIM STATUS 1/8 RO[1] U OFF [ ] RO[2] U OFF [ ] RO[3] U OFF [ ] RO[4] U OFF [ ] RO[5] U OFF [ ] RO[6] U OFF [ ] RO[7] U OFF [ ] RO[8] U OFF [ ] Sorted by port number. [TYPE] IN/OUT ON OFF	– # = Defines the number of I/O to be used or configured. – SIM= Indicates the simulation status of the I/O. – F4:SIMULATE→ I/O is being simulated at software level (without generating physical voltages). – F5:UNSIM→ I/O is forced by generating a physical voltage. – STATUS=Declares the status of each I/O. – F4:ON – F5:OFF
Port details screen NEXT→ DETAIL  I/O Robot Out Port Detail 1/3 Robot Dig. Output [1] 1 Comment: [ ] 2 Polarity: NORMAL 3 Complementary: FALSE[1 - 2] [TYPE] PRV-PT NXT-PT	– Comment: Allows to declare a comment for the I/O. – Polarity: Sets the polarity of the I/O. – F4:INVERSE→ ON=0V and OFF=24V – F5:NORMAL→ ON=24V and OFF=0V – Complementary: Associates two outputs at the same time. – F4: TRUE →For DO[1]=ON, DO[2]=OFF For DO[1]=OFF, DO[2]=ON – F5: FALSE →The outputs are independent.

Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

4.2.5 Standard Operator Panel (SOP) I/O "SI/SO"

These are digital signals that show the status of the system without generating physical voltages and allowing them to be queried and used in programming. Here, the output signals (SO) correspond to the panel digital output signals (PDO) and the inputs (SI) correspond to the panel digital input signals (PDI) in the internal controller. The direction to find the list of this type of I/O is MENU → 5-I/O → -SOP→ F3:IN/OUT (see in Figure 4.2).

Figure 4.2 Standard Operator Panel inputs SI



```
I/O SOP In
        #   STATUS              0/15
SI[   0]    ON    [                ]
SI[   1]    OFF   [Fault reset     ]
SI[   2]    ON    [Remote          ]
SI[   3]    ON    [Hold            ]
SI[   4]    OFF   [User PB#1       ]
SI[   5]    OFF   [User PB#2       ]
SI[   6]    OFF   [Cycle start     ]
SI[   7]    OFF   [                ]
SI[   8]    ON    [CE/CR Select b0 ]
SI[   9]    ON    [CE/CR Select b1 ]
SI[  10]    OFF   [                ]
```

Source: Image obtained from Roboguide software

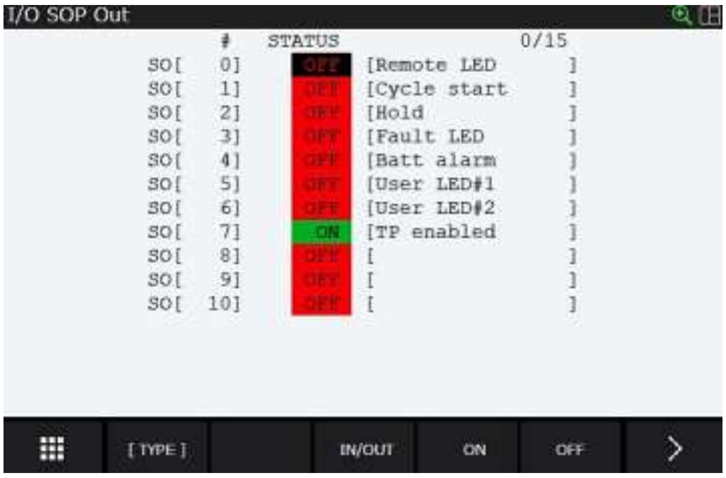
Table 4.5 Description of the inputs SI of the Standard Operator Panel

SI	Activation	Signal	Description
1	Normally off	Fault reset	Indicates that the FAULT RESET button is not, being pressed.
2	Always on	Remote	Indicates the controller configuration: – ON=Remote mode – OFF=Local mode To turn the remote signal on or off, follow the direction: MENU → 0-NEXT → 6-SYSTEM → F1:[TYPE] → 5-CONFIG→ Remote/local setup (approx. Item 42) → F4:[CHOICE].
3	Always on	Hold	When lit, it indicates that the HOLD key is not being pressed. Otherwise, if any program is running, it will slow down the speed of the robot and stop all action temporarily.
4	Normally off	User PB#1	When off, it indicates that the variable is not being pressed or used.
5	Normally off	User PB#2	When off, it indicates that the variable is not being pressed or used.
6	Normally off	Cycle start	Indicates the activation of the cycle button. When activated, it executes the selected program and once the action is finished, it is deactivated again.
8	Normally off	CE/CR b0	The offered signal is used for the mode selection switch.
9	Normally off	CE/CR b1	The offered signal is used for the mode selection switch.
0,7,10-15	Normally off	Vacant	Slots available for additional PDI.

Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

As can be seen, Figure 4.2 and Table 4.5 describe the inputs of the SI standard operator panel, therefore, Figure 4.3 and Table 4.6 will describe the SO outputs, as seen below.

Figure 4.3 Standard Operator Panel outputs SO



```
I/O SOP Out
                 #   STATUS               0/15
        SO[   0]     OFF   [Remote LED     ]
        SO[   1]     OFF   [Cycle start    ]
        SO[   2]     OFF   [Hold           ]
        SO[   3]     OFF   [Fault LED      ]
        SO[   4]     OFF   [Batt alarm     ]
        SO[   5]     OFF   [User LED#1     ]
        SO[   6]     OFF   [User LED#2     ]
        SO[   7]      ON   [TP enabled     ]
        SO[   8]     OFF   [               ]
        SO[   9]     OFF   [               ]
        SO[  10]     OFF   [               ]
```

Source: Image obtained from Roboguide software

Table 4.6 Description of the outputs SO of the Standard Operator Panel

SO	Activation	Signal	Description
0	Normally off	Remote LED	Indicates whether the controller is configured as remote. In this case, it will change to ON when the following conditions are satisfied: – The TP is OFF. – The controller is in remote mode. – A program is running.
1	Normally off	Cycle Start	Indicates the activation of the cycle button. When activated, it executes the selected program and once the action is finished, it is deactivated again.
2	Normally off	Hold	These lights up when the HOLD key is pressed and indicates that the execution of a program will be stopped.
3	Normally off	Fault LED	Indicates that a fault has occurred. If not automatically deactivated, the fault has not been reset.
4	Normally off	Batt alarm	If it is on, it indicates that the CPU voltage or the voltage of the batteries in the robot is low, so a replacement or maintenance must be done.
5	Normally off	User LED#1	For processes with output signals for Handling tool, this signal is not used, however, for other types of applications, the signal can be defined by the user.

6	Normally off	User LED#2	For processes with output signals for Handling tool, this signal is not used, however, for other types of applications, the signal can be defined by the user.
7	Normally on in local mode	TP Enable	Indicates that the portable console (TP) is enabled and only turns on when the TP activation switch is OFF.
8-15	Normally off	Vacant	Slots available for additional PDI.

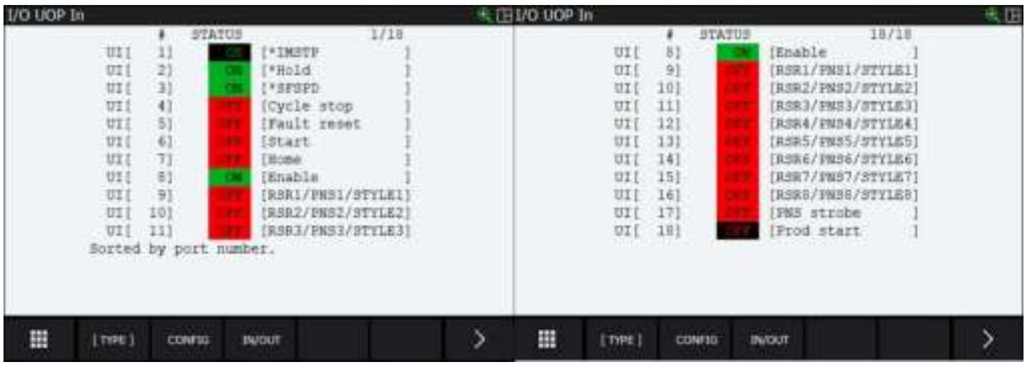
Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

It is worth mentioning that no input or output that is already declared can be configured, in that mode, only the outputs can be activated by the user by pressing F4:ON or F5:OFF .

4.2.6 User Operator Panel (UOP) I/O "UI/UO"

This version of the controller has 18 inputs and 20 outputs that can be used. These allow to manipulate a robot remotely with signals used as communication interface (e.g., PLC's). The direction to find the available list of this type of I/O is MENU → 5-I/O → 7-UOP→ F3:IN/OUT.

Figure 4.4 User Operator Panel inputs UI



```
I/O UOP In
          #   STATUS           1/18
   UI[   1]     ON   [*IMSTP           ]
   UI[   2]     ON   [*Hold            ]
   UI[   3]     ON   [*SFSPD           ]
   UI[   4]    OFF   [Cycle stop       ]
   UI[   5]    OFF   [Fault reset      ]
   UI[   6]    OFF   [Start            ]
   UI[   7]    OFF   [Home             ]
   UI[   8]     ON   [Enable           ]
   UI[   9]    OFF   [RSR1/PNS1/STYLE1]
   UI[  10]    OFF   [RSR2/PNS2/STYLE2]
   UI[  11]    OFF   [RSR3/PNS3/STYLE3]
Sorted by port number.
```

```
I/O UOP In
          #   STATUS           18/18
   UI[   8]     ON   [Enable           ]
   UI[   9]    OFF   [RSR1/PNS1/STYLE1]
   UI[  10]    OFF   [RSR2/PNS2/STYLE2]
   UI[  11]    OFF   [RSR3/PNS3/STYLE3]
   UI[  12]    OFF   [RSR4/PNS4/STYLE4]
   UI[  13]    OFF   [RSR5/PNS5/STYLE5]
   UI[  14]    OFF   [RSR6/PNS6/STYLE6]
   UI[  15]    OFF   [RSR7/PNS7/STYLE7]
   UI[  16]    OFF   [RSR8/PNS8/STYLE8]
   UI[  17]    OFF   [PNS strobe       ]
   UI[  18]    OFF   [Prod start       ]
```

Source: Image obtained from Roboguide software

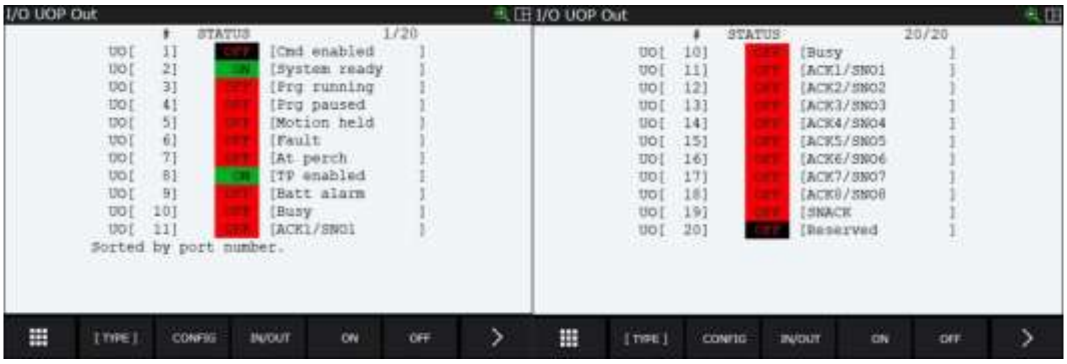
Table 4.7 Description of the inputs UI of the User Operator Panel

UI	Activation	Signal	Description
1	Normally on	*IMSTP	It is a software stop signal that is always on. When its status changes to OFF it generates: – That the execution of the running program is stopped. – The brakes are activated, and the robot servos are disconnected.
2	Normally on	*HOLD	It is a signal that is normally ON. When the status changes, the running program execution is stopped, and an external electrical shutdown is activated.
3	Normally on	*SFSPD	When the signal changes to OFF, the program stops and the speed variable in the system is reduced. This generates the SYST-009 error, and the robot can only move in manual mode and with a limit on the speed.
4	Always on	Cycle stop	It is a cycle stop signal, when OFF, the program continues to the end. Otherwise, any execution action will be aborted.
5	Always on	Fault reset	This signal is manipulated externally, so it can accept or reset faults, activating the robot servos, and resuming a program.
6	Active only in remote mode	Start	Using the remote mode, this signal can send a signal to run, resume or restart a program (using the Prod Start signal).
7	Active only in remote mode	Home	Signal that moves the robot to its home status, provided that the home position is programmed in a macro-command.
8	Normally on	Enable	Keeps the control enabled to move the robot when it is ON, otherwise in remote mode, the error SYST-016 may appear.
9-16	Active only in remote mode	RSR/ PNS 1-8	Depending on the configuration and the program to be executed, RSR are programs that generate robot service signals and PNS are programs to be executed together with START and PROD_START signals.
17	Active only in remote mode	PNS strobe	These are signals that validate the number of the program to be executed, using PNS signals.
18	Active only in remote mode	Prod Start	This signal initiates the launch of a program. Using PNS, this signal executes the program, otherwise the program is executed from the last cursor position.

Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

As can be seen, Figure 4.4 and Table 4.7 describe the inputs of the UI user operator panel, therefore, Figure 4.5 and Table 4.8 will describe the UO outputs, as seen below.

Figure 4.5 User Operator Panel outputs UO



```
I/O UOP Out                                      I/O UOP Out
         #    STATUS              1/20                    #    STATUS              20/20
  UO[   1]     OFF   [Cmd enabled     ]            UO[  10]     OFF   [Busy            ]
  UO[   2]     ON    [System ready    ]            UO[  11]     OFF   [ACK1/SNO1       ]
  UO[   3]     OFF   [Prg running     ]            UO[  12]     OFF   [ACK2/SNO2       ]
  UO[   4]     OFF   [Prg paused      ]            UO[  13]     OFF   [ACK3/SNO3       ]
  UO[   5]     OFF   [Motion held     ]            UO[  14]     OFF   [ACK4/SNO4       ]
  UO[   6]     OFF   [Fault           ]            UO[  15]     OFF   [ACK5/SNO5       ]
  UO[   7]     OFF   [At perch        ]            UO[  16]     OFF   [ACK6/SNO6       ]
  UO[   8]     ON    [TP enabled      ]            UO[  17]     OFF   [ACK7/SNO7       ]
  UO[   9]     OFF   [Batt alarm      ]            UO[  18]     OFF   [ACK8/SNO8       ]
  UO[  10]     OFF   [Busy            ]            UO[  19]     OFF   [SNACK           ]
  UO[  11]     OFF   [ACK1/SNO1       ]            UO[  20]     OFF   [Reserved        ]
Sorted by port number.
```

Source: Image obtained from Roboguide software

Table 4.8 Description of the outputs UO of the User Operator Panel

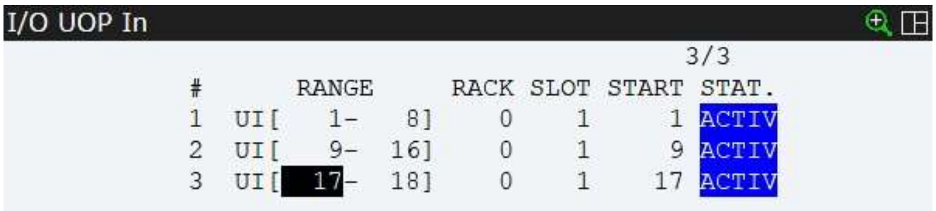
UO	Signal	Description
1	Cmd enable	The signal is activated when the conditions are satisfied: – TP OFF and STEP mode disabled. – The controller is in remote mode. It can be activated from MENU → 0-NEXT → 6-SYSTEM → F1:[TYPE] → 5-CONFIG→ Remote/local setup (approx. Item 42) → F4:[CHOICE]. – SFSPD (UI[3]) and ENBL (UI[8]) signals enabled in ON state. – System variable $RMT_MASTER set to 0. – Controller in automatic "AUTO" mode.
2	System ready	Indicates that the robot servos are energized, and the system is ready.
3	Prg running	Activated when a program is running.
4	Prg paused	It is activated when the execution of a program has been interrupted.
5	Motion held	The signal is activated only when the HOLD key is pressed or when the HOLD input for the UOP is OFF.
6	Fault	This output is activated when a program enters error status, with exceptions in WARN alarms and when the FAULT_RESET input is activated.
7	At perch	This signal is activated when the robot reaches a position that was defined before by the user. Here, the first reference position is the one that will receive the At perch signal, all others will get digital outputs.
8	TP enable	This signal is activated when the TP has its status selector ON.
9	Batt alarm	If it is on, it indicates that the voltage of the CPU batteries or the voltage of the batteries powering the robot encoders is low, so a replacement or maintenance has to be done.
10	Busy	It is activated when the robot is executing a program or action, in general, it sends the signal when the processor is busy.
11-18	ACK/SNO 1-8	The ACK signal group is responsible for sending a pulse when the RSR signal of a program is received. On the other hand, the SNO signal group performs the same action as the ACK signals, in this case, for PNS programs, making both signal groups work as pulse generators by recognition.
19	SNACK	This signal is sent when the system recognizes a program selected by a PNS input, in which case it confirms the acceptance of this program.

Source: Information obtained from ROBOGUIDE software and (FANUC Robotics Ibérica S. L., 2018)

I/O configuration "SOP"

To configure any SOP input or output, press the F2:CONFIG key from the TP, where you will find the following definitions and keys that will help us to perform this action (see figure 4.6).

Figure 4.6 Configuration screen for UOP I/O



```
I/O UOP In
                                          3/3
   #      RANGE      RACK  SLOT  START  STAT.
   1  UI[    1-    8]   0     1      1  ACTIV
   2  UI[    9-   16]   0     1      9  ACTIV
   3  UI[   17-   18]   0     1     17  ACTIV
```

Source: Image obtained from Roboguide software

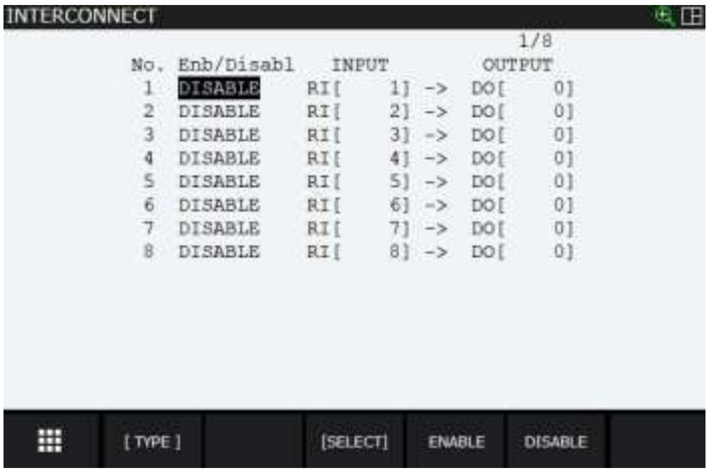
- RANGE = Shows the number of I/O that are in a signal range.
- RACK= The physical location where the I/O board or modules are mounted.
- SLOT= The space in the RACK where the I/O module is connected.
- START= The port used from the board or I/O module.
- STAT= Indicates the status of the digital signal.
  - ACTIV → The assignment is valid.
  - UNASG→ The assignment has not been made.
  - INVAL→ The assignment is invalid
  - PEND→ The assignment is valid but not active.
- F2: MONITOR= Returns to the previous screen.
- F3:IN/OUT= Switches from the input screen to the output screen and vice versa.

Note: After any configuration, you must do a cold start, so you must turn the robot controller off and on again for the system to start with the new variables.

4.2.7 I/O interconnection

This option allows you to interconnect or redirect inputs, for digital or robot outputs, so that there is a way to inform other elements of the status of these I/Os. You can find these interconnections by pressing MENU → 5-I/O → 9-INTERCONNECT (see Figure 4.7).

Figure 4.7 Interconnections screen for I/O



```
INTERCONNECT
                                        1/8
 No. Enb/Disabl   INPUT           OUTPUT
  1  DISABLE     RI[    1] ->  DO[     0]
  2  DISABLE     RI[    2] ->  DO[     0]
  3  DISABLE     RI[    3] ->  DO[     0]
  4  DISABLE     RI[    4] ->  DO[     0]
  5  DISABLE     RI[    5] ->  DO[     0]
  6  DISABLE     RI[    6] ->  DO[     0]
  7  DISABLE     RI[    7] ->  DO[     0]
  8  DISABLE     RI[    8] ->  DO[     0]
```

Source: Image obtained from Roboguide software

There are 5 types of interconnections when pressing F3:[SELECT], which are:

- RI → DO (8 interconnections available)
- DI → RO (8 interconnections available)
- DI → DO (32 interconnections available)
- SI → DO (16 interconnections available)
- ES → DO (16 interconnections available)

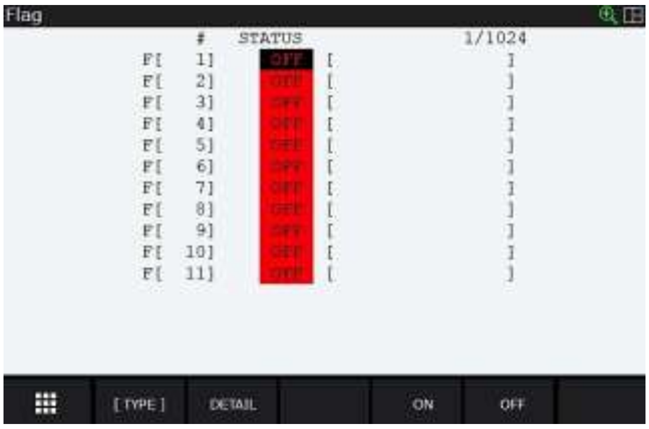
The latter, allows the use of safety signals within which, some examples are:

- EMGOP – Signal generated by the controller emergency stop.
- EMGTP – Signal generated by the emergency stop of the TP.
- DEADMAN – Signal generated by pressing the DEADMAN Switch of the TP.

4.2.7 Flags

Flags are internal I/O ports that are used to maintain the status (ON or OFF) of a condition or other output. They are not connected to an I/O device, but you can call them from a program or modify them manually. The flags available in the system can be observed from: **MENU → 5-I/O → 0-NEXT→ 2-FLAG** (see in Figure 4.8).

Figure 4.8 Display of available flags on the R-30iB Plus controller



```
            #   STATUS                1/1024
     F[    1]    OFF   [                   ]
     F[    2]    OFF   [                   ]
     F[    3]    OFF   [                   ]
     F[    4]    OFF   [                   ]
     F[    5]    OFF   [                   ]
     F[    6]    OFF   [                   ]
     F[    7]    OFF   [                   ]
     F[    8]    OFF   [                   ]
     F[    9]    OFF   [                   ]
     F[   10]    OFF   [                   ]
     F[   11]    OFF   [                   ]
```

Source: Image obtained from Roboguide software

If the value of the flags is lost for any reason, there is a way to recover them by performing a warm start, however, if any of the following conditions shown below occur, the flags will always be or change to OFF:

- If a cold start, controlled start or INIT Start is performed.
- If there are changes in I/O assignments or configurations.

In addition to the possibility of changing the value of a flag manually, we can place a comment to each flag by positioning the cursor on one of them and pressing F2:DETAIL, this will redirect us to a new screen where you can use the function keys F2:PRV-PT and F3: NXT-PT to navigate between flags.

4.2.9 I/O configuration

There are different ways to configure the value of the inputs and outputs in the system, among the most known are those shown below, of which you can follow the steps to configure them yourself.

- **Manual**

1: Identify that the system variable $IO_AUTO_CFG = 0, if not, change the value.
2: Manually configure the I/O you need.
3: Perform a cold start (turning the controller on and off) to save the changes you made.

- **Automatic**

1: Identify that the system variable $IO_AUTO_CFG = 1, if not, change the value.
2: Press from the TP, keys MENU → 5-I/O → 0-NEXT→ 1-LINK DEVICE → F5:CLR-ASG, this to clear all I/O assignments.
3: Perform a cold start (turning the controller on and off). Here the controller will automatically search and configure the connected I/O according to the Hardware.

NOTE: If the automatic mode does not detect assignments, all I/O in the system will not be declared and therefore, in sections such as SIM and STATUS an asterisk will appear, and you will have to assign them again.

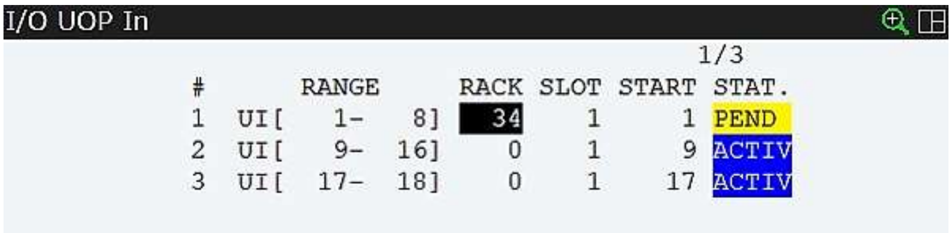
– **Configuration with flags**

It is possible to perform an I/O assignment with flags, i.e., the value taken by the configured I/O will be the same as the value given to the specified flag (see Figure 4.9). This is possible only for digital (DI/DO), group (GI/GO) and user operator (UI/UO) inputs and outputs.

1. For DI, DO, GI, GO, UI and UO inputs press the F2 key: CONFIG.

2. Select a range and change the following assignments.
a. Rack = 34
b. Slot = 1
c. Start = 1

Figure 4.9 Configuration of flags from UI inputs



```
I/O UOP In
                                 1/3
 #       RANGE       RACK SLOT START STAT.
 1   UI[   1-   8]     34    1     1 PEND
 2   UI[   9-  16]      0    1     9 ACTIV
 3   UI[  17-  18]      0    1    17 ACTIV
```

Source: Image obtained from Roboguide software

3. Perform a cold start (turning the controller on and off)

NOTE: Once the steps have been performed, changing the state of a flag within the specified range will also change the state for the I/O. For example, if the F[1] flag changes to OFF, the UI[1:*IMSTP] input will also change to OFF.

If you are looking to reverse the action, reset the original mapping values from step 2 and perform a cold start to safeguard the data.

4.2.10 I/O Controlled

The I/O control allows us to perform tests both for the physical inputs and outputs of the robot (with real voltages), as well as to test programming logics. Among these controls are:

– **Forced outputs**

Here, an output is forced when its state is changed from ON to OFF and can be changed manually or through I/O instructions in a program. To force an output, it is necessary that this output is configured, after that follow the steps below:

1. Press MENU → 5-I/O → F1:[TYPE].

2. Select the type of output you want to force.

3. Identify that the output contains a U in the SIM column, this will indicate that the signal is not being simulated, if not, press the F5:UNSIM key.

4. Depending on the type of output you have chosen, you can force it using:
a. For digital (DI/DO), UOP and SOP outputs
   - F4: ON
   - F5: OFF
b. For analog (AI/AO) and group (GI/GO) outputs.

Move the cursor to the VALUE column, manually enter the decimal value and press ENTER. You can also insert hexadecimal values by pressing F4: FORMAT.

– **I/O simulation**

When the inputs and outputs are simulated, they do not generate any real voltage on the robot, so the main function of them is to perform tests on the logic of a program, among other things. It is worth mentioning that the simulation works only for digital, analog and group I/O, where it is necessary to be configured. To simulate I/O follow the steps below:

1. Press MENU → 5-I/O → F1:[TYPE].

2. Select the type of I/O (Digital, Analog or Group).

3. Identify that the output or input contains an S in the SIM column, this will indicate that the signal is being simulated, if not, press the F4:SIMULATE key.

4. Once you have done the above steps, you can force the I/O status by pressing F4:ON/F5:OFF for digital and group I/O or, by setting the value manually for analog I/O.

Note: In case you want to set all I/O signals as non-simulated, you can press from the TP the FCNT→ --0 NEXT-- → UNSIM ALL I/O key.

– **Cancel I/O**

To cancel an input or output there is the WAIT RELEASE function, which removes the pause generated by waiting for an I/O condition to be satisfied in a program. To use this function, it is necessary that a program is running and that there are programming lines that are waiting for an I/O response. Having the above:

1: Press the FCNT→ 7-RELEASE WAIT TP key.

4.3 Macro-commands

Macros are a type of program that is used as a complex instruction, however, they contain a set of simple instructions that are executed continuously to perform a specific operation, where the user can execute it as many times as needed. Macros are available for all programs and can be created, assigned, or called.

– **Creation of a macro-command**

The creation of a macro-command is identical to that of a normal program, with the difference that you must change the subtype and the appropriate mask group (see Figure 4.10). You can follow the steps below:

1. Press from the TP: SELECT→ F2:CREATE → Place a name → ENTER → F2: DETAIL.

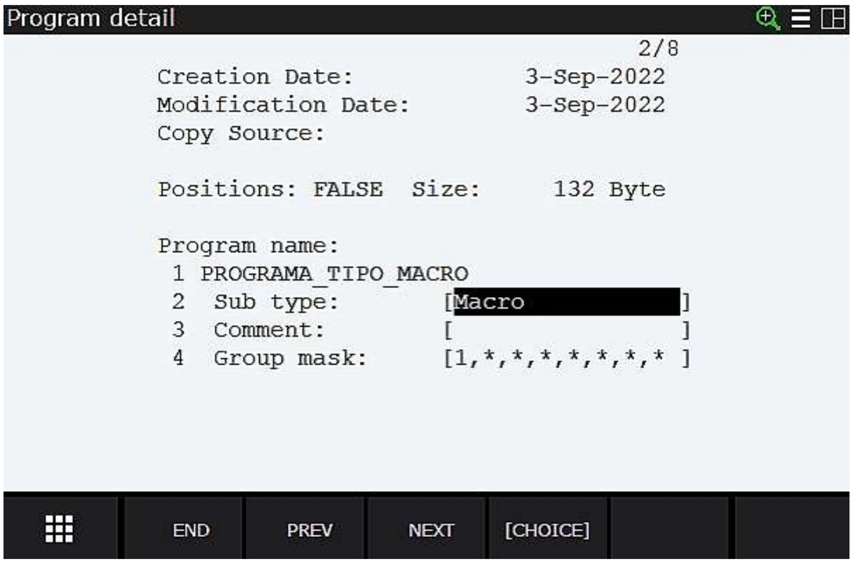
2. Position the TP cursor on item 2 "Sub type" and change the variable to MACRO by pressing F4:[CHOICE]→ 3-MACRO.

3. Depending on the type of macro you need, change the mask group:
   a. For signal-triggered macros: Select some mask group by placing a 1 in one of the available slots.
   b. For keystroke-activated macros: Place all the slots of the mask groups with an asterisk (*).

4. Press F1: END.

Note: For macros that are activated by a special keystroke, no robot movements can be programmed, because they do not belong to any mask group.

Figure 4.10 Creation of a MACRO program



```
                                       2/8
        Creation Date:            3-Sep-2022
        Modification Date:        3-Sep-2022
        Copy Source:

        Positions: FALSE  Size:     132 Byte

        Program name:
         1 PROGRAMA_TIPO_MACRO
         2  Sub type:          [Macro           ]
         3  Comment:           [                ]
         4  Group mask:        [1,*,*,*,*,*,*,* ]
```

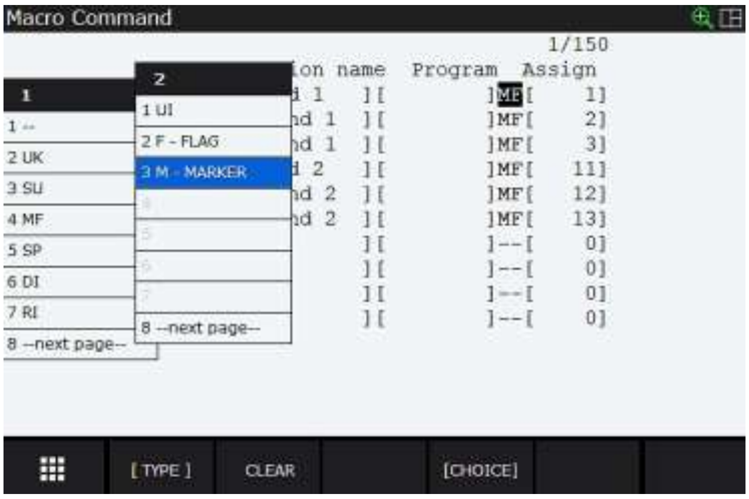
Source: Image obtained from Roboguide software

– **Macro-command assignment.**

Macro type programs can be called with the CALL or RUN instruction, however, there are already MACROS that are declared in "MENU →3-MANUAL FCTNS", to which you can configure or assign the previously created program and the type of signal that will activate it, where, among the available options are (see Figure 4.11):

- UK=Activation with a user key on the TP.
- SU=Activation with a user key on the TP + SHIFT key.
- MF=Selection of a MANUAL FCTNS menu item.
- SP= Activation with a user key on the controller.
- DI/RI/UI=Activation with a digital, robot or user input.
- F-FLAG=Activation with a flag.
- M-Marker=Activation with a marker.

Figure 4.11 Possible triggers for a MACRO program



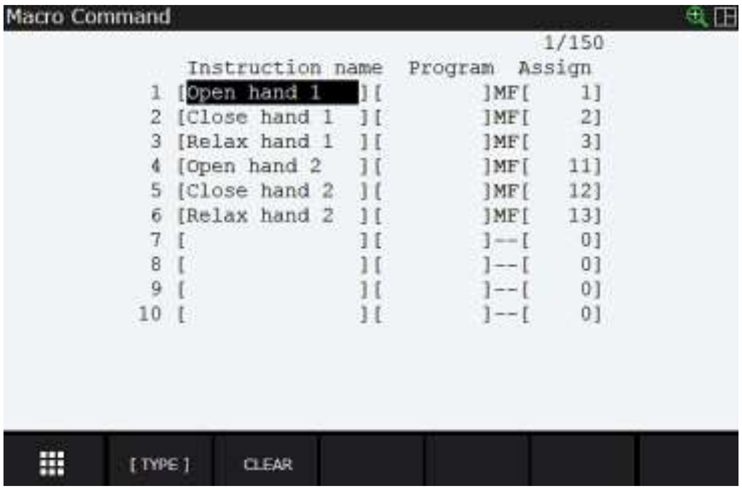
```
                              1/150
on name   Program  Assign
i 1     ][         ]MF[     1]
nd 1    ][         ]MF[     2]
nd 1    ][         ]MF[     3]
i 2     ][         ]MF[    11]
nd 2    ][         ]MF[    12]
nd 2    ][         ]MF[    13]
        ][         ]--[     0]
        ][         ]--[     0]
        ][         ]--[     0]
        ][         ]--[     0]
```

Source: Image obtained from Roboguide software

The address to find the list of macro commands that can be seen in Figure 4.11 is: MENU → 6-SET UP → 6-MACRO. In it (see in figure 4.12), to assign a previously created macro program we have the following options:

- Instruction name= The name of the MACRO type instruction is placed or set.
- Program= Select the program to be executed when the instruction is activated.
- Assign= Set the type of activation with which the instruction will be executed by pressing the F4:[CHOICE] key (see figure 4.11).

Figure 4.12 Main macro command screen



```
Macro Command
                                      1/150
          Instruction name   Program  Assign
    1 [Open hand 1     ][          ]MF[     1]
    2 [Close hand 1    ][          ]MF[     2]
    3 [Relax hand 1    ][          ]MF[     3]
    4 [Open hand 2     ][          ]MF[    11]
    5 [Close hand 2    ][          ]MF[    12]
    6 [Relax hand 2    ][          ]MF[    13]
    7 [                ][          ]--[     0]
    8 [                ][          ]--[     0]
    9 [                ][          ]--[     0]
   10 [                ][          ]--[     0]
```

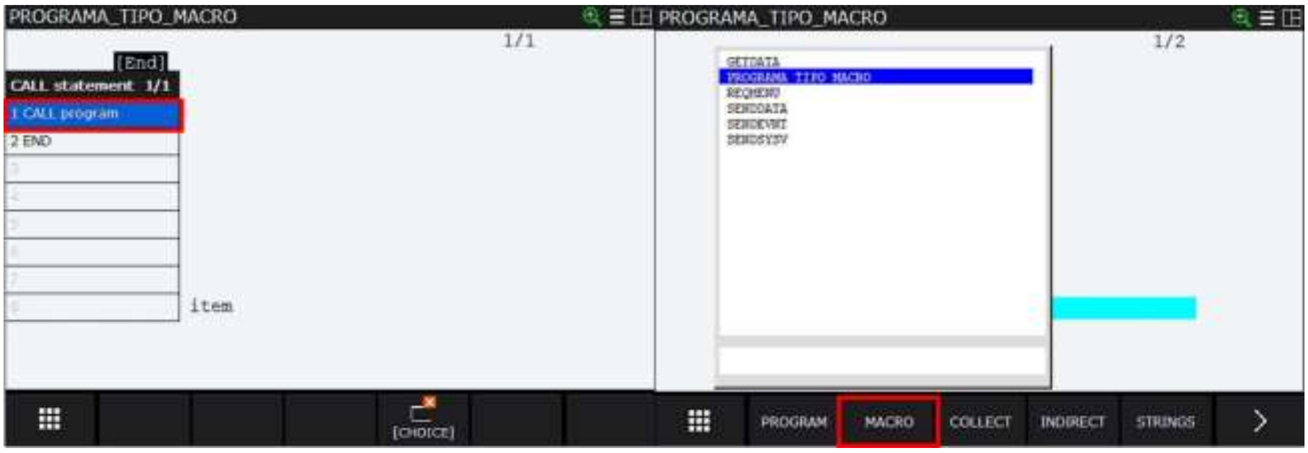
Source: Image obtained from Roboguide software

- **Macro-command as instruction**

To use a macro command from a program, you can use two types of instructions: CALL/RUN or the MACRO instruction (see in Figure 4.13). The former call directly to any program, in this case, for any defined as macro type, where, once the programming line is reached, the program is executed (here you exit the original program, execute the macro program, and return to the main execution). To use these instructions, click in:

F1: [INST]→ 6-CALL→ 1-CALL prog o F1: [INST]→ 8-NEXT→ 5-Multiple control → 1-RUN

Figure 4.13 Use of CALL and RUN instructions for macro commands



```
PROGRAMA_TIPO_MACRO                 1/1
[End]
CALL statement 1/1
1 CALL program
2 END
                 item
```

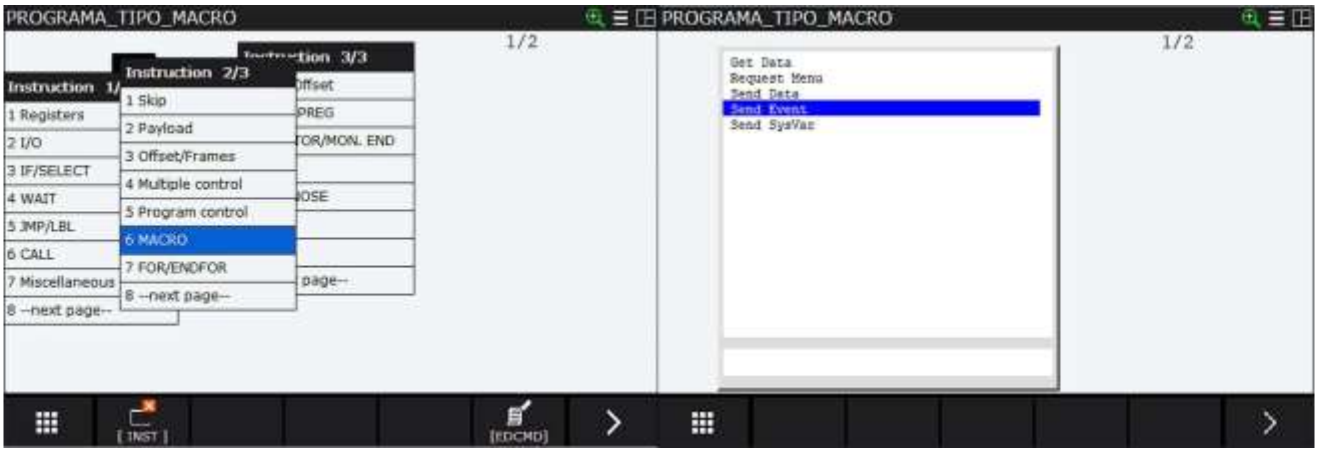
```
PROGRAMA_TIPO_MACRO                 1/2
GETDATA
PROGRAMA_TIPO_MACRO
REQMENU
SENDDATA
SENDEVNT
SENDSYSV
```

Source: Image obtained from Roboguide software

On the other hand, the MACRO instruction, when pressed, gives us a list of the macro commands that are declared from MENU → 6-SET UP → 6-MACRO, which already contain an assignment, in that case, when the programming reaches the line where this instruction is located or the assigned activation is performed, the program with which it was configured will be executed (see in Figure 4.14). To use this instruction, press:

F1: [INST]→ 8-NEXT→ 7-MACRO

Figure 4.14 Use of MACRO instruction



Source: Image obtained from Roboguide software

4.4 Repair and maintenance using macros

To perform repairs and maintenance on the robot, it is common to use macro type programs to develop specific movements or instructions. In addition, there is the possibility to execute a MACRO without using a program by pressing the SELECT key on the TP, to do so follow the steps below:

1. Press the MENU → MANUAL FCTNS key on the TP.
2. Unless other options appear in F1:[TYPE], press 1-Macros.
3. Select the macro you want to run, considering that it should already have assignments.
4. Enable the TP by pressing and holding the DEADMAN switch and press the RESET key.
5. Subsequently press and hold the SHIFT key and press the F3:EXEC key.

Note: the SHIFT key must be always pressed, since, if this condition is not met, the program is aborted and will not be allowed to resume at all.

Depending on the case, there is the possibility that, when testing a macro, it disables or needs to disable the FWD and BWD keys. To do so, follow the steps below:

1. Press on the TP the FCTN→ 2-Disable FWD/BWD key.

To know if the keys are disabled, the character "FBD" will appear at the top of the TP screen, in any case, if you try to use them, the error TPIF-052 FWD/BWD is disabled will be marked.

2. To enable the keys again, repeat step 1.

4.4.1 Program execution monitoring

A program can be monitored while it is being executed. To do so, the program must be displayed on the TP screen and the program cursor will highlight the line that is being executed in real time. When a program is running you will be able to monitor your program in two ways:

- **Using SELECT→ F4:MONITOR**

While a program is running and for some reason the TP does not display the program screen. You can press SELECT, this will send you to the list of available programs, in that case, place the cursor on the one you are looking for and press F4: MONITOR. This will show you the program on the entire TP screen, as well as a cursor, which will highlight the line you are executing in real time and move in conjunction with that execution.

### –	Using from the monitor → F2: LOOK

This function, when pressed while the program is running, will allow you to see other areas of the program by pressing the arrow keys on the TP. This is functional with very large programs where the user wants to navigate through them without having to wait for the cursor to go through a specific programming line. To remove the condition, press F2: MONITOR again and the cursor will indicate the normally executed line.

References

FANUC Robotics Ibérica S. L. (2018). ROBÓTICA PROGRAMACIÓN FANUC. *SEAT*. http://www.infoplc.net/files/descargas/fanuc_robotica/infoPLC_net_ManualFANUCR-J3iB.pdf

Pinto Hernández, J. de. (2016). *Programación de robot Kuka para paletizado*. http://hdl.handle.net/10016/24236

Chapter 5 Mastering, backups and additional information

Abstract

Although knowing the control and programming of an industrial robot is important to develop specific applications, there are methods that help to verify and even repair the robot directly after a series of commands or actions when for some specific reason it cannot move. This is since the robot lost the data of its registration position of the encoders of each of the servomotors that constitute it. With this it is said that the robot must be mastered or, in other words, mastered, which is basically to reset the registration data that the robot had from the beginning, so that it can be manipulated again. Changing the scenario, there are also multiple ways to validate actions or programs performed with the robotic system, this can be done with backups, which are backups that work mainly to safeguard important information or validate actions. For this reason, chapter 5 will describe important information, external to the programming as the previous one, so that it can be replicated by other users who have access to a similar robot version.

Mastering, Verifying, Repairing

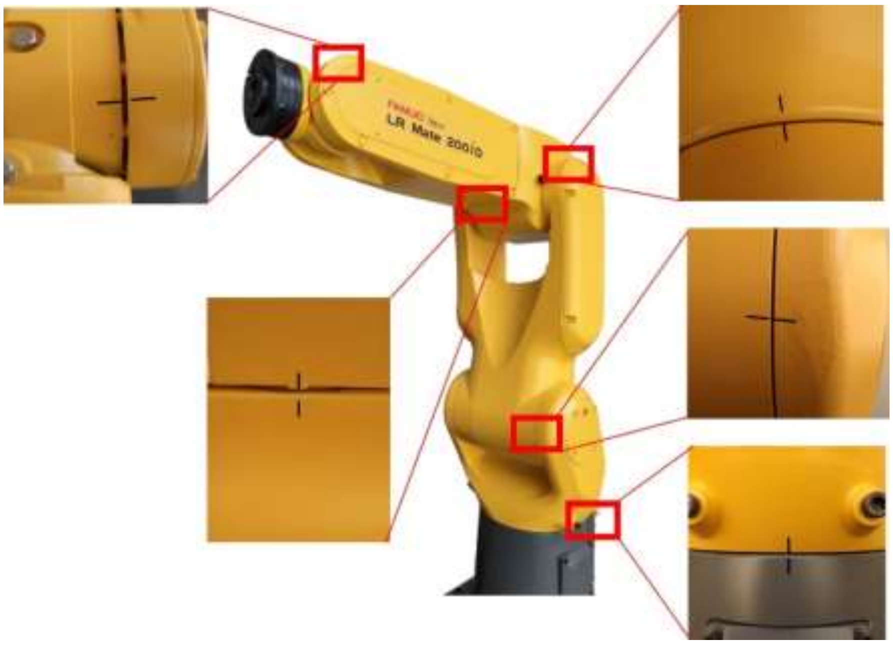
5.1 Introduction

Generally, it is said that all industrial robots are mastered, since if this is not the case, no operation can be performed with them (Limón Molina, 2008). In that case the mastering associates the angle between the fixed and mobile marks found in the robot, with the pulses generated by the encoders found in each of the servomotors of the robot trying to get as close as possible to the mechanical zero (FANUC Robotics Ibérica S. L., 2018). On the other hand, there are backups, which are elements that automatically safeguard each one of the programs performed and to be performed in the controller memory, however, these data can be extracted to have an alternate backup or simply to validate actions. Thus, there are different issues that are important for users who constantly work with industrial robots. The focus of chapter 5 is based on describing extra information to the programming, specifying points that are commonly used such as mastering, backups, fault detection, among other points. All generalized so that the information can be used by users that match the version of the Fanuc LR Mate 200iD robot.

5.2 Mastering and calibration

Mastering consists of adjusting the mechanical zero on the robot or retrieving the last zero set, guided by calibration marks on each axis junction (see figure 5.1). When there is no mastering process, no operation can be performed on the robot (Limón Molina, 2008).

Figure 5.1 Location of calibration marks on FANUC LR MATE 200iD robot



Source: Elaborated by the work team

5.2.1 Common mistakes in mastering and their solution

– **How to find MASTER/CAL**

Whenever we want to do a mastering we have to follow the direction:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 3-MASTER/CAL

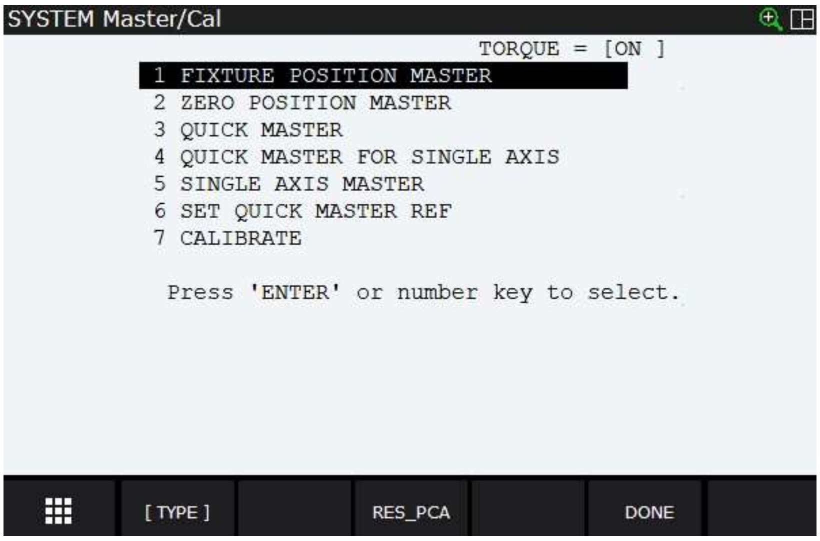
However, there is always the case where the option does not appear. To solve it, just go to the following direction:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 2--VARIABLES

Once there, with the teach cursors or by pressing the ITEM key, we will move to approximately ITEM 360 with the name $MASTER_ENB and change the value from 0 to 1. After that we press the following direction, and we will get the screen in figure 5.2:

F1: [TYPE] → 3-MASTER/CAL

Figure 5.2 Master/Cal main screen



```
SYSTEM Master/Cal
                                TORQUE = [ON ]
          1 FIXTURE POSITION MASTER
          2 ZERO POSITION MASTER
          3 QUICK MASTER
          4 QUICK MASTER FOR SINGLE AXIS
          5 SINGLE AXIS MASTER
          6 SET QUICK MASTER REF
          7 CALIBRATE

           Press 'ENTER' or number key to select.
```

Source: Image obtained from Roboguide software

– **ERROR SRVO-062 BZAL alarm**

According to FANUC Robotics Ibérica S.L. The situations for which the robot would need mastering may be the following (FANUC Robotics Ibérica S. L., 2018):

- The robot lost hardware mastering and now cannot move.
- The robot lost voltage due to battery wear.
- The power supply from the batteries to the encoders was cut due to malfunctions, replacement of some motor, etc.

Whatever the current situation of the robot, almost always will appear the error SERVO 062-BZAL alarm, which indicates that there is an error to detect the batteries, since it marks them as if they were worn out. To solve this error, there are two methods:

1) We go to MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 3-MASTER/CAL, and there we press F3: RES_PCA → F4: YES. This resets the encoder alarm.

2) If the first option does not work, the second option directs us to MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 2-VARIABLES and we move with the cursors to ITEM $MCR approximately at option 330, press ENTER and at ITEM 14 $SPC_RESET we change the variable FALSE to TRUE, immediately the variable will change back to FALSE, but the change will already be applied.

   Finally, a cold start is made by turning off the controller and turning it back on after a few seconds.

– **ERROR SRVO-075 Pulse not established (G:1 A:6)**

This error can happen when solving the error SRVO-062 after turning on the controller again, this should not appear if we want to perform a mastering, so we must correct it, for this we must move each of the axes of the robot +/-10° with JOINT coordinates and when finished, press the RESET key. When the error disappears, we must return the robot back to its calibration marks or to its additional marks (in case it has them), all this with movements in JOINT coordinates.

5.2.2 Mastering and calibration “Zero posicional master”

The mastering "Zero positional master" consists of moving the robot to its calibration marks to try to recover the last mechanical zero, for this we follow the steps below:

1) Through JOINT coordinates, we move the robot to the calibration marks seen in figure 5.1 as centered as possible.

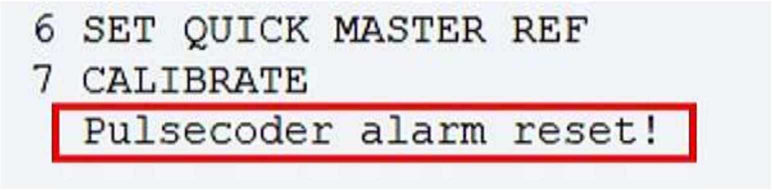
2) After that, we press the following address on the Teach Pendant:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 3-MASTER/CAL

(If MASTER/CAL does not appear go back to point 5.2.1).

3) Now select F3: [RES_PCA] and press F4: [YES]. This will reset the enconder alarm and the message "Pulsecoder alarm reset!" should appear (see figure 5.3).

Figure 5.3 Note "Pulse encoder alarm reset!



```
6 SET QUICK MASTER REF
7 CALIBRATE
  Pulsecoder alarm reset!
```

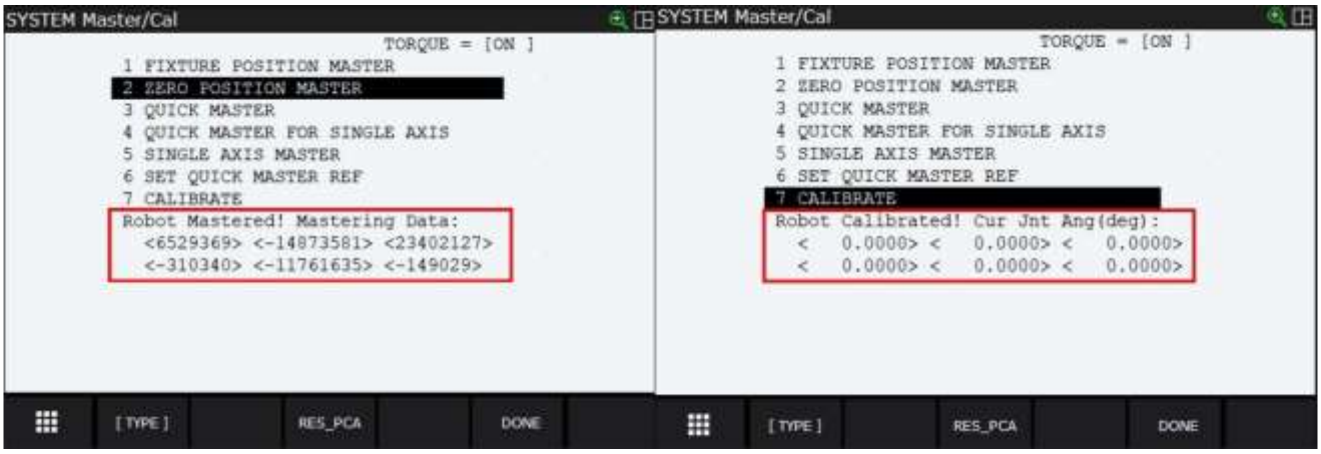
Source: Image obtained from Roboguide software

4) Press SHIFT+RESET to clear the general alarms of the robot.

5) Subsequently, press the DEADMAN and RESET to bring in the robot contactors and, without releasing the deadman, select 2-ZERO POSITION MASTER and press YES. This will give us the legend "Robot Mastered" (see figure 5.4) and we will have the robot mastered.

6) To calibrate the robot, select 7-CALIBRATE and press YES, and the process will give us the sentence "Robot Calibrated" (see figure 5.4) and the robot will be calibrated.

Figure 5.4 Pop-up notes when mastering and calibrating the robot



```
Robot Mastered! Mastering Data:
  <6529369> <-14873581> <23402127>
  <-310340> <-11761635> <-149029>
```

```
Robot Calibrated! Cur Jnt Ang(deg):
  <    0.0000> <    0.0000> <    0.0000>
  <    0.0000> <    0.0000> <    0.0000>
```

Source: Image obtained from Roboguide software

Note: In case the legends do not appear, or the mastering did not work, try this method again. If the problem persists try using a different method.

5.2.3 Mastering and calibration "Quick master"

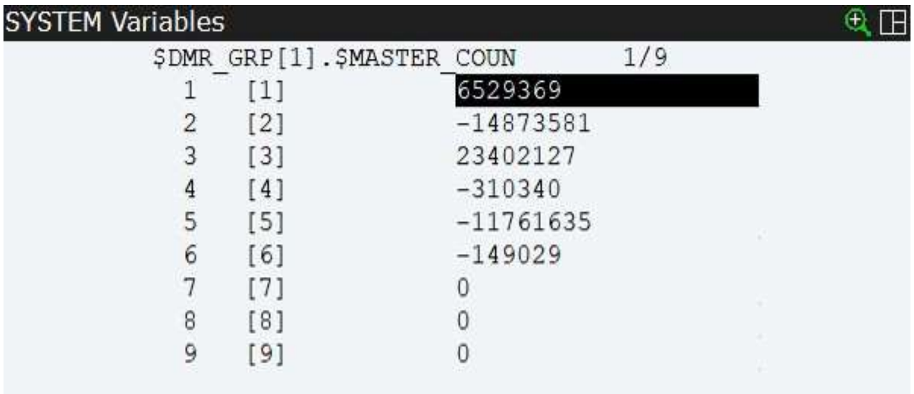
To perform a mastering through Quick Master, a subprocess has to be done depending on the case and the current situation of the robot, these cases depend on whether the robot has already lost the mastering or if it still has it. In addition, it is worth mentioning that these cases are raised once the SRVO-062 error for spent batteries has been solved.

5.2.3.1 When mastering is already lost

When the mastering is lost and there is no way to move the robot, we have to perform a series of steps involving manual data entry, since we take the old positions that are recorded from the last mastering and pass them to some reference subvariables, the steps to achieve this are as follows:

1) We go to MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 2-VARIABLES and select ITEM $DMR_GPR (Approximately between ITEM 150 and 175). Then press ENTER → ENTER and select ITEM 4 $MASTER_COUN → ENTER, when you enter a list of variables will appear (see in Figure 5.5) and you should write down the values that appear on a sheet of paper or on some device.

Figure 5.5 Variables in $MASTER_COUN

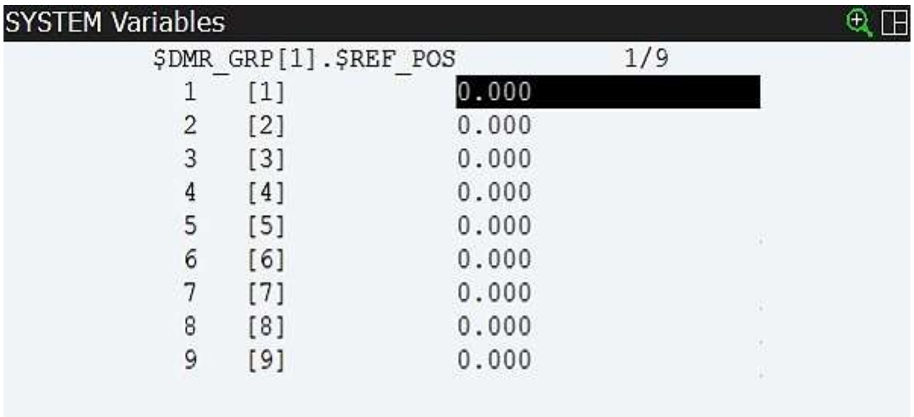


```
        $DMR_GRP[1].$MASTER_COUN        1/9
           1   [1]           6529369
           2   [2]           -14873581
           3   [3]           23402127
           4   [4]           -310340
           5   [5]           -11761635
           6   [6]           -149029
           7   [7]           0
           8   [8]           0
           9   [9]           0
```

Source: Image obtained from Roboguide software

2) Once you have noted down the values, press PREV to return to the previous menu and select ITEM6 $REF_POS → ENTER. Here make sure that all variables are zero, if not change them manually (see Figure 5.6).

Figure 5.6 Variables in $REF_POS

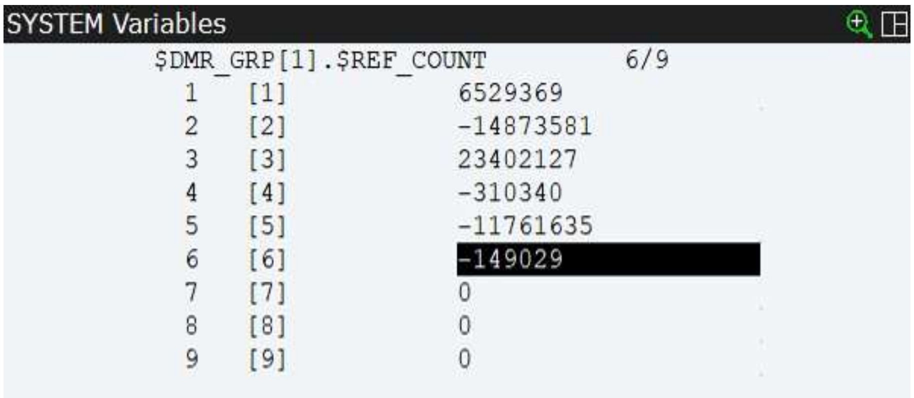


```
SYSTEM Variables
          $DMR_GRP[1].$REF_POS           1/9
            1    [1]            0.000
            2    [2]            0.000
            3    [3]            0.000
            4    [4]            0.000
            5    [5]            0.000
            6    [6]            0.000
            7    [7]            0.000
            8    [8]            0.000
            9    [9]            0.000
```

Source: Image obtained from Roboguide software

3) Return to the menu by pressing PREV and now select ITEM 7 $REF_COUN→ ENTER and note the values of the first step (see in Figure 5.7).

Figure 5.7 Variables in $REF_COUN

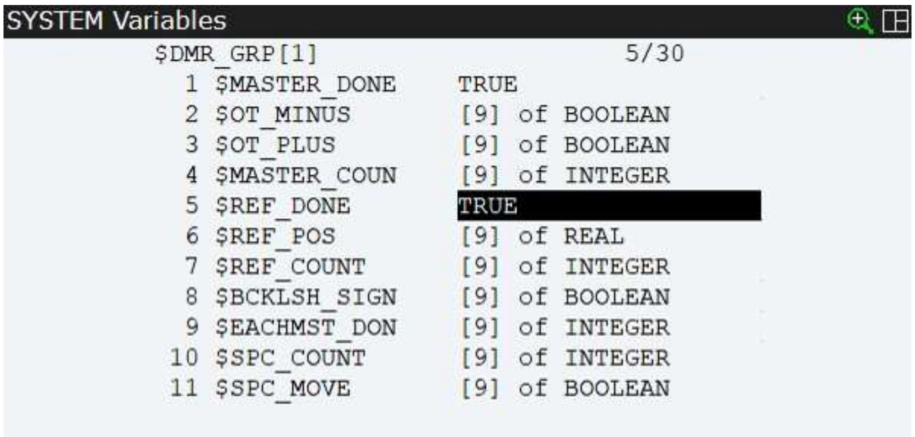


```
SYSTEM Variables
          $DMR_GRP[1].$REF_COUNT         6/9
            1    [1]            6529369
            2    [2]            -14873581
            3    [3]            23402127
            4    [4]            -310340
            5    [5]            -11761635
            6    [6]            -149029
            7    [7]            0
            8    [8]            0
            9    [9]            0
```

Source: Image obtained from Roboguide software

4) Finally, go back one more time by pressing PREV and in ITEM 5 $REF_DONE change the sub variable to TRUE, as in figure 5.8.

Figure 5.8 Variable $RE_DONE set to TRUE



```
SYSTEM Variables
          $DMR_GRP[1]                     5/30
            1 $MASTER_DONE      TRUE
            2 $OT_MINUS         [9] of BOOLEAN
            3 $OT_PLUS          [9] of BOOLEAN
            4 $MASTER_COUN      [9] of INTEGER
            5 $REF_DONE         TRUE
            6 $REF_POS          [9] of REAL
            7 $REF_COUNT        [9] of INTEGER
            8 $BCKLSH_SIGN      [9] of BOOLEAN
            9 $EACHMST_DON      [9] of INTEGER
           10 $SPC_COUNT        [9] of INTEGER
           11 $SPC_MOVE         [9] of BOOLEAN
```

Source: Image obtained from Roboguide software

Return to the main menu by pressing the PREV key twice and go to point 5.2.3.3 Quick master.

5.2.3.2 When mastering is not yet lost

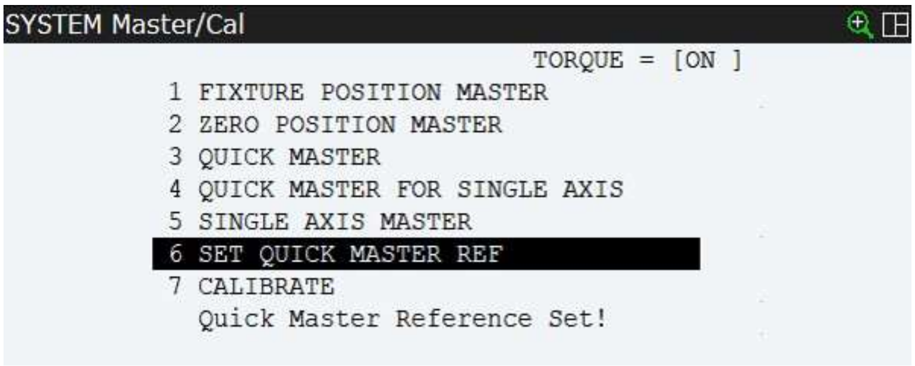
This mastering case is used for two main reasons: because you want to adjust the mechanical zero of the robots, or because a mastering is required for additional calibration marks on it. Whatever the case, a new reference position is created to replace the old one, which directly helps us to perform the QUICK MASTER. To develop this process, we must define the reference position we want:

A. New Reference Position (With additional marks or by adjusting the mechanical zero)

Here you must move the robot axis by axis with coordinates or JOINT movements, this until you reach the additional marks of the robot or until you correctly adjust the current marks. Then save the reference position in SET QUICK MASTER REF (see figure 5.9) by pressing:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 3-MASTER/CAL→ 6-SET QUICK MASTER REF → ENTER → F4 : YES

Figure 5.9 SET QUICK MASTER



```
SYSTEM Master/Cal
                              TORQUE = [ON ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE
       Quick Master Reference Set!
```

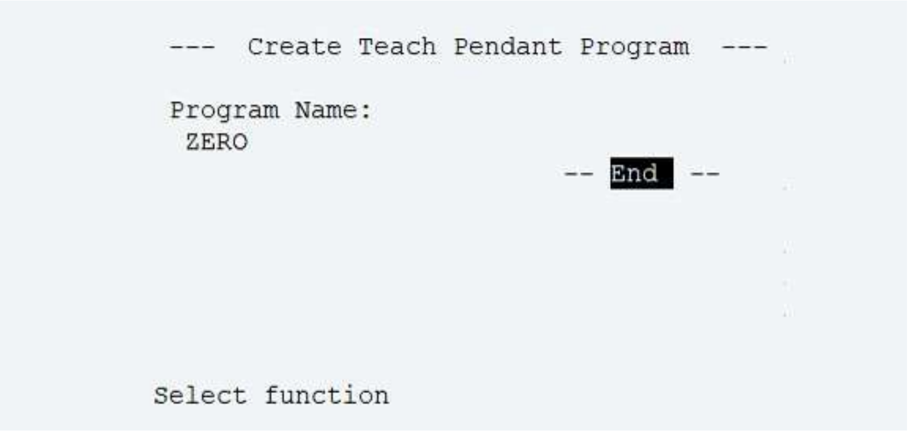
Source: Image obtained from Roboguide software

Now go to point 5.2.3.3 QUICK MASTER.

B. Current Reference Position (Placing the robot on its current marks)

It is always recommended to make a ZERO program to place the robot on its current marks, to do this press SELECT → F2: CREATE → place the name ZERO → ENTER → F3: EDIT (If you need to make configurations to the program, press F2: DETAIL and when finished continue).

Figure 5.10 Creation of the ZERO program

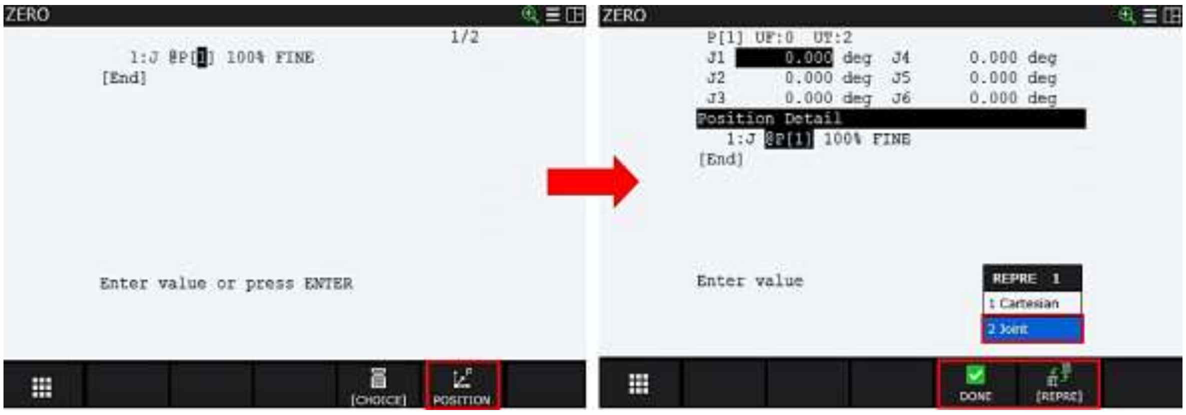


```
---  Create Teach Pendant Program  ---

Program Name:
 ZERO
                         -- End --




Select function
```

Source: Image obtained from Roboguide software

After that record an automatic point by pressing SHIFT+F1: POINT and go through the cursor to the coordinate number of it, there press F5: POSITION, once inside press F5: [REPRE] → 2-JOINT and manually change all coordinates to zero. Finally, press F4: DONE.

Figure 5.11 Position adjustment for a recorded point



```
ZERO
                                   1/2
   1:J @P[1] 100% FINE
  [End]



  Enter value or press ENTER
```

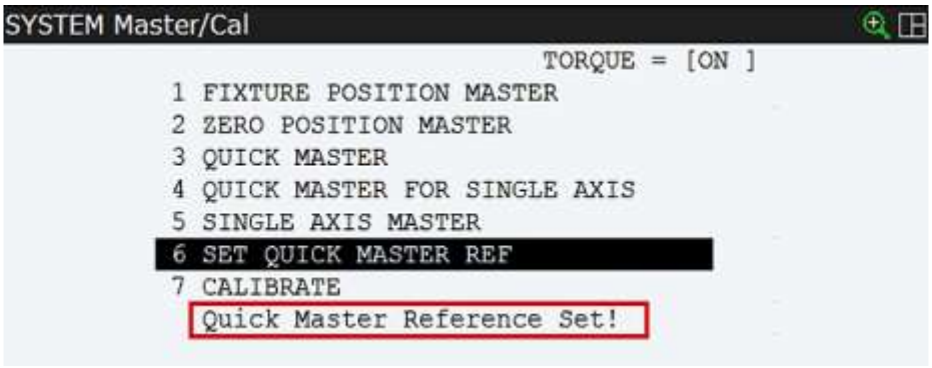
```
ZERO
  P[1] UF:0  UT:2
  J1       0.000 deg  J4      0.000 deg
  J2       0.000 deg  J5      0.000 deg
  J3       0.000 deg  J6      0.000 deg
Position Detail
   1:J @P[1] 100% FINE
 [End]


Enter value
```

Source: Image obtained from Roboguide software

Finally, we record the reference position through the SET QUICK MASTER REF as shown in figure 5.12.

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 3-MASTER/CAL→ 6-SET QUICK MASTER REF → ENTER → F4 : YES

Figure 5.12 Note obtained when choosing SET QUICK MASTER REF.



```
SYSTEM Master/Cal
                          TORQUE = [ON ]
      1 FIXTURE POSITION MASTER
      2 ZERO POSITION MASTER
      3 QUICK MASTER
      4 QUICK MASTER FOR SINGLE AXIS
      5 SINGLE AXIS MASTER
      6 SET QUICK MASTER REF
      7 CALIBRATE
        Quick Master Reference Set!
```

Source: Image obtained from Roboguide software

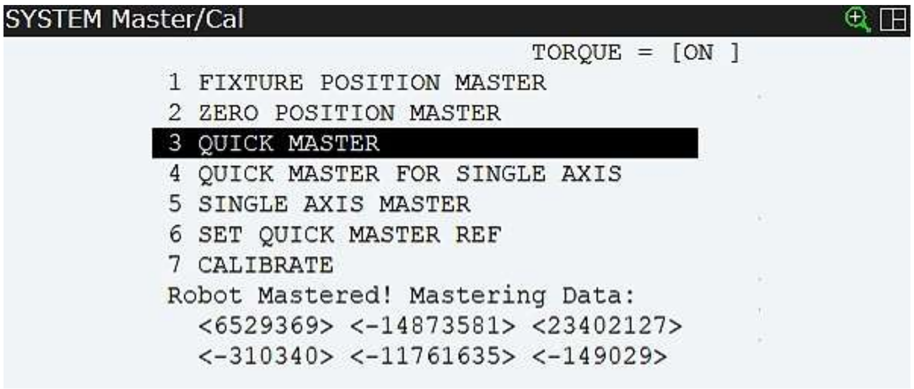
Now go to point 5.2.3.3 Quick master.

5.2.3.3 Quick master

At this point, you have already recorded the reference position in the SET QUICK MASTER (if not, go back to the beginning of the block and do it), in that case, you will most probably get errors like: SRVO-062 and SRVO-075, if they appear, go back to the beginning of point 5.2.1 "Common errors" and correct them. Then follow the next direction to obtain the screen shown in figure 5.13:

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 3-MASTER/CAL→ 3-QUICK MASTER→ ENTER → F4 : YES

Figure 5.13 Robot mastered through QUICK MASTER



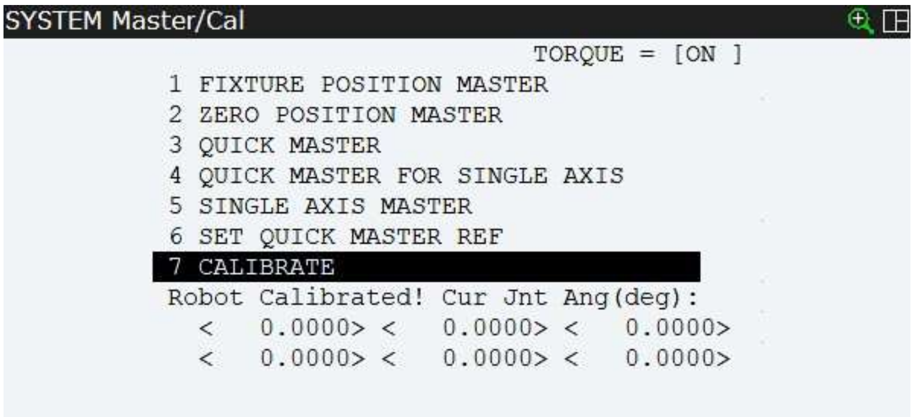
```
SYSTEM Master/Cal
                                   TORQUE = [ON ]
          1 FIXTURE POSITION MASTER
          2 ZERO POSITION MASTER
          3 QUICK MASTER
          4 QUICK MASTER FOR SINGLE AXIS
          5 SINGLE AXIS MASTER
          6 SET QUICK MASTER REF
          7 CALIBRATE
          Robot Mastered! Mastering Data:
            <6529369> <-14873581> <23402127>
            <-310340> <-11761635> <-149029>
```

Source: Image obtained from Roboguide software

Without leaving the menu and once mastered, you must calibrate the robot using so that the mastering is completed correctly, so click on the address shown below to obtain the screen shown in figure 5.14.

7-CALIBRATE→ ENTER → F4 : YES

Figure 5.14 Calibrated robot data



```
SYSTEM Master/Cal
                                   TORQUE = [ON ]
          1 FIXTURE POSITION MASTER
          2 ZERO POSITION MASTER
          3 QUICK MASTER
          4 QUICK MASTER FOR SINGLE AXIS
          5 SINGLE AXIS MASTER
          6 SET QUICK MASTER REF
          7 CALIBRATE
          Robot Calibrated! Cur Jnt Ang(deg):
            <    0.0000> <    0.0000> <    0.0000>
            <    0.0000> <    0.0000> <    0.0000>
```

Source: Image obtained from Roboguide software

The robot will already be fully mastered, so it is recommended to run a ZERO program (see in point 5.2.3.2-point B) and move the robot in a position outside its calibration marks. Then we run the program and make sure that the robot returns correctly to its marks, if it does so properly, the mastering was successful. If not, try the procedure again or run another method for mastering.

Note: when using the SET QUICK MASTER option, a reference position will be created for the robot, however, it is not until we use the QUICK MASTER that the new reference position is replaced by the old one, this causes an offset to be created in the encoder that corrects the error between them, calculating the return deviation in degrees.

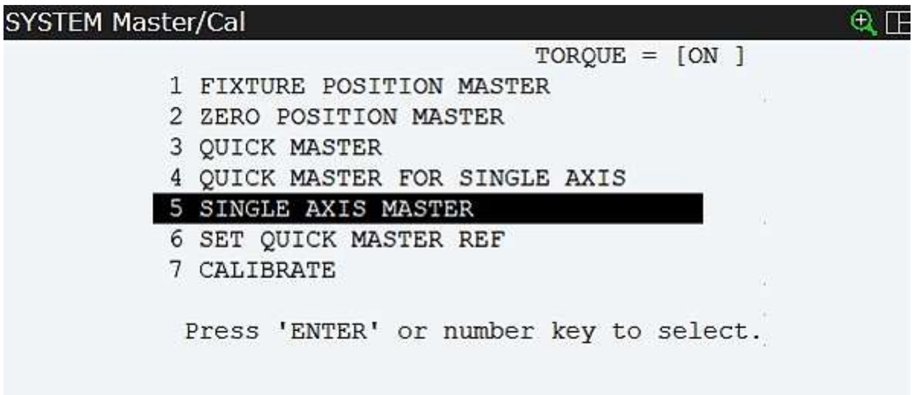
5.2.4 Mastering and calibration “Single axis master”

If you need to master a single axis, two axes or in general less than 6 axes, you can use SINGLE AXIS MASTER. Assuming the current situation of the robot, you will most likely get errors such as: SRVO-062 (for battery wear) and SRVO-075 (for pulses not set), if so, go back to point 5.2.1 "Common errors and fix it, otherwise continue with the following.

To master, follow the direction below to obtain the screen shown in figure 5.15.

MENU → 0-NEXT → 6-SYSTEM → F1: [TYPE] → 3-MASTER/CAL→ 5-SINGLE AXIS MASTER→ ENTER → F4: YES

Figure 5.15 single axis master



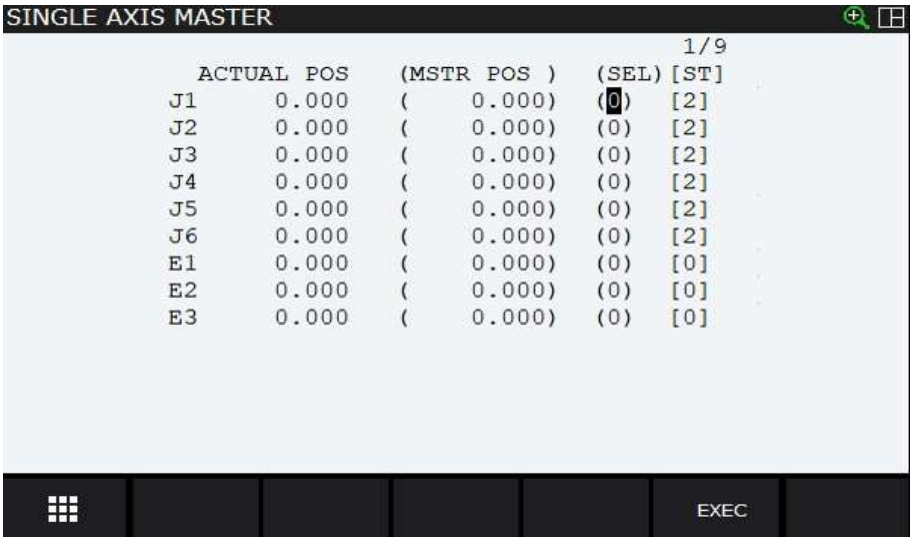
```
SYSTEM Master/Cal
                          TORQUE = [ON ]
  1 FIXTURE POSITION MASTER
  2 ZERO POSITION MASTER
  3 QUICK MASTER
  4 QUICK MASTER FOR SINGLE AXIS
  5 SINGLE AXIS MASTER
  6 SET QUICK MASTER REF
  7 CALIBRATE

   Press 'ENTER' or number key to select.
```

Source: Image obtained from Roboguide software

We will get a configuration screen where we must follow the following aspects that we can configure as needed:

- ACTUAL POS = It shows us the current position of each axis in the robot.
- (MSTR POS) = Shows us the mastered position of each axis in the robot. If any axis is not mastered, its value is manually changed to zero.
- (SEL) = In this section, we must place a one only in the axes that are not mastered.
- [ST] = Here we can see if an axis is mastered. A number 2 appears when it is mastered and a number 0 when it is not.

Figure 5.16 SINGLE AXIS MASTER Configuration Screen



```
SINGLE AXIS MASTER
                                              1/9
      ACTUAL POS     (MSTR POS )   (SEL) [ST]
  J1       0.000     (     0.000)  (0)   [2]
  J2       0.000     (     0.000)  (0)   [2]
  J3       0.000     (     0.000)  (0)   [2]
  J4       0.000     (     0.000)  (0)   [2]
  J5       0.000     (     0.000)  (0)   [2]
  J6       0.000     (     0.000)  (0)   [2]
  E1       0.000     (     0.000)  (0)   [0]
  E2       0.000     (     0.000)  (0)   [0]
  E3       0.000     (     0.000)  (0)   [0]

                                     EXEC
```

Source: Image obtained from Roboguide software

Depending on the above, we must configure the aspects according to our own case and once configured, we must press F5 : EXEC to execute the instruction. After that we just must return to the MASTER/CAL main menu by pressing PREV and then select 7-CALIBRATE→ ENTER → F4 : YES. With this we will have already calibrated the independent axes.

5.3 Backups

A backup is a backup copy where all the programs created with the robot are automatically saved. These programs are stored in the controller's memory and can be copied or moved to a remote memory disk for backup purposes or used to continue programming from software.

5.3.1 Backups AOA

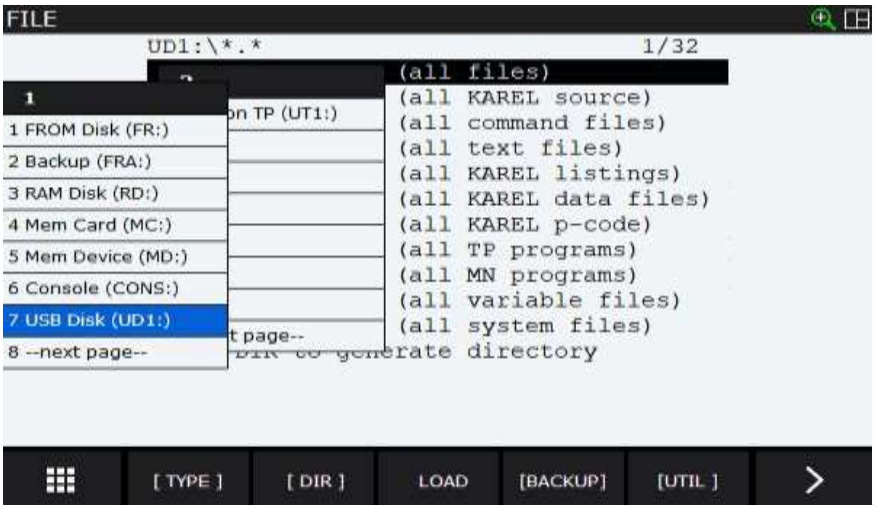
An AOA backup covers all the files in the TP, such as:

- System files
- TP programs
- Applications
- Applications with the TP
- Error Log
- Diagnostics
- Data view files

The steps to make this type of backup are:

1) We must insert the memory that will have our backup in the USB port of our controller or, in the back of the TP, then a warning will come out telling us in which port the memory is connected, UD1 (for the controller) or UT1 (for the TP).
2) We follow the direction: MENU→ 7--FILE →Enter. That will send us to the memory register or directory.
3) Already in the memory register, we press: F5: [UNTIL] → 1-Set device → the memory location that appeared in the warning in step 1 (see in Figure 5.17).

Figure 5.17 Memory register



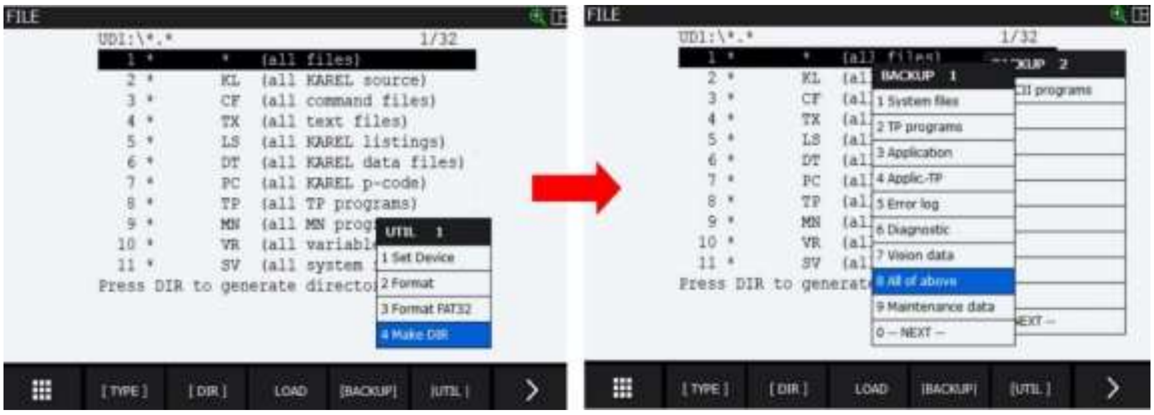
Source: Image obtained from Roboguide software

4) Unless you already have a folder in which to save the information, it can be generated from the memory register (see Figure 5.18) by pressing F5:[UNTIL] → 4- Make DIR → Enter the desired name (it is recommended to put the date in the folder name) →Enter.

Note: You can generate the folders or directories at your convenience following step 4, the main idea is that you can navigate between them without any problem.

5) Then press F4:|BACKUP| → 8-All above → YES.

Figure 5.18 1) Generation of a DIR, 2) Backup All of above



```
FILE
UD1:\*.*                          1/32
   1 *      *   (all files)
   2 *      KL  (all KAREL source)
   3 *      CF  (all command files)
   4 *      TX  (all text files)
   5 *      LS  (all KAREL listings)
   6 *      DT  (all KAREL data files)
   7 *      PC  (all KAREL p-code)
   8 *      TP  (all TP programs)
   9 *      MN  (all MN prog
  10 *      VR  (all variabl
  11 *      SV  (all system
Press DIR to generate directo
```

Source: Image obtained from Roboguide software

6) To corroborate the creation of the backup, we press ENTER in "Up one level" and again ENTER in the name of the created folder. The backed-up programs will appear, and you can look for a program that you have made. If it is in the list, the backup is done.

7) When finished, to remove the memory from the controller or the TP, we must press in the TP F5: [UNTIL] → 1-Set device → 1-FROM Disk (FR:) and we will be able to remove it without any problem.

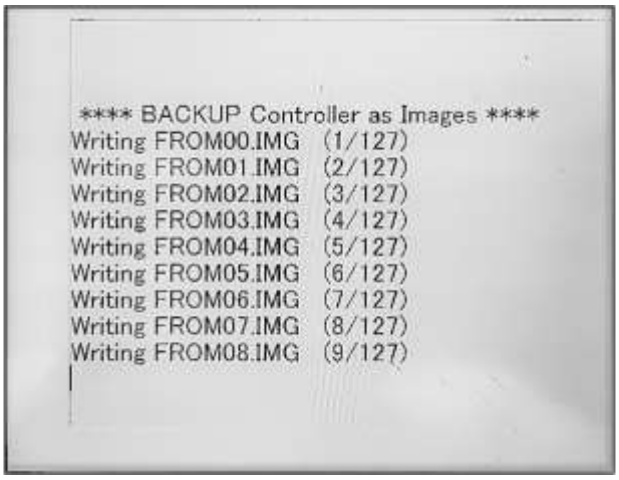
Note: To make a backup AOA, corresponds to make a backup of everything that is in the robot, however, you can make a backup of specific files, this following step 5 and instead of pressing |BACKUP| select what you want, then you can press YES to accept files one by one or press ALL for all, after that follow with steps 6 and 7.

5.3.2 Image backups

Unlike an AOA backup, an image backup is a backup of everything on the controller's computer. Basically, it backs up system files that can faithfully reproduce the data they contain. With a full image backup, everything from boot information, declared tools, other data, etc. is protected. To perform a full image backup, the following steps are performed:

1) The first four steps of point 5.3.1 "AOA backups" are performed.
2) Then click on: F4: |BACKUP| → 0-NEXT → 2-Image backup → ENTER → 1- Current Directory → OK.
3) A cold start is made (turn the controller switch to its OFF state, wait about one minute and turn it back to its ON state). When the teach pendant is fully powered up, you will see a loading screen which will be performing the backup, file by file as in figure 5.19.

Figure 5.19 Loading screen for an image backup



```
**** BACKUP Controller as Images ****
Writing FROM00.IMG  (1/127)
Writing FROM01.IMG  (2/127)
Writing FROM02.IMG  (3/127)
Writing FROM03.IMG  (4/127)
Writing FROM04.IMG  (5/127)
Writing FROM05.IMG  (6/127)
Writing FROM06.IMG  (7/127)
Writing FROM07.IMG  (8/127)
Writing FROM08.IMG  (9/127)
```

Source: Elaborated by the work team

4) When finished, the robot will finish its normal start and to remove the memory from the controller or the TP we will have to press:

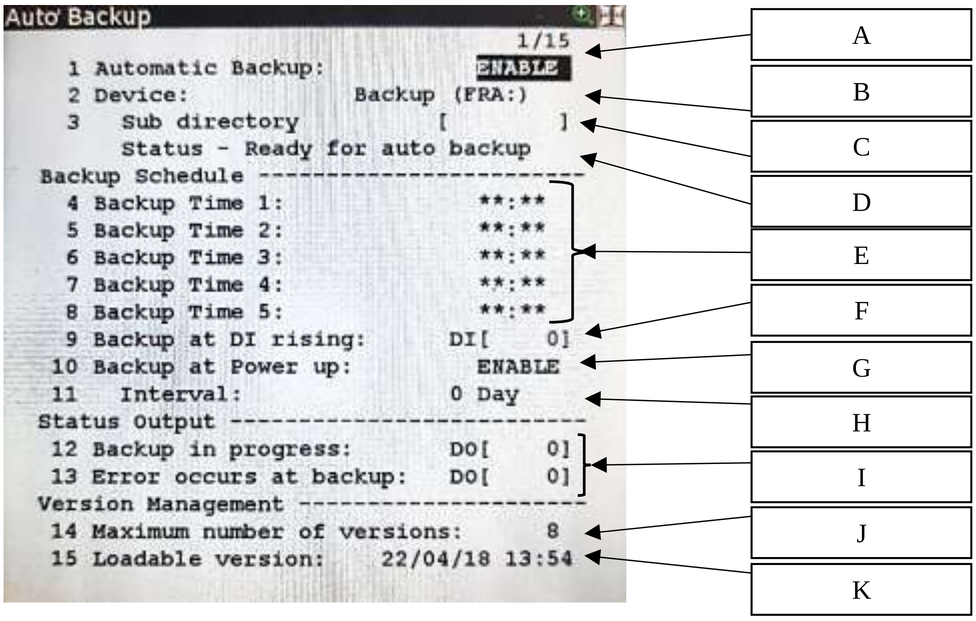
F5: [UNTIL] → 1-Set device → 1-FROM Disk (FR:)

5.3.3 Automatic backups

An automatic backup is a copy that is made within the robot and is loaded from time to time as determined by us, and the information it saves is essential for image backups. Each of the options offered by this type of backup is configurable and important (see Figure 5.20). The direction to find it is as follows:

MENU → 7-FILE → F:1[TYPE] → 4-Auto Backup

Figure 5.20 Automatic backup menu



```
Auto Backup
                                         1/15
  1 Automatic Backup:                  ENABLE
  2 Device:              Backup (FRA:)
  3   Sub directory          [           ]
      Status - Ready for auto backup
Backup Schedule ----------------------------
  4 Backup Time 1:                     **:**
  5 Backup Time 2:                     **:**
  6 Backup Time 3:                     **:**
  7 Backup Time 4:                     **:**
  8 Backup Time 5:                     **:**
  9 Backup at DI rising:        DI[     0]
 10 Backup at Power up:            ENABLE
 11   Interval:                 0 Day
Status Output ------------------------------
 12 Backup in progress:         DO[     0]
 13 Error occurs at backup:     DO[     0]
Version Management -------------------------
 14 Maximum number of versions:        8
 15 Loadable version:       22/04/18 13:54
```

Source: Elaborated by the work team

A. Whenever this section contains the ENABLE option is when the backups will be performed, otherwise nothing will be performed.

B. Device where the backup data will be stored.

C. Sub-addresses that specify the location of the backup.

D. Displays the status of the equipment.

E. Here you can determine the schedules in which the backup will be performed. It contains up to 5 slots in which we can specify different schedules.

F. If the DI specifically contains a 1, it tells us that the backup is being performed, otherwise not.

G. When enabled, this option allows the backup to be performed every time the robot has power according to the time interval in item H.

H. Here you configure the time interval to perform a backup, it can be done per day, hour, or minutes. In addition, when the option of point G is enabled, the following points must be considered:

1. The factory defined interval is 7 days, i.e., once the backup has been made, even if the robot is restarted, a new copy will not be made until these 7 days have passed.
2. If the current date coincides with the date of the last backup version, the backup will not be made.
3. If in the option of interval, we place a 0, a backup copy will be made every time we turn on the robot, this does not consider point 2.

I. Displays the backup status. 1 is when the action is being performed and 0 when it is not.

J. This option configures the number of backup versions that can be backed up. This version of the robot contains up to 8 available slots.

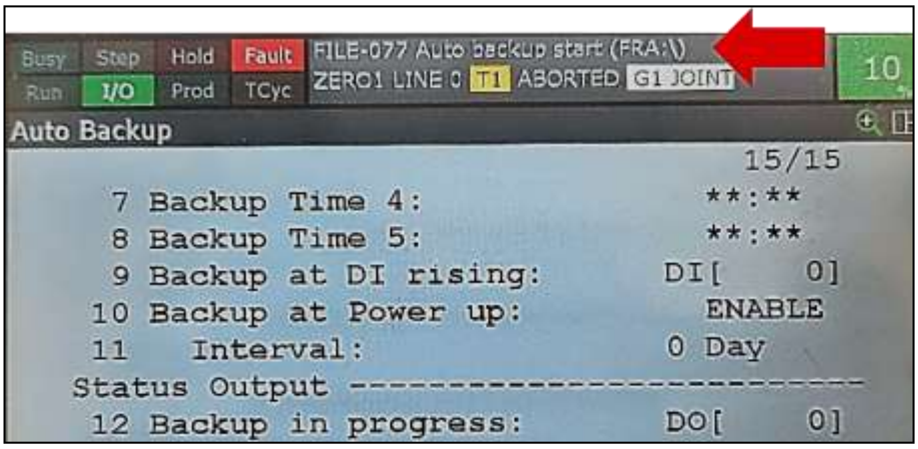
K. It shows us the last version of the backup, however, we can go back to previous versions by pressing the CHOICE key.

Note: If you have too many versions within the automatic backup, you can clear the list by pressing the INIT_DEV → YES option.

5.3.3.1 Creating an automatic backup

According to FANUC Robotics Iberic S.L., the automatic backup is performed as soon as one or more conditions that we could see in section 5.3.3 are satisfied or declared (FANUC Robotics Ibérica S. L., 2018). This version of the robot allows different actions to be performed normally as it sends the backup in the background and marks it in the TP as the warning "FILE-077 Auto backup start (FRA:FRA:\\)" (see in Figure 5.21).

Figure 5.21 Alarm for automatic backups "FILE-077 Auto backup start (FRA: \\)"



```
FILE-077 Auto backup start (FRA:\)
ZERO1 LINE 0 T1 ABORTED G1 JOINT                10
Auto Backup
                                            15/15
     7 Backup Time 4:                    **:**
     8 Backup Time 5:                    **:**
     9 Backup at DI rising:            DI[     0]
    10 Backup at Power up:               ENABLE
    11    Interval:                     0 Day
   Status Output -------------------------
    12 Backup in progress:             DO[     0]
```

Source: Elaborated by the work team

At the end, the TP will detect another alarm indicating that the process is finished and will be able to continue with its activities. Although the autobackup in this version allows performing activities while it is being developed, it is recommended to wait for the transition to be completed, since we are given the necessary options to allow us to develop these backups in defined times.

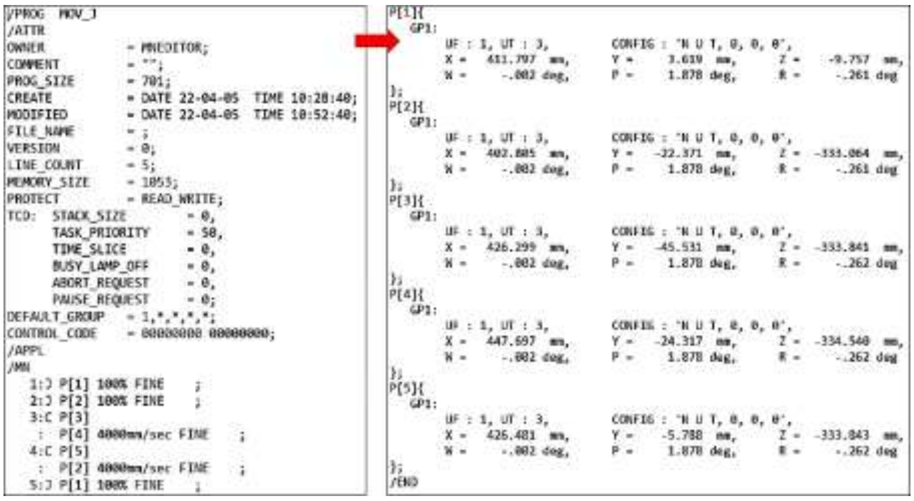
5.3.4 ASCII backups

This is a type of backup that saves the information in text format, for .LS files. The way to perform it is by following the first four steps of the AOA backups, once it is done, click on the button:

F4:|BACKUP| → 0-NEXT → 2 ASCII Programs → ALL (Para todos los programas en general) / YES o NO (Para seleccionar programa por programa)

Once the procedure is done, each of the programs can be visualized by means of some text format software, giving us as a result, each one of the characteristics of the program, the recorded points and their positions, as shown in figure 5.22.

Figure 5.22 Example of an ASCII backup as seen from a notepad



```
/PROG  MOV_J
/ATTR
OWNER           = MNEDITOR;
COMMENT         = "";
PROG_SIZE       = 701;
CREATE          = DATE 22-04-05  TIME 10:28:40;
MODIFIED        = DATE 22-04-05  TIME 10:52:40;
FILE_NAME       = ;
VERSION         = 0;
LINE_COUNT      = 5;
MEMORY_SIZE     = 1053;
PROTECT         = READ_WRITE;
TCD:  STACK_SIZE        = 0,
      TASK_PRIORITY     = 50,
      TIME_SLICE        = 0,
      BUSY_LAMP_OFF     = 0,
      ABORT_REQUEST     = 0,
      PAUSE_REQUEST     = 0;
DEFAULT_GROUP   = 1,*,*,*,*;
CONTROL_CODE    = 00000000 00000000;
/APPL
/MN
   1:J P[1] 100% FINE    ;
   2:J P[2] 100% FINE    ;
   3:C P[3]
    :  P[4] 4000mm/sec FINE    ;
   4:C P[5]
    :  P[2] 4000mm/sec FINE    ;
   5:J P[1] 100% FINE    ;
```

```
P[1]{
   GP1:
        UF : 1, UT : 3,         CONFIG : 'N U T, 0, 0, 0',
        X =   411.797  mm,      Y =     3.619  mm,      Z =    -9.757  mm,
        W =     -.002 deg,      P =     1.878 deg,      R =     -.261 deg
};
P[2]{
   GP1:
        UF : 1, UT : 3,         CONFIG : 'N U T, 0, 0, 0',
        X =   402.885  mm,      Y =   -22.371  mm,      Z =  -333.064  mm,
        W =     -.002 deg,      P =     1.878 deg,      R =     -.261 deg
};
P[3]{
   GP1:
        UF : 1, UT : 3,         CONFIG : 'N U T, 0, 0, 0',
        X =   426.299  mm,      Y =   -45.531  mm,      Z =  -333.841  mm,
        W =     -.002 deg,      P =     1.878 deg,      R =     -.262 deg
};
P[4]{
   GP1:
        UF : 1, UT : 3,         CONFIG : 'N U T, 0, 0, 0',
        X =   447.697  mm,      Y =   -24.317  mm,      Z =  -334.540  mm,
        W =     -.002 deg,      P =     1.878 deg,      R =     -.262 deg
};
P[5]{
   GP1:
        UF : 1, UT : 3,         CONFIG : 'N U T, 0, 0, 0',
        X =   426.481  mm,      Y =    -5.788  mm,      Z =  -333.843  mm,
        W =     -.002 deg,      P =     1.878 deg,      R =     -.262 deg
};
/END
```

Source: Elaborated by the work team

5.3.4 AOA backups from Roboguide

To perform a backup all of above from the ROBOGUIDE software, we have to open a cell and develop the programs we need, or we open a project already made and follow the steps below:

1) From the virtual TP we press MENU → 7-FILE →Enter.

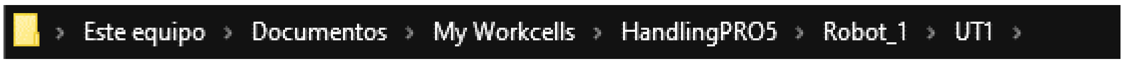
2) Already in the memory register, we press F5: [UNTIL] → 1-Set device → depending on us, we select UD1 (for the controller) or UT1 (for the TP).

3) Folders are generated at your convenience following F5: [UNTIL] → 4-Make DIR → Create a name.

4) Then you press |BACKUP| → 8-All of above → YES (this option warns you that, if there is any other content in the folder where the backup will be created, it will delete it).

5) Finally, to find the folder where the backup was saved so that it can be used to transfer it to the physical robot if needed, you can follow the address in Figure 5.23 on your computer, considering the name of the cell you worked with.

Figure 5.23 Example of direction on a normal computer to find a backup



› Este equipo › Documentos › My Workcells › HandlingPRO5 › Robot_1 › UT1 ›

Source: Elaborated by the work team

6) Depending on whether you saved the backup in the UD1 or UT1 folder, it will save the folder that was created inside on a remote memory, and you will have your backup ready.

Note: In the software there is no option to make image backups, so you can only work with any backup that falls within the range of AOA or ACII. In this case, if we want to dump programs or the AOA to the robot, it can be done through a restore, if we are warned that passing programs from a software to a robot can generate errors.

5.3.6 Dump backup data Robot > PC

After creating a backup from the robot, if we want to open it from the ROBOGUIDE software, we need to open the software and follow the following steps:

1) You start the software and choose HandlingPRO → NEXT.

2) Enter the name of your choice and press NEXT.

3) Choose the option "Create a robot from a file backup" and find the folder where the backup was created from the robot and select it:

a. BACKDATE.DT = For an AOA backup.

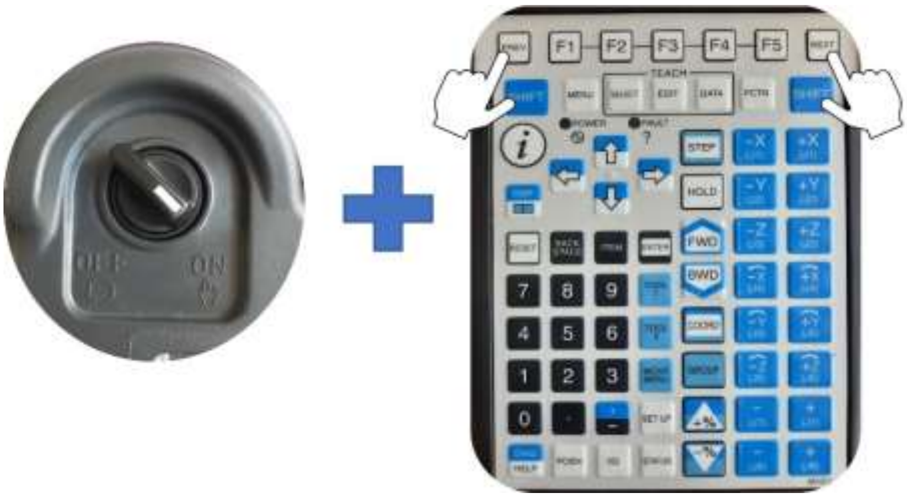
b. FROM00 = For an ISO backup.

4) Finally, press NEXT → NEXT → FINISH. This way you will be able to work with the software from our backup.

5.3.7 Restoring AOA backups

To restore an AOA backup, you need to do a controlled start on the robot, for this it is necessary to have the TP activated in its ON state and press and hold PREV+NEXT (see Figure 5.24) while initializing.

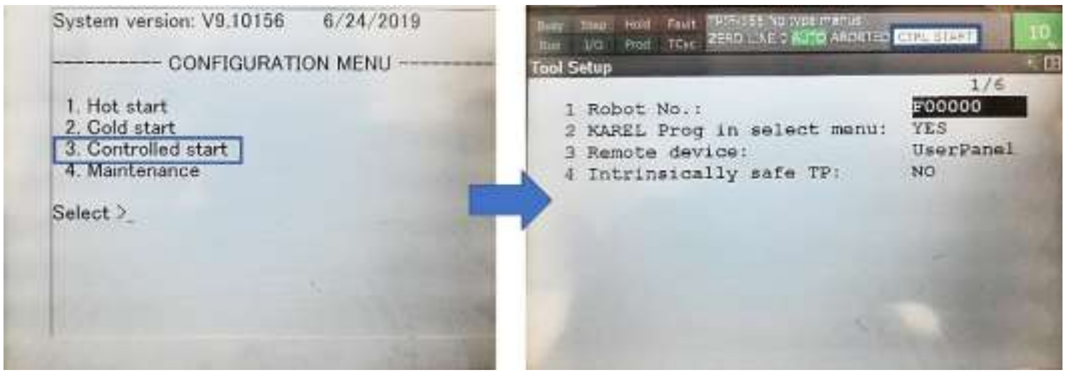
Figure 5.24 Instruction to make a controlled start



Source: Elaborated by the work team

Subsequently, option 4 is selected and this will take us to a screen similar to the one we normally work with; however, CTRL START appears in the upper right corner (see Figure 5.25).

Figure 5.25 Selection of controlled start



```
System version: V9.10156     6/24/2019

---------- CONFIGURATION MENU ----------

 1. Hot start
 2. Cold start
 3. Controlled start
 4. Maintenance

Select >_
```

```
Tool Setup
                                          1/6
 1 Robot No.:                        F00000
 2 KAREL Prog in select menu:        YES
 3 Remote device:                    UserPanel
 4 Intrinsically safe TP:            NO
```

Source: Elaborated by the work team

Now, to restore the backup we need, we place the device that contains it and press MENU → F5: [UNTIL] → 1-Set device → Choose our device. When we find the backup we will have two options to do the restore:

1) First we restore the system files by pressing:

F4:[RESTOR] → 1-SYSTEM FILES → F4:[YES]

Finally, we restore the TP programs.

F4:[RESTOR] → 2-TP Program → F4:[YES]

2) Or, if we want to make the complete restoration in a direct way, you can use:

F4:[RESTOR] → 6-ALL OF ABOVE → F4:[YES]

When you have successfully performed the backup restore you must press the FNCT key on the TP and select the 1-START COLD option, in this step you turn off the robot controller, wait about 30 seconds and turn on again. At full startup, you will see that we will be back to the main screen of our PT (CTRL START no longer must appear at the top).

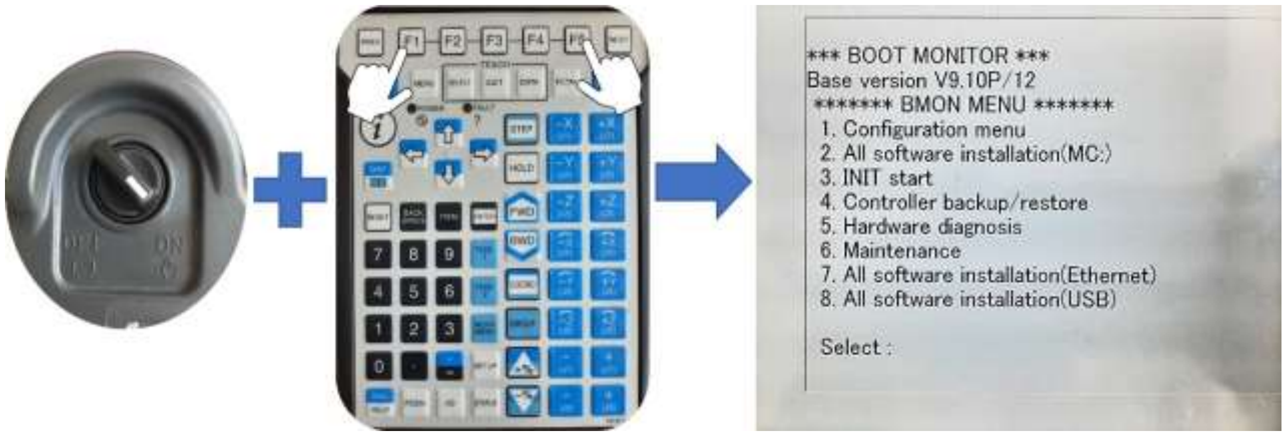
Note: This type of restore can also work to dump data or programs from a PC to the Robot, for this you must make a backup from your PC following the point "BACKUP AOA FROM ROBOGUIDE" and then following the steps in this point.

5.3.8 Restoring image backups

To restore an image backup, it is necessary to access the BOOT MONITOR memory, for this it is necessary to have the TP activated in its ON state and press and hold F1+F5 while starting the robot (see figure 5.26). This will send us to a screen different from the screen we normally work with, in it there will be 8 options to configure the system.

In this case, the option 4.Controler backup/restore → ENTER is selected.

Figure 5.26 Startup to access the BOOT MONITOR memory



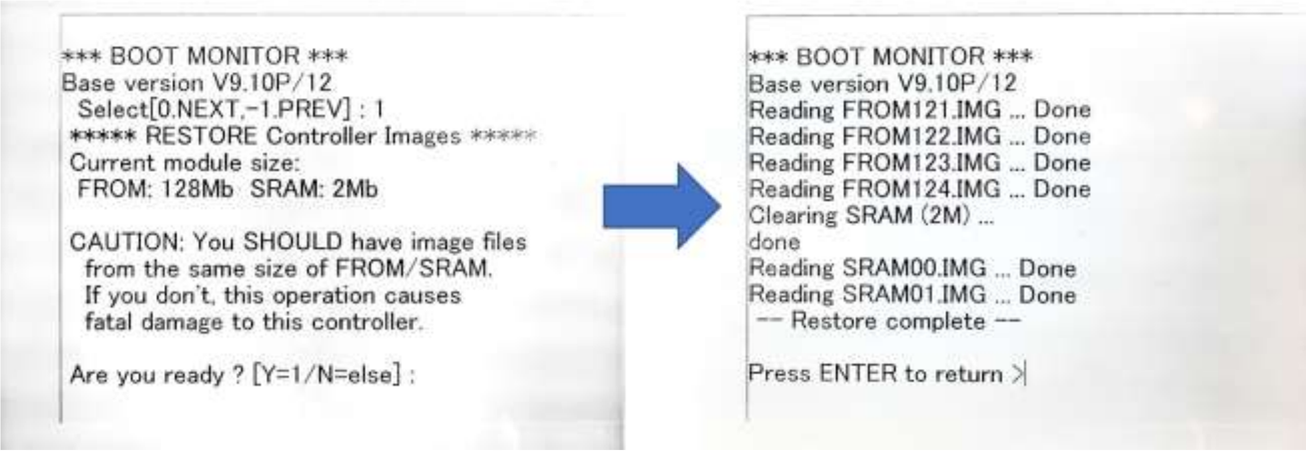
```
*** BOOT MONITOR ***
Base version V9.10P/12
******* BMON MENU *******
 1. Configuration menu
 2. All software installation(MC:)
 3. INIT start
 4. Controller backup/restore
 5. Hardware diagnosis
 6. Maintenance
 7. All software installation(Ethernet)
 8. All software installation(USB)

 Select :
```

Source: Elaborated by the work team

Restore Controller Images → ENTER, this will take us to a menu that shows us the devices connected to the robot. Here we will browse our own account until we find the backup we want to restore. After finding the address we press 1.OK (Current Directory) → ENTER.

A warning screen will appear telling us that, if the FROM and SRAM do not match in size compared to those of the robot, open a fatal error as the original files will be deleted, if you are sure of the action press 1→ ENTER, otherwise press any other number. In the case of accepting, a loading screen will appear where all the backup files are being restored (see in Figure 5.27), we wait for it to finish and the legend Restore complete appears, after that we press ENTER.

Figure 5.27 Restoration start for image backups



```
*** BOOT MONITOR ***
Base version V9.10P/12
 Select[0.NEXT,-1.PREV] : 1
***** RESTORE Controller Images *****
Current module size:
 FROM: 128Mb  SRAM: 2Mb

CAUTION: You SHOULD have image files
  from the same size of FROM/SRAM.
  If you don't, this operation causes
  fatal damage to this controller.

Are you ready ? [Y=1/N=else] :
```

```
*** BOOT MONITOR ***
Base version V9.10P/12
Reading FROM121.IMG ... Done
Reading FROM122.IMG ... Done
Reading FROM123.IMG ... Done
Reading FROM124.IMG ... Done
Clearing SRAM (2M) ...
done
Reading SRAM00.IMG ... Done
Reading SRAM01.IMG ... Done
 -- Restore complete --

Press ENTER to return >
```

Source: Elaborated by the work team

Finally, we will return to the initial screen and press 1.Configuration menu, here we will have multiple options to continue with the robot startup, If we want the startup to proceed normally press 1.Hot start → ENTER, otherwise, if you are looking to restart the equipment by turning the controller off and on again press 2. Cold start → ENTER.

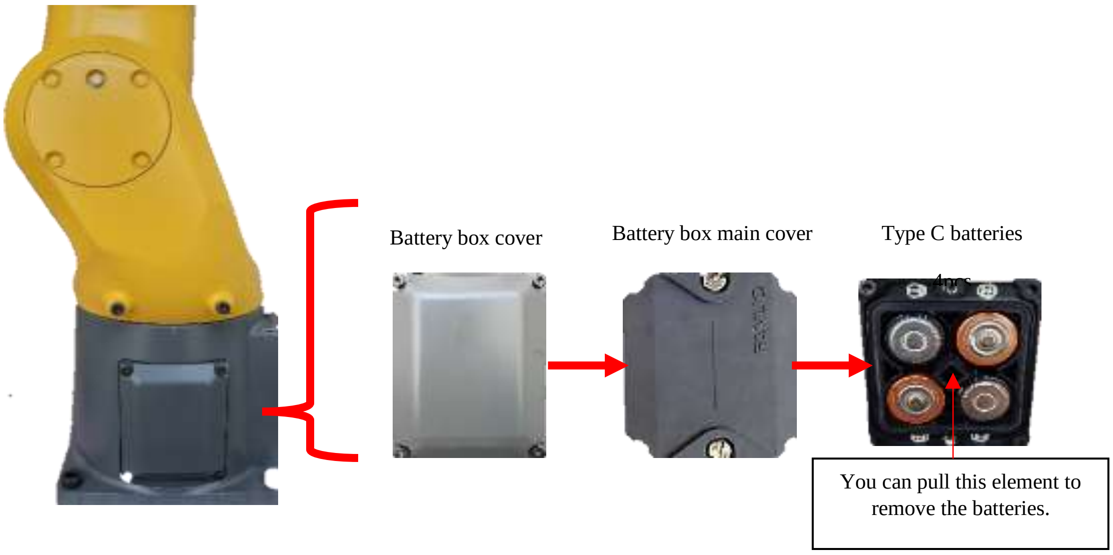
5.4 Additional information

5.4.1 Battery replacement

For a good maintenance of the robot, the internal batteries should be replaced every year if they are in good condition or when the SRVO 065- BLAL alarm (low battery alarm) appears. In addition, they can also be replaced when the "SRVO 062- BZAL" battery voltage drop alarm is triggered on the teach pendant. The replacement procedure is as follows (see figure 5.28):

1. In case you have the robot power supply on, press the emergency stop button to prevent the robot from moving. If not, proceed to the next point.
2. Remove the battery box cover and replace the gasket if necessary.
3. Remove the battery box main by loosening the flat head screws located there.
4. Replace the batteries considering their polarity. You can lengthen the element that is in the center of them to remove them with better facility
5. After replacing the batteries, perform the same procedure in reverse.

Figure 5.28 Composition of the battery box



Source: Elaborated by the work team

Note: if you replace the batteries while the robot has no power, its position data in the encoders will be lost and therefore you will have to remaster it.

5.4.2 Singularity

Singularity is a unique position of axes that can prevent the execution of motions of a robot, i.e., it is infinite solutions for a motion that the robot does not know how to manage. The singularity described in a simpler way, is a point in the robot trajectory that can generate an unexpected rotation of the wrist when an infinite number of angles or turns in the minor axes (4,5,6) generate the same set of Cartesian data, to complete the movement, the controller makes the trajectory calculation and can cause acceleration in the robot. When this failure occurs, it is necessary to move the affected axis in JOINT coordinates (10 degrees approx.) either positively or negatively and re-record the point that is causing such singularity.

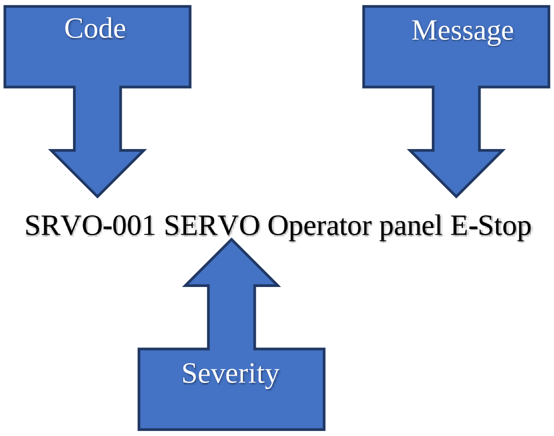
5.4.3 Fault diagnostics

5.4.3.1 Error Codes

Error codes are alarms that indicate that the robotic equipment has entered an error state and can appear at any time due to specific situations. These codes are made up of three important parts as shown in Figure 5.29.

- Code: Describes the type of alarm and its assignment number.
- Severity: Type of severity that sets the alarm.
- Message: Brief description of the alarm.

Figure 5.29 Error codes composition



Source: (FANUC Robotics Ibérica S. L., 2018)

We can identify the severity of each alarm according to its contraction, through Table 5.1, where .L and .G., is the local scope (only applies to the program that generated the alarm) or global scope (applies to all programs) respectively:

Table 5.1 Severity according to the activated alarm

Alarm severity	Program status	Robot status	Robot servo voltages
NONE (ninguna)	No stopping	No stopping	No shutdown
WARN			
PAUSE ".L/.G"	It stops	Decelerates and stops	
STOP ".L/.G"			
SERVO		Stops immediately	Turns off
ABORT ".L/.G"	Forced to stop	Decelerates and stops	No shutdown
SERVO 2		Stops immediately	Turns off
SYSTEM			

Source: (FANUC Robotics Ibérica S. L., 2018)

Considering the above information, the errors can be divided into two: robot errors and external errors. These are detailed below.

Robot errors

- Hardware= Errors that identify
  - Emergencies: such as activated emergency stops, deadman switch, etc.
    Example. SRVO-002 SERVO Teach pendant E-stop (Indicates that the E-stop button on the PT is pressed)

  - Breakdowns: such as faulty wiring, low battery level, etc.
    Example. SRVO-065 WARN BLAL alarm (alarm detecting low battery level of the encoders)

- Software = Logical faults in programming.
  - Example. MOTN-023 In singularity (The robot cannot move because it is in a singularity state)

External errors

- Hardware=Signals emitted by the external emergency board wiring.
  - Example. SRVO-007 SERVO External emergency stops (External emergency signal triggered)

- Software = Errors in UI input signals coming from PLCs.
  - Example. SRVO-037 SERVO IMSTP input (Group:%d) (*IMSTP signal of the UI input set to OFF status)

5.4.3.2 Most common error codes

The R-30iB controller contains several error codes to identify the problem when an alarm is triggered. Depending on the error types, there is a total number of error codes per error (e.g., for SRVO type errors there are 497 errors recorded). Of the most common, according to Robotics Iberic S.L. are:

- **SRVO:**

SRVO-001 Operator panel E-Stop.

- Cause: The emergency stop button on the controller has been pressed.
- Solution: Release the button by turning it clockwise and press RESET.

SRVO-002 Teach pendant E-stop.

- Cause: The emergency stop button of the teach pendant has been pressed.
- Solution: Release the button by turning it clockwise and press RESET.

SRVO-003 Deadman switch released.

- Cause: The deadman switch of the teach pendant is not activated and the console selector is in the ON state.
- Solution: Considering that the deadman switch has three positions, press it to the second position and hold it there, then press the RESET key. If the above does not work, replace the TP.

SRVO-004 Fence open.

- Cause: The security fence is open.
- Solution: Close the security fences after confirming that there is no user inside the fence, then press the RESET key on the TP.

SRVO-005 Robot overtravel.

- Cause: The robot exceeded the maximum range of motion for an axis, triggering a hardware limit sending the robot to an overtravel state.
- Solution: Press from the TP, SHIFT + RESET and move the axis out of overtravel, do this with low speeds.

SRVO-006 Hand broken.

- Cause: The arm safety is broken or the HBK signal of the robot connection cable is equal to 0V. In either case the tool collision sensor was activated.
- Solution: Press from the TP, SHIFT + RESET and move the robot out of the collision.

SRVO-007 External emergency stops.

- Cause: An external emergency stop button was pressed.
- Solution: Release the button and press RESET on the TP.

SRVO-015 System overheat.

- Cause: The controller has cooling problems, so the temperature in the control unit exceeds the specified temperature.
- Solution: Identify the cause if the ambient temperature exceeds 45°C try to reduce this temperature. Check if the fan motor is working, if not, check the wires to which it is connected and replace the parts if necessary.

SRVO-027 Robot not mastered (Group:%d).

- Cause: An attempt was made to calibrate the robot, however, there is no mastering, or the current mastering is not complete.
- Solution: Master the robot by following one of the methods shown in this manual or by pressing MENU>6-SYSTEM>Master/Cal on the TP.

SRVO-031 User servo alarm (Group:%d).

- Cause: A SERVO type alarm, configured by the user, has been activated.
- Solution: Check the cause of the alarm, then press RESET.

SRVO-033 Robot not calibrated (Group:%d).

- Cause: An attempt was made to set a calibration setpoint in "Quick mastering" but the calibration is not yet complete.
- Solution: Perform robot calibration to reset the robot values.

SRVO-034 Ref pos not set (Group:%d).

- Cause: A quick mastering was attempted; however, the necessary reference point has not been defined.
- Solution: Program the reference position for fast mastering.

SRVO-037 IMSTP input (Group:%d).

- Cause: the *IMSTP signal is in the OFF state.
- Solution: Turn on or try to force the IMSTP signal ON (this action is typical with paint software). If the fault persists, check the cause of this signal input. To avoid failures related to UI user inputs, set the option "ENABLE UI SIGNALS" as false, this option is found in MENU, 0-NEXT, 6-SYSTEM, 6-CONFIG (approx. in item 7).

SRVO-050 Collision Detect alarm (Group:%d Axis:%d).

- Cause: An excessively large disturbance torque is estimated by the servo software (a collision was detected).
- Solution: For not too strong collisions, press SHIFT + RESET and move the robot out of the collision, if this does not work, refer to additional reset procedure in the controller manual or on the Training CD.

SRVO-062 BZAL alarm (Group:%d Axis:%d).

- Cause: When this alarm is triggered it indicates that the batteries in the mechanical unit are completely exhausted and therefore the absolute position data of the encoders is lost. In case the problem is not in the voltage level of the batteries, the fault can be generated by an encoder or the wiring of an encoder.
- Solution: If the failure was generated by the low voltage level in the batteries of the mechanical unit, replace the batteries and perform a mastering (you can be guided by this manual, in section 13). On the other hand, if the problem was generated by any encoder, follow a manual to perform a replacement.

SRVO 065- BLAL alarm (Group:%d Axis:%d)

- Cause: A warning is generated telling us that the voltage level of the batteries in the mechanical unit is at the lowest level allowed.
- Solution: Replace the batteries as soon as possible.

Note: When this alarm appears, replace the batteries immediately while the robot is powered on, otherwise, if the batteries are not replaced in time, the error will evolve to error SRVO-062 BZAL alarm, losing the absolute position of the encoders and you will have to master the robot again to correct it.

SRVO-075 Pulse not established (Group:%d Axis:%d).

- Cause: The absolute position of the encoder has not yet been established (this error is typical when mastering).
- Solution: Remove other alarms found in the TP, move each of the robot axes +/-10° with JOINT coordinates so that the encoders detect their zero mark and when finished, press the RESET key.

SRVO-105 Door open or E.Stop

- Cause: The controller door is open or the TP specification in relation to the controller is not correct.
- Solution: Try to perform robot operations always with the controller door closed.

SRVO -214 6ch amplifier fuse blown (R:1)

- Cause: One of the fuses of the 6-axis servo amplifier inside the controller is blown.
- Solution: Replace the FS2 fuse located on the servo amplifier board.

- **INTP:**

INTP-251 (%s^4, %d^5) Invalid uframe numer

- Cause: The user frame used for the operation is invalid.
- Solution: Identify and change the frame to the frame used for the programming you want to execute by pressing SHIFT + COORD. Otherwise, you can follow the steps in section 9.3.4 in this manual.

NTP-251 (%s^4, %d^5) Invalid utool numer

- Cause: The user tool used for the operation is invalid.
- Solution: Identify and change the tool to the tool used for the programming you want to execute by pressing SHIFT + COORD. Otherwise, you can follow the steps in section 9.3.4 in this manual.

INTP-252 User frame number mismatch

- Cause: The selected user coordinate system does not match the one saved in the positions of the program being executed.
- Solution: Query the user coordinate system of the current program positions and set it by pressing SHIFT + COORD.

INTP-253 Tool frame number mismatch

- Cause: The selected tool coordinate system does not match the one saved in the positions of the program being executed.
- Solution: Consult the tool coordinate system of the positions of the running program and set it by pressing SHIFT + COORD.

- **MOTN:**

MOTN-018 Position not reachable (G:%D^2)

- Cause: The robot will not be able to reach the programmed position.
- Solution: This may be due, among other causes, to a wrong calculation of PR[ ] registers, to an impossible linear movement or to accidentally advanced/delayed program steps with the cursor. In this order of priority: Check the coordinates of the position for which this alarm occurs; check and review the program; if necessary, reset the system.

MOTN-019 In singularity

- Cause: The robot is about to reach a position with singularity.
- Solution: Try to maneuver the robot to prevent it from moving in such a position.

MOTN-023 In singularity

- Cause: The robot is in a position with singularity.
- Solution: Reprogram and perform maneuvers to prevent the robot from reaching that position.

MOTN-136 STOP Circular Points too close(L:%d^5)

- Cause: The points of a circular path have been programmed too close and the radius of the circle cannot be calculated correctly.
- Solution: Reprogram such positions.

- **SYST:**

SYST-003 TP is enabled

- Cause: An attempt was made to perform an operation, but it is not allowed because the TP is enabled.
- Solution: Disable the TP by changing the status selector to OFF, press RESET and try your operation again.

SYST-005 UOP is the master device

- Cause: An operation was attempted but is not allowed because the operation is configured to be maneuvered with a remote device.
- Solution: In MENU>0-Next>6-SYSTEM>F1-Type>6-Config, set the boot mode to REMOTE.

SYST-011 Failed to run task

- Cause: The system did not execute the selected program, it could be because a different operation was being executed or because the cycle start button was pressed from the controller, without the system being configured in AUTO mode.
- Solution: Press from the TP, FCTN>1-ABORT (ALL) to abort any operation.

SYST-016 ENBL signal is off

- Cause: The ENBL signal is OFF and does not allow the system to start.
- Solution: Turn the signal ON.

References

FANUC Robotics Ibérica S. L. (2018). ROBÓTICA PROGRAMACIÓN FANUC. *SEAT*. http://www.infoplc.net/files/descargas/fanuc_robotica/infoPLC_net_ManualFANUCR-J3iB.pdf

Limón Molina, G. M. (2008). Módulo de calidad controlado por la tecnología DEVICE NET. *Universidad Autónoma de Baja California. Facultad de Ingeniería.* https://hdl.handle.net/20.500.12930/3002

Thanks

The authors of the book are grateful for the support received from the Technological Institute of Higher Studies of Jocotitlán and the University of Ixtlahuaca CUI/Faculty of Engineering, for giving us the use of their facilities for the development of this work.

[[Title in Times New Roman and Bold No. 14 in English and Spanish]

Surname (IN UPPERCASE), Name 1st Author†*, Surname (IN UPPERCASE), Name 1st Coauthor, Surname (IN UPPERCASE), Name 2nd Coauthor and Surname (IN UPPERCASE), Name 3rd Coauthor

Institution of Affiliation of the Author including dependency (in Times New Roman No.10 and Italics)

International Identification of Science - Technology and Innovation

ID 1st Author: (ORC ID - Researcher ID Thomson, arXiv Author ID - PubMed Author ID - Open ID) and CVU 1st author: (Scholar-PNPC or SNI-CONACYT) (No.10 Times New Roman)

ID 1st Coauthor: (ORC ID - Researcher ID Thomson, arXiv Author ID - PubMed Author ID - Open ID) and CVU 1st coauthor: (Scholar or SNI) (No.10 Times New Roman)

ID 2nd Coauthor: (ORC ID - Researcher ID Thomson, arXiv Author ID - PubMed Author ID - Open ID) and CVU 2nd coauthor: (Scholar or SNI) (No.10 Times New Roman)

ID 3rd Coauthor: (ORC ID - Researcher ID Thomson, arXiv Author ID - PubMed Author ID - Open ID) and CVU 3rd coauthor: (Scholar or SNI) (No.10 Times New Roman)

(Report Submission Date: Month, Day, and Year); Accepted (Insert date of Acceptance: Use Only ECORFAN)

Citation: First letter (IN UPPERCASE) of the Name of the 1st Author. Surname, First letter (IN UPPERCASE) of the First Coauthor's Name. Surname, First letter (IN UPPERCASE) of the Name of the 2nd Co-author. Surname, First letter (IN UPPERCASE) of the Name of the 3rd Co-author. Last name

Institutional mail [Times New Roman No.10]

First letter (IN UPPERCASE) of the Name Publishers. Surnames (eds.) Title of the Books [Times New Roman No.10], Selected Topics of the corresponding area © ECORFAN- Subsidiary, Year.

Abstract (In English, 150-200 words)

Text written in Times New Roman No.12, single space

Keywords (In English)

Indicate 3 keywords in Times New Roman and Bold No. 12

1 Introduction

Text in Times New Roman No.12, single space.

General explanation of the subject and explain why it is important.

What is your added value with respect to other techniques?

Clearly focus each of its features

Clearly explain the problem to be solved and the central hypothesis.

Explanation of sections Chapter.

Development of headings and subheadings of the chapter with subsequent numbers

[Title No.12 in Times New Roman, single spaced and bold]

Products in development No.12 Times New Roman, single spaced.

Including graphs, figures and tables-Editable

In the Chapter content any graphic, table and figure should be editable formats that can change size, type and number of letter, for the purposes of edition, these must be high quality, not pixelated and should be noticeable even reducing image scale.

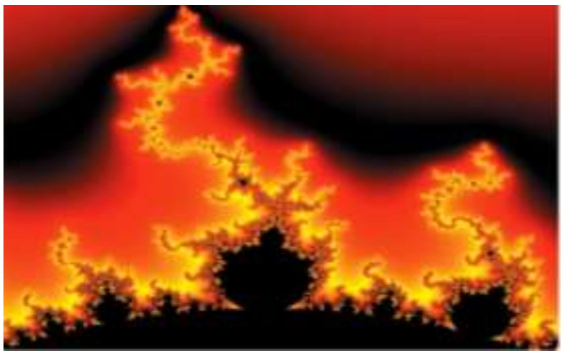
[Indicating the title at the bottom with No.10 and Times New Roman Bold]

Table 1.1 Title

Variable	Descripción	Valor
V_V	Volumen de Venta	20000
P_V	Postura de venta	490.61
V_C	Volumen de Compra	20000
P_c	Postura de Compra	485.39
P^{Uh}	Precio último Hecho	491.61
V_o	Volumen Operado	1241979
P_u	Precio/Utilidad	0
P^{VL}	Precio/Valor Libro	0
U_a	Utilidad p/Acción	0
V^{La}	Valor Libro p/Acción	0

Source (in italics)
Should not be images-everything must be editable.

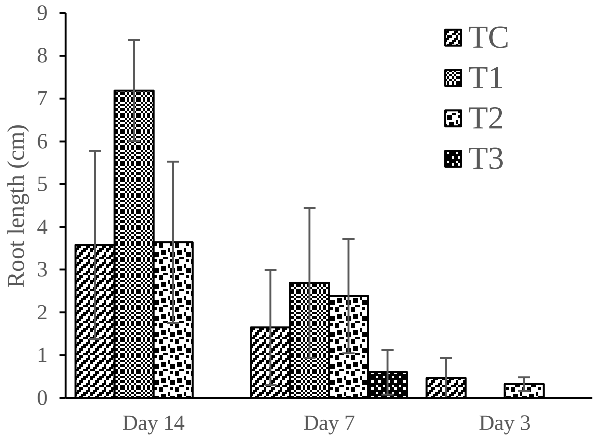
Figure 1.1 Title



Source (in italics)

Should not be images-everything must be editable.

Graphic 1.1 Title



Source (in italics)

Should not be images-everything must be editable.

Each chapter shall present separately in **3 folders**: a) Figures, b) Charts and c) Tables in .JPG format, indicating the number and sequential Bold Title.

For the use of equations, noted as follows:

$$\int_{\boldsymbol{lim^{-1}}}^{\boldsymbol{lim^{1}}} = \int \frac{\boldsymbol{lim^{1}}}{\boldsymbol{lim^{-1}}} = \left[\frac{\mathbf{1}\,(-\mathbf{1})}{\boldsymbol{lim}}\right]^{\mathbf{2}} = \frac{(\mathbf{0})^{\mathbf{2}}}{\boldsymbol{lim}} = \sqrt{\boldsymbol{lim}} = \mathbf{0} = \mathbf{0} \rightarrow \propto \tag{1}$$

Must be editable and number aligned on the right side.

Methodology

Develop give the meaning of the variables in linear writing and important is the comparison of the used criteria.

Results

The results shall be by section of the Chapter.

Annexes

Tables and adequate sources

Thanks

Indicate if they were financed by any institution, University or company.

Conclusions

Explain clearly the results and possibilities of improvement.

References

Use APA system. Should not be numbered, nor with bullets, however if necessary numbering will be because reference or mention is made somewhere in the Chapter.

Use Roman Alphabet, all references you have used must be in the Roman Alphabet, even if you have quoted an Chapter, book in any of the official languages of the United Nations (English, French, German, Chinese, Russian, Portuguese, Italian, Spanish, Arabic), you must write the reference in Roman script and not in any of the official languages.

Technical Specifications

Each chapter must submit your dates into a Word document (.docx):

Books title
Chapter title
Abstract
Keywords

Proceedings sections, for example:

1. Introduction
2. Description of the method
3. Analysis from the regression demand curve
4. Results
5. Thanks
6. Conclusions
7. References

Author Name (s)
Email Correspondence to Author
References

Intellectual Property Requirements for editing:

- Authentic Signature in Color of Originality Format Author and Coauthors
- Authentic Signature in Color of the Acceptance Format of Author and Coauthors

Reservation of Editorial Policy

ECORFAN Books reserves the right to make the editorial changes required to adapt the Scientific Work to the Editorial Policy of ECORFAN Books. Once the Scientific Work is accepted in its final version, ECORFAN Books will send the author the proofs for its revision. ECORFAN® will only accept the correction of errata and errors or omissions arising from the publication process of the journal, reserving in its entirety the copyright and content dissemination. No deletions, substitutions or additions that alter the formation of the Scientific Work will be accepted.

Code of Ethics - Good Practices and Declaration of Solution to Editorial Conflicts

Declaration of Originality and unpublished character of the Scientific Work, of Authorship, on the obtaining of data and interpretation of results, Acknowledgments, Conflict of interests, Assignment of rights and distribution.

The ECORFAN-Mexico, S.C Directorate asserts to the Authors of the Scientific Work that its content must be original, unpublished and of Scientific, Technological and Innovation content to be submitted for evaluation.

The Authors signing the Scientific Work must be the same that have contributed to its conception, realization and development, as well as the obtaining of data, interpretation of the results, its writing and revision. The Correspondent Author of the proposed Scientific Work will request the form that follows. Title of the Scientific Work:

– The sending of a Scientific Work to ECORFAN Books emanates the commitment of the author not to submit it simultaneously to the consideration of other serial publications for it must complement the Format of Originality for its Scientific Work, unless it is rejected by the Arbitration Committee, may be withdrawn.

– None of the data presented in this Scientific Work has been plagiarized or invented. The original data are clearly distinguishable from those already published. And you have knowledge of the test in PLAGSCAN if a level of plagiarism is detected Positive will not proceed to arbitrate.

– References are cited on which the information contained in the Scientific Work is based, as well as theories and data from other previously published Scientific Works.

– The authors sign the Authorization Form for their Scientific Work to be disseminated by means that ECORFAN-Mexico, S.C. in its Holding Mexico consider relevant for the dissemination and dissemination of its Scientific Work by giving up its Scientific Work Rights.

– The consent of those who have provided unpublished data obtained by verbal or written communication has been obtained, and such communication and authorship are adequately identified.

– The Author and Co-Authors who sign this work have participated in its planning, design and execution, as well as in the interpretation of the results. They also critically reviewed the paper, approved its final version and agreed with its publication.

– No signature responsible for the work has been omitted and the criteria of Scientific Authorization are satisfied.

– The results of this Scientific Work have been interpreted objectively. Any result contrary to the point of view of those who sign is exposed and discussed in the Scientific Work.

Copyright and Access

The publication of this Scientific Work entails the transfer of the copyright to ECORFAN-Mexico, SC in its Mexico Holding for its ECORFAN Books, which reserves the right to distribute on the Web the published version of the Scientific Work and the making available of the Scientific Work in this format supposes for its Authors the fulfillment of what is established in the Law of Science and Technology of the United States of Mexico, regarding the obligation to allow access to the results of Scientific Research.

Title of the Scientific Work:

Name and surnames of the Contact Author and the Coauthors	Signature
1.	
2.	
3.	
4.	

Principles of Ethics and Declaration of Solution to Editorial Conflicts

Publisher Responsibilities

The Publisher undertakes to guarantee the confidentiality of the evaluation process, it may not disclose to the Arbitrators the identity of the Authors, nor may it reveal the identity of the Arbitrators at any time.

The Editor assumes the responsibility of properly informing the Author of the phase of the editorial process in which the text is sent, as well as the resolutions of Double Blind Arbitration.

The Editor must evaluate the manuscripts and their intellectual content without distinction of race, gender, sexual orientation, religious beliefs, ethnicity, nationality, or the political philosophy of the Authors.

The Editor and his editing team of ECORFAN® Holdings will not disclose any information about the Scientific Work sent to anyone other than the corresponding Author.

The Editor must make fair and impartial decisions and ensure a fair peer arbitration process.

Responsibilities of the Editorial Board

The description of the processes of peer review is made known by the Editorial Board in order that the Authors know the evaluation criteria and will always be willing to justify any controversy in the evaluation process. In case of Detection of Plagiarism to the Scientific Work the Committee notifies the Authors for Violation to the Right of Scientific, Technological and Innovation Authorization.

Responsibilities of the Arbitration Committee

The Arbitrators undertake to notify about any unethical conduct by the Authors and to indicate all the information that may be reason to reject the publication of the Scientific Work. In addition, they must commit to keep confidential information related to the Scientific Work that they evaluate.

Any manuscript received for your arbitration must be treated as confidential, must not be displayed or discussed with other experts, except with the permission of the Editor.

The Referees should conduct themselves objectively, any personal criticism of the Author is inappropriate.

The Arbitrators must express their points of view with clear and valid arguments that contribute to the Scientific, Technological and Innovation of the Author.

The Arbitrators should not evaluate the manuscripts in which they have conflicts of interest and that they have been notified to the Editor before submitting the Scientific Work to evaluation.

Responsibilities of Authors

Authors must ensure that their Scientific Works are the product of their original work and that the data have been obtained in an ethical manner.

Authors must ensure they have not been previously published or are not being considered in another serial publication.

Authors must strictly follow the rules for the publication of Scientific Works defined by the Editorial Board.

Authors should consider that plagiarism in all its forms constitutes unethical editorial conduct and is unacceptable, consequently any manuscript that incurs plagiarism will be removed and not considered for publication.

Authors should cite publications that have been influential in the nature of the Scientific Work submitted to arbitration.

Information services

Indexing - Bases and Repositories

RESEARCH GATE (Germany)
MENDELEY (Bibliographic References Manager)
GOOGLE SCHOLAR (Citation indices-Google)
REDIB Ibero-American Network of Innovation and Scientific Knowledge-CSIC

Publishing Services

Citation and Index Identification H
Management of Originality Format and Authorization
Testing of Books with PLAGSCAN
Evaluation of Scientific Work
Issuance of Certificate of Arbitration
Edition of Scientific Work
Web layout
Indexing and Repository
Publication of Scientific Work
Certificate of Scientific Work
Editing Service Billing

Editorial Policy and Management

143 – 50 Itzopan, Ecatepec de Morelos–Mexico. Phones: +52 1 55 6159 2296, +52 1 55 1260 0355, +52 1 55 6034 9181; Email: contact@ecorfan.org www.ecorfan.org

ECORFAN®

Chief Editor
VARGAS-DELGADO, Oscar. PhD

Executive Director
RAMOS-ESCAMILLA, María. PhD

Editorial Director
PERALTA-CASTRO, Enrique. MsC

Web Designer
ESCAMILLA-BOUCHAN, Imelda. PhD

Web Diagrammer
LUNA-SOTO, Vladimir. PhD

Editorial Assistant
REYES-VILLAO, Angélica. BsC

Translator
DÍAZ-OCAMPO, Javier. BsC

Philologist
RAMOS-ARANCIBIA, Alejandra. BsC

Advertising & Sponsorship

(ECORFAN® -Mexico – Bolivia – Spain – Ecuador – Cameroon – Colombia - El Salvador – Guatemala -Nicaragua-Peru-Paraguay-Democratic Republic of The Congo, Taiwan), sponsorships@ecorfan.org

Site Licences

03-2010-032610094200-01-For printed material ,03-2010-031613323600-01-For Electronic material,03-2010-032610105200-01-For Photographic material,03-2010-032610115700-14-For the facts Compilation,04-2010-031613323600-01-For its Web page,19502-For the Iberoamerican and Caribbean Indexation,20-281 HB9-For its indexation in Latin-American in Social Sciences and Humanities,671-For its indexing in Electronic Scientific Journals Spanish and Latin-America,7045008-For its divulgation and edition in the Ministry of Education and Culture-Spain,25409-For its repository in the Biblioteca Universitaria-Madrid,16258-For its indexing in the Dialnet,20589-For its indexing in the edited Journals in the countries of Iberian-America and the Caribbean, 15048-For the international registration of Congress and Colloquiums. financingprograms@ecorfan.org

Management Offices

143 – 50 Itzopan, Ecatepec de Morelos–México.
21 Santa Lucía, CP-5220. Libertadores -Sucre–Bolivia.
38 Matacerquillas, CP-28411. Moralzarzal –Madrid-España.
18 Marcial Romero, CP-241550. Avenue, Salinas l - Santa Elena-Ecuador.
1047 La Raza Avenue -Santa Ana, Cusco-Peru.
Boulevard de la Liberté, Immeuble Kassap, CP-5963.Akwa- Douala-Cameroon.
Southwest Avenue, San Sebastian – León-Nicaragua.
31 Kinshasa 6593 – Republique Démocratique du Congo.
San Quentin Avenue, R 1-17 Miralvalle - San Salvador-El Salvador.
16 Kilometro, American Highway, House Terra Alta, D7 Mixco Zona 1-Guatemala.
105 Alberdi Rivarola Captain, CP-2060. Luque City- Paraguay.
69 Street. YongHe district, ZhongXin. Taipei-Taiwan.
43 Street # 30 -90 B. El Triunfo CP.50001. Bogota Colombia.

ISBN 978-607-8695-79-9

www.ecorfan.org