KMeleon Guide

Informations sur le document :

Author : Florian DORDAIN
Company : Tesseract Solutions
Address : 6 Avenue Archimède, 02100 Saint-Quentin
Date : 03/01/2020

Version : 1.0

Table of contents

Table of illustrations

Document identification

DOCUMENT IDENTIFICATION	
Name of the project	KMeleon
Number	KM-001
Version	1.0
Date	14/11/20

	Name	Signature	Date
Project manager	Florian DORDAIN	Florian DORDAIN	14/11/20
Author	Florian DORDAIN	Florian DORDAIN	14/11/20

Revisions list

Revision	Description	Modified by	Date
1	Validation of countermeasures	Florian DORDAIN	14/11/20

Foreword

Description of the product

The software Kmeleon allows to create programs for several brands of robots.

Programming is graphic and without any line of code. Programming is made with block functions moved by the drag and drop system.

What does this guide aim to ?

This guide aims to explain how to create a first project on KMeleon. This document is made for operators, technicians, engineers and every user who would be likely to work with this software.

What are the prerequisites to install KMeleon ?

Basic technical notions must be taken into account:

- To know what is a sensor
- To know what is a actuator
- To know what is a teach pendant
- To have already manipulated or worked with a robot arm

What are the prerequisites on Fanuc robots ?

- User socket messaging Option
- Modbus TCP Option

What are the prerequisites on Universal Robots robots ?

- Have the 30002 port free

What are the prerequisites on Staübli robots ?

- Modbus Option

Safety rules

Regulations and prevention rules against accidents are linked with the safety, occupational medicine, and environment protection must be respected at any time.

During the conception of the software, everything was made to make your work safer. Care is irreplaceable, there is no best rule to avoid accidents.

Warning :

- This software and its box do not have safety inputs and outputs.

Prohibitions :

- So, it is strictly prohibited to configure safety functions through KMeleon. If so, the editor will be ejected of any responsibility.

Risks in case of bad use :

- Mechanical impact
- Electrical discharge
- Assets deterioration

User qualification

This software must be used exclusively by entitled people (nominaly designated and qualified), previously educated on the software management and authorized to use it.

The only user who can use the software must be trained on it. This user should have received basic training by the company TESSERACT SOLUTIONS.

Before any use of the software, you have to be familiarized with every commands and its correct use. This guide is here to remind safety points and instructions to use KMeleon.

Warning :

The entitled user must :

- Have read this guide
- Have followed a training

Manufacturer and user responsibilities

You must respect all of the installation, functioning, adjustment maintenance and repairs advices included in this guide.

You must exclusively use spare parts and accessories which are in keeping with the manufacturer's recommendations.

Do not change by yourself or create modifications by another person without asking a manufacturer's written agreement.

The non-respect of these rules can make your software dangerous. In case of degradations or injuries, the manufacturer will be ejected from any responsibility.

Warnings and pictograms

Warnings and pictograms which are mentioned in the software provide indications about safety measures that you have to strictly follow to avoid any accident.

Rearmament of the software

The rearmament can be done through the reamrded button on the software.

This button allows the machine rearmament remotely.

	Warning : • The machine rearmament is done after that you have done the emergency stop of the robot. • Before rearming, the operator must have known his environment and guarantee that the cycle will begin again safely for goods and people. • In case of bad use, the editor will be ejected from any responsibility.

Interface of the software

Interface of the software :

When you open the software, you find this interface below:

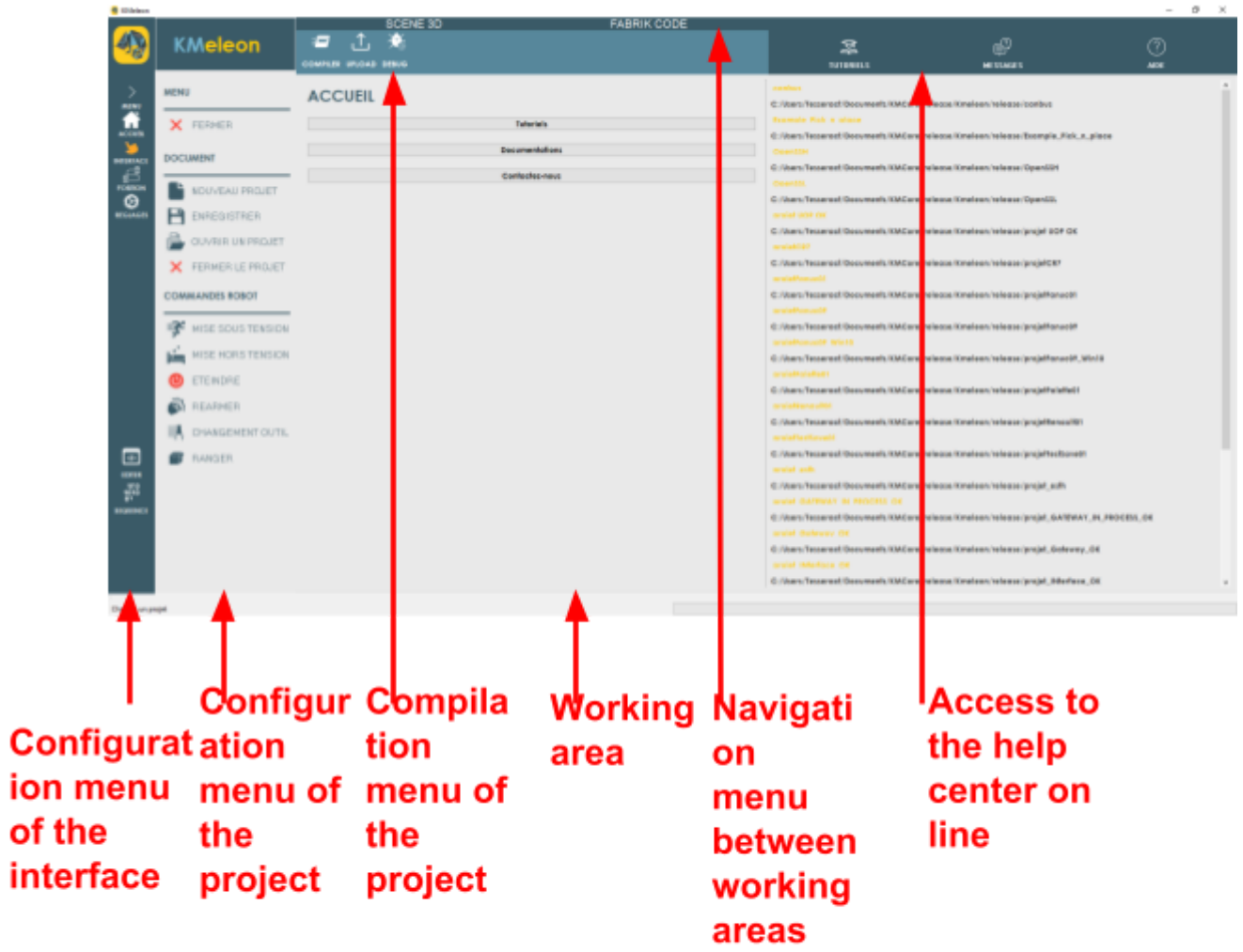


Image 1 : Interface of KMeleon software from the start screen

Configuration menu of the project

This menu allows you to manage your project (new project, save the project, open and close a project) and to manage the robot which fits with the project (tensioning and switch off of the axis, switching off the robot's bay, rearmed the robot, change its position).

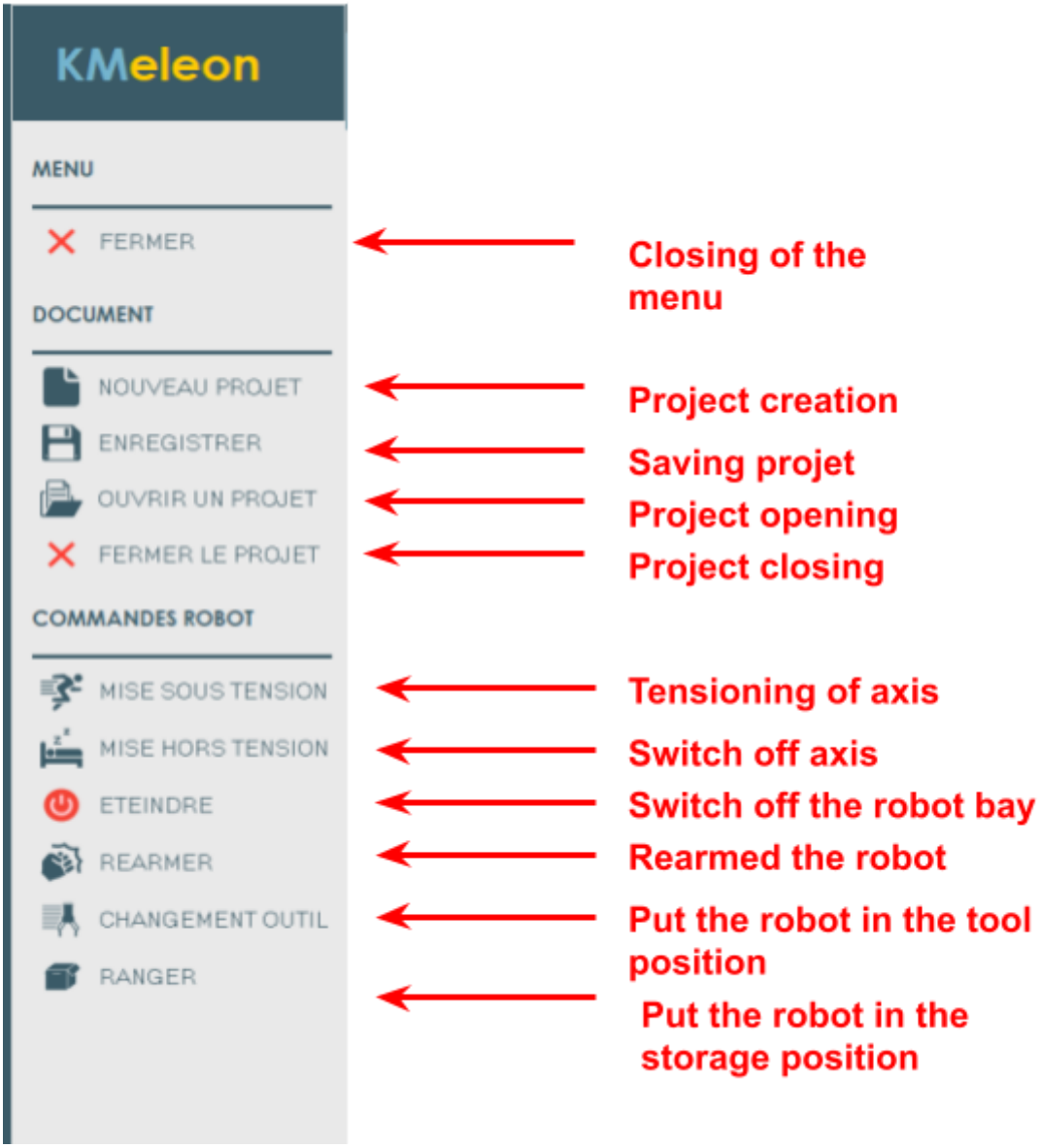


Image 2 : Configuration menu of the left of the software

Installation of robot drivers

1. Via USB stick

a. Fanuc robot

Click on the "MENU", then "FILE". The window illustrated below will open.

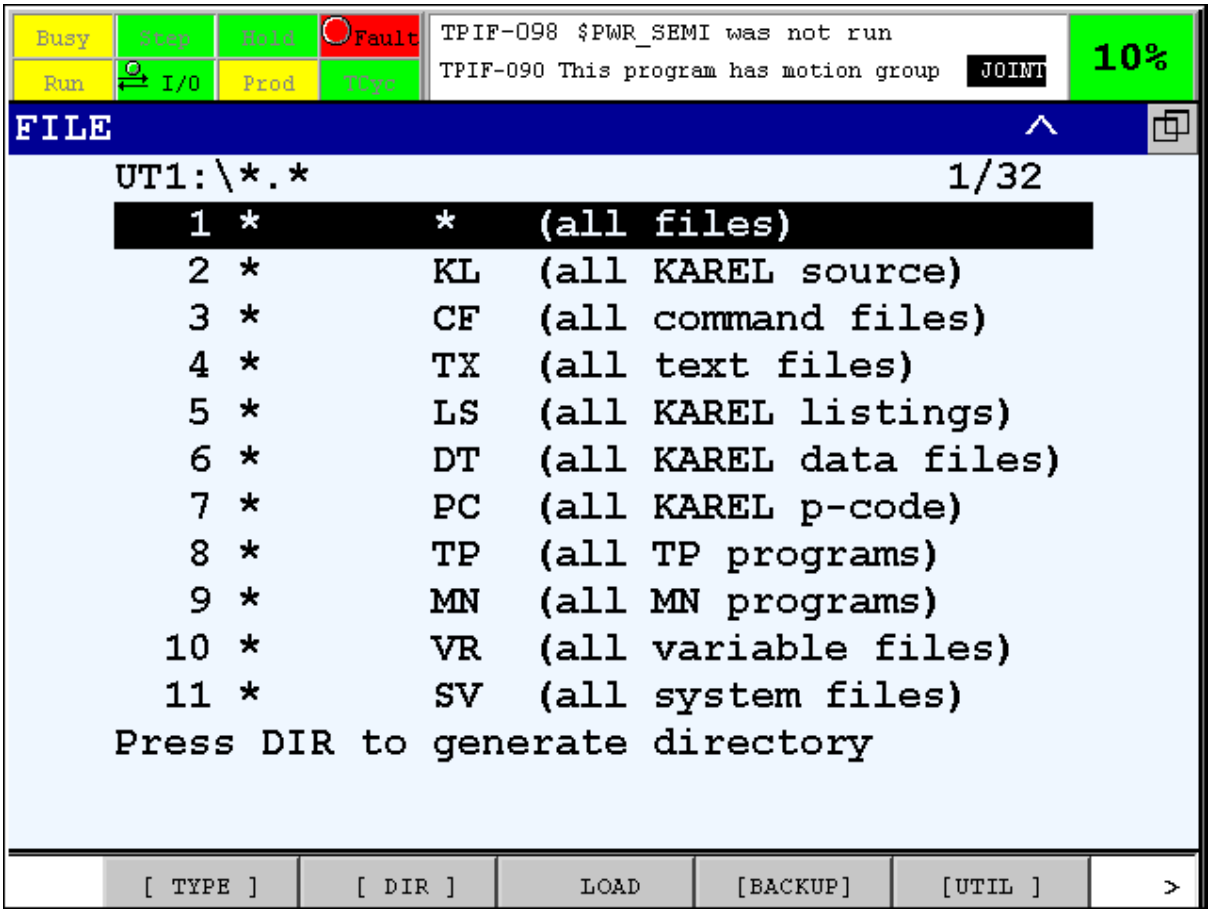
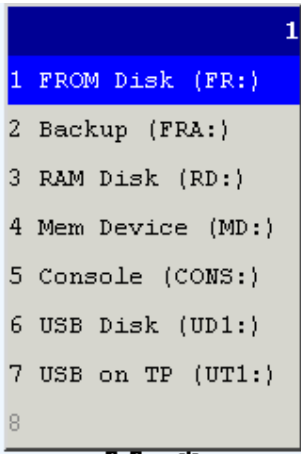


Image 3 : FILE menu

Then, click on "UTIL", after click on "SET DEVICE".



- If your USB stick is connected on the USB port of the bay, click on "USB Disk".
- If your USB stick is connected to the *teach pendant*, click on "USB on TP".

Image 4 : FILE FANUC Menu

The machine is now selected. Click on "PC (all KAREL p-code)", programmes illustrated below will open:

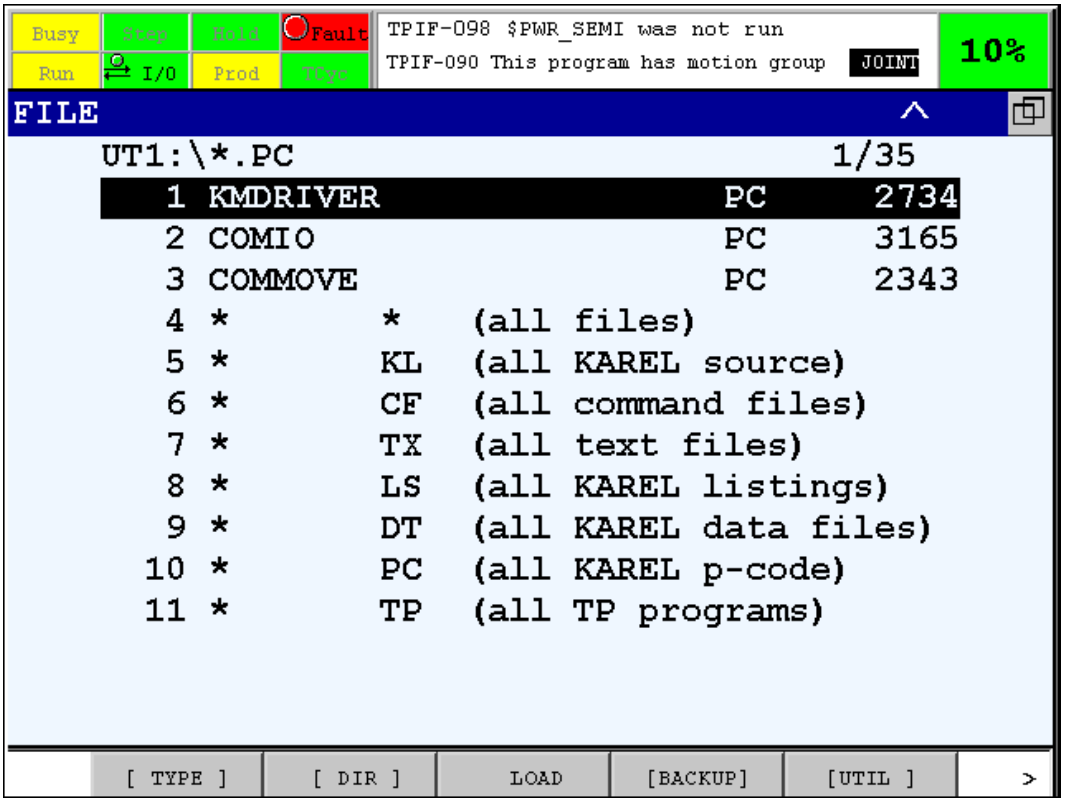


Image 5 : FILE FANUC Menu - list of programs

Chose “COMIO” and “COMMOVE”, then, click on “LOAD”. Your drivers are now loaded.

Configure the Ethernet connexion

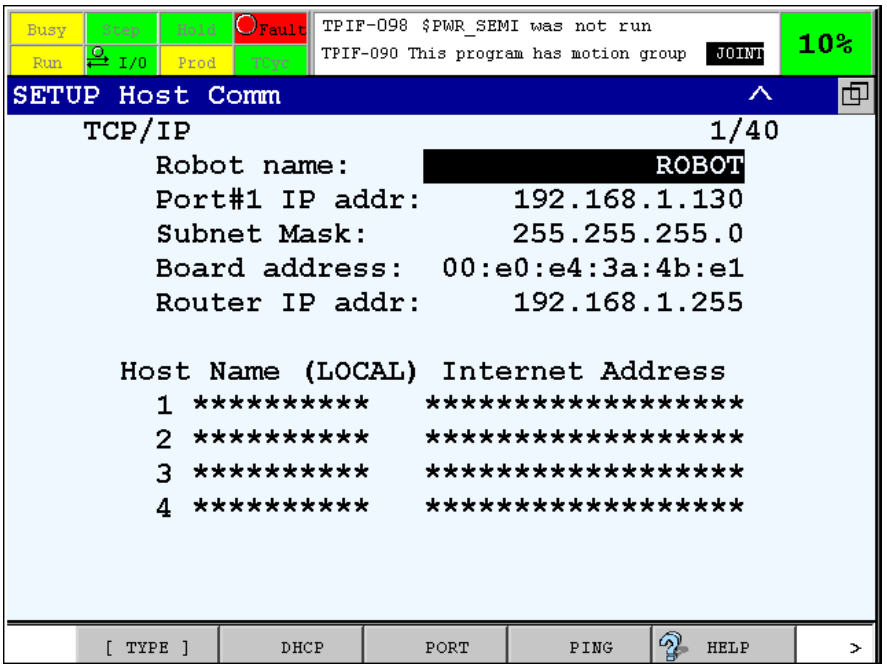
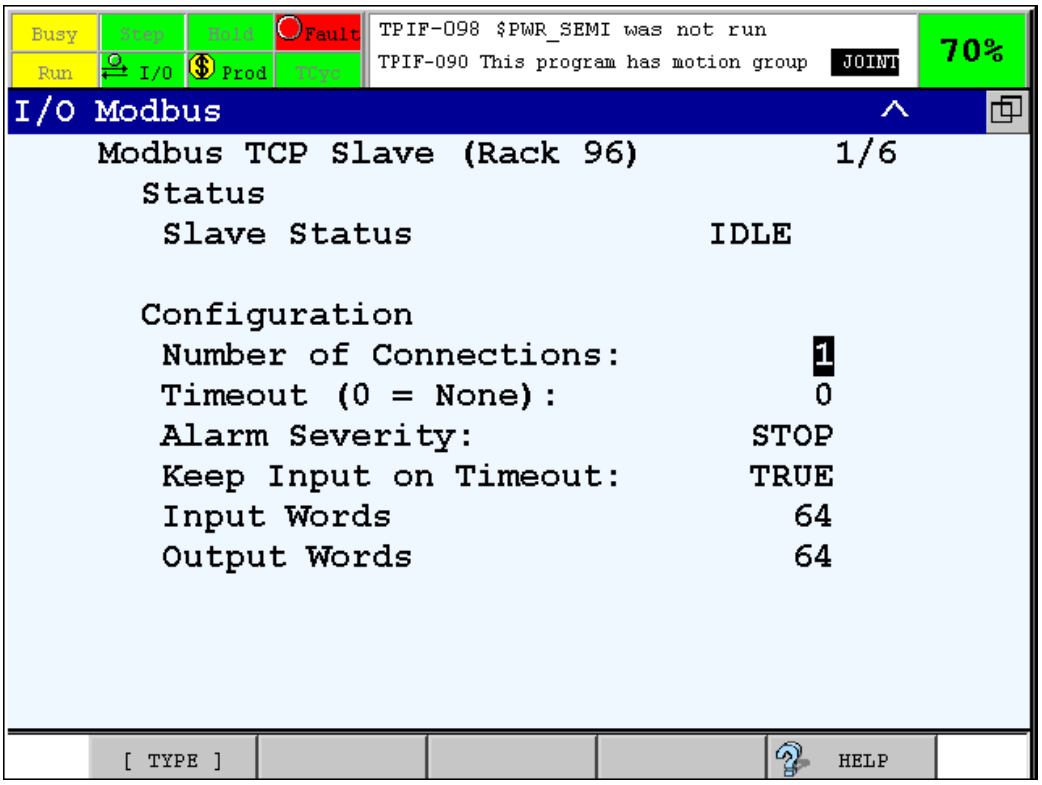


Image 6 : HOST FANUC Menu

Configure the Modbus connexion

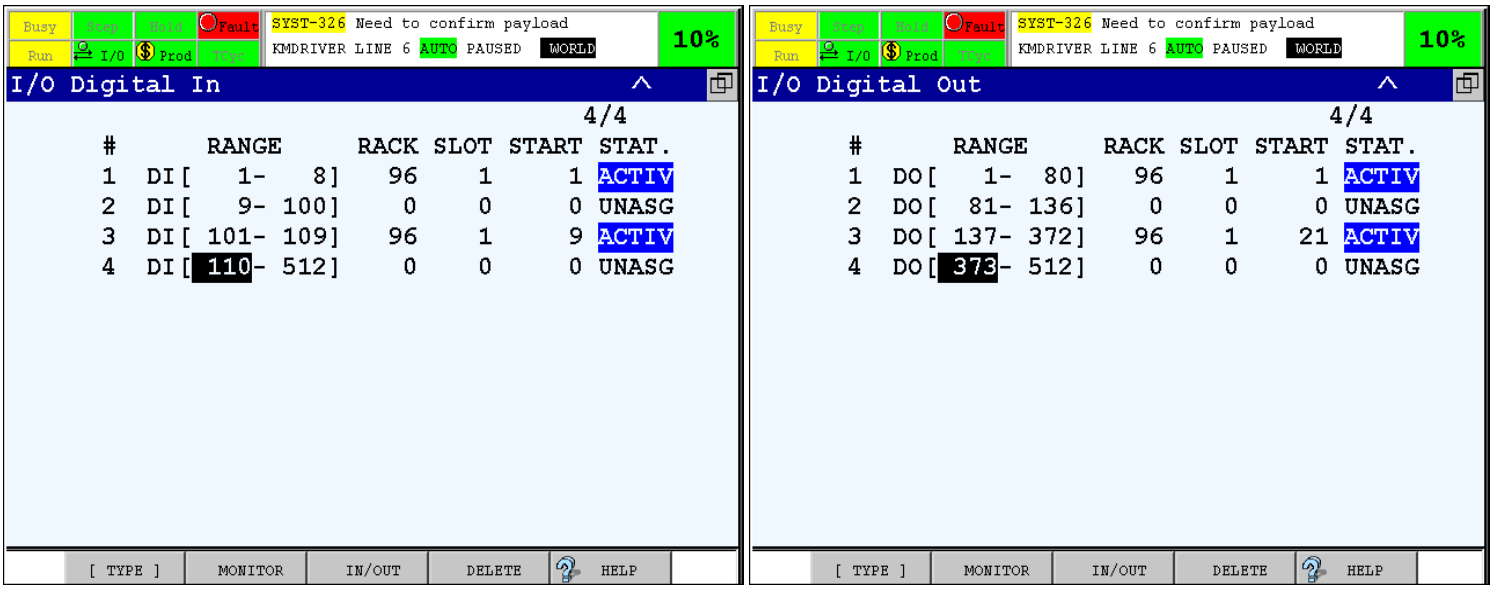
You can follow the configuration below :

The requirement Timeout must imperatively be on 0.



Configure the inputs and outputs

- On the *teach pendant* of the robot, go in the "MENU", then "5 I/O" and "3 Digital".
- Click on "F2 CONFIG" to go in the setup.
- Click on F3 to switch from inputs setup to outputs setup.



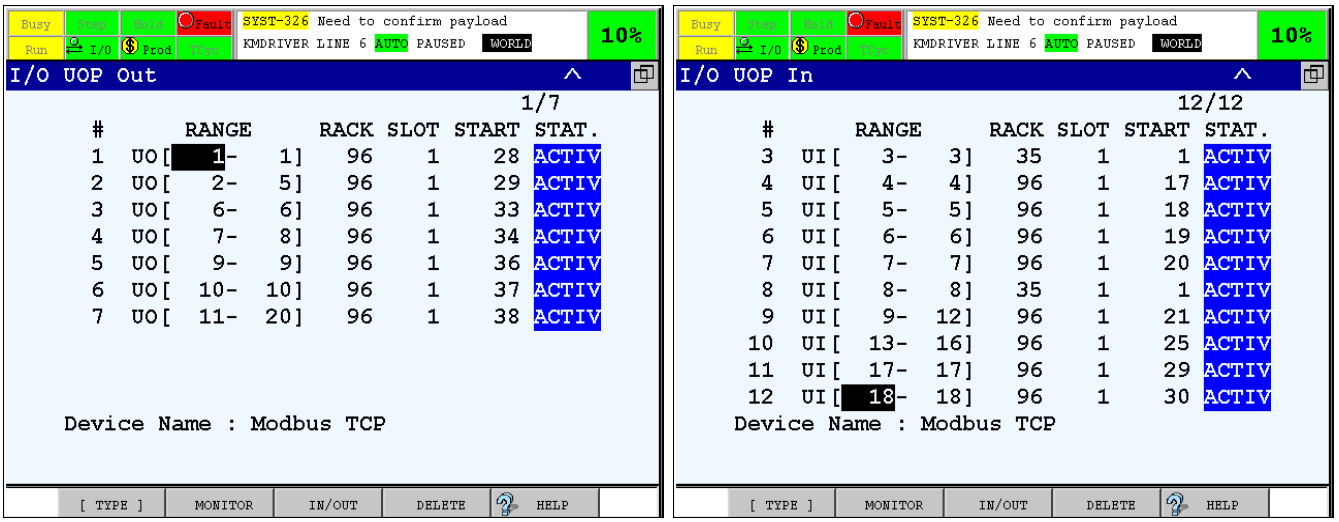
- Set up the inputs - outputs that you want in the Modbus port 96 - 1 - X).

For a standard configuration, follow the configuration below:

Image 7 : Configuration screen of inputs and outputs

Set up the UOP

- On the *teach pendant* of the robot, go into"MENU", then "5 I/O" and "7 UOP".
- Then, click on the button "F2 CONFIG" to go in the setup.
- Click on "F3" to switch from the inputs setup to the outputs setup



- Configure the inputs - outputs on the Modbus port (96 - 1 - X).

Image 8 : Set up screen of UOP.

For a standard configuration, follow the configuration below :

Set up into the remote mode

In the SETUP menu :

- On the *teach pendant* of the robot, go into "MENU", then "6 SETUP" and "1 Prog Select".

- In "Program select mode", choose the method OTHER from the button "F4 CHOICE".
- In "Production Start Method" choose "UOP" from the button "F4 CHOICE".

In the SYSTEM menu :

- On the *teach pendant* of the robot, go into "MENU", then → "0 NEXT", "6 SYSTEM" and "6 Config".
- Change the variable "7 Enable UI Signals" vers "true".
- Change the variable "43 Remote /Local Setup" toward "Remote" from the button "F4 CHOICE".

Start automatically the driver from KMeleon

In the KMeleon menu, click on the icon "interface". This action opens the user interface.

Click on the button "START" : the program will begin, it can take few seconds.

To stop the cycle, click on the button "STOP".

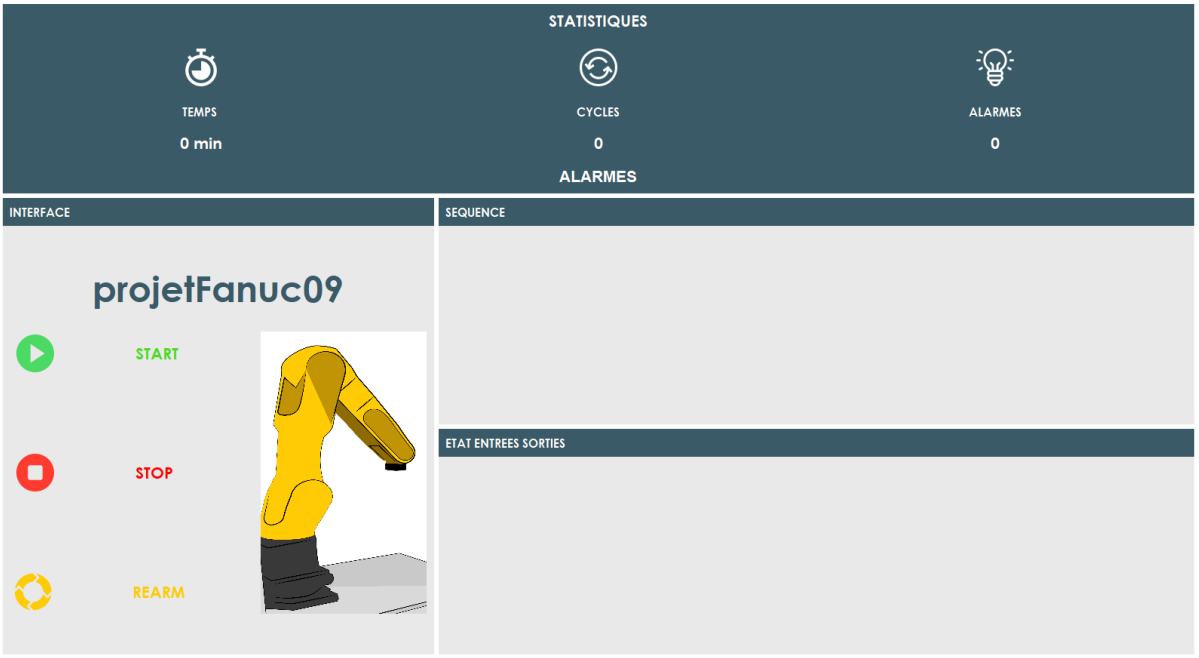


Image 9 : KMeleon user interface

Start the driver manually

In the standard mode, the *driver* is started by UOP.

If you want to start it manually, the *driver* :

In the SYSTEM menu:

- On the teach pendant of the robot, go into "MENU", then "0 NEXT", "6 SYSTEM" and "6 Config".
- Change the variable "7 Enable UI Signals" toward "false"
- Change the variable "43 Remote /Local Setup" toward "Local" from the button "F4 CHOICE"

On the teach pendant click on "SELECT", then "KMDRIVER".

After, turn the switch of the teach for the position "OFF" and turn the switch of the controller for the position AUTO".

Click on the green button "Cycle Start" to start the *driver*.

Interface

1. Simulation

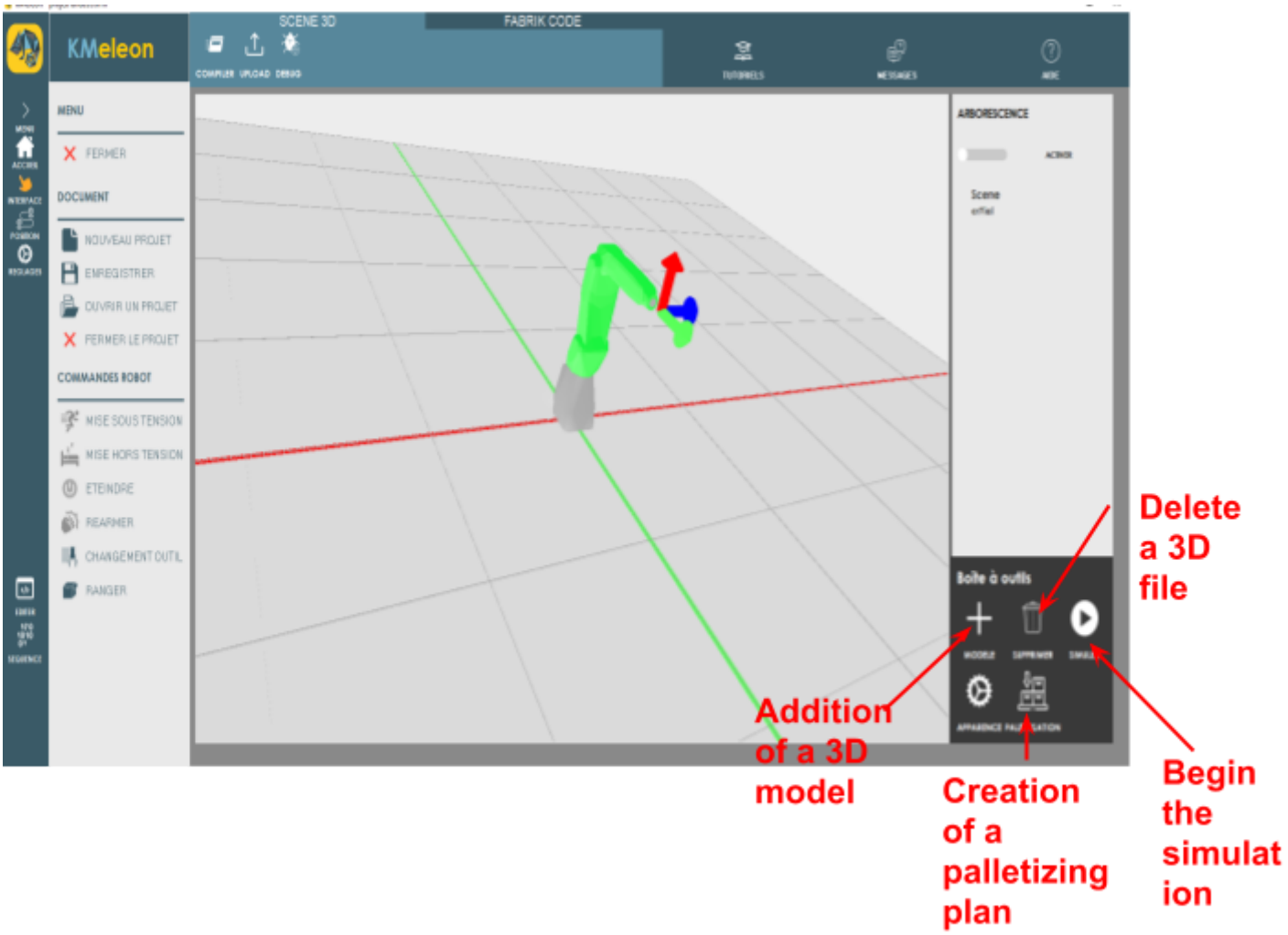


Image 10 : 3D scene of KMeleon

The button "MODÈLE" allows you to add a 3D model and then, select the robot arm that you want.

To move the model in the scene, indicate a translation step via the selector, then choose an axis of motion.

To change the model orientation, from the selector, indicate a rotation step, then indicate an axis of rotation.

*__Translation :__ *In geometry, a translation is a geometric transformation which corresponds to the intuitive idea of « sliding » of a thing, without rotation, reversal or distortion of this thing. (Source : Wikipédia).*

To load a robot

In the tool box the simulation interface, click on the button"modèle".

A window opens and proposes you models from the chosen brand.

Select the model that you want and click on "OK" at the bottom of the window.

Load a STL file

In the tool box of the simulation interface, click on the button "modèle".

A window opens, at the bottom of the window a button "Fichier STL" appears, click below.

A new window opens, choose your file and click on "OK".

The file is now loaded in the simulation window.

Views management

Right-click on the interface : a contextual menu appears. Select "vues", then choose the view that you want.

Movement of the pieces

In the simulation interface, in the tree view of pieces, click on the piece you want to move.

A contextual menu appears, select the action you want to do. You can move the pieces with less precision thanks to the mouse click or move them precisely thanks to the function “coordonnées”.

2. FabrikCode

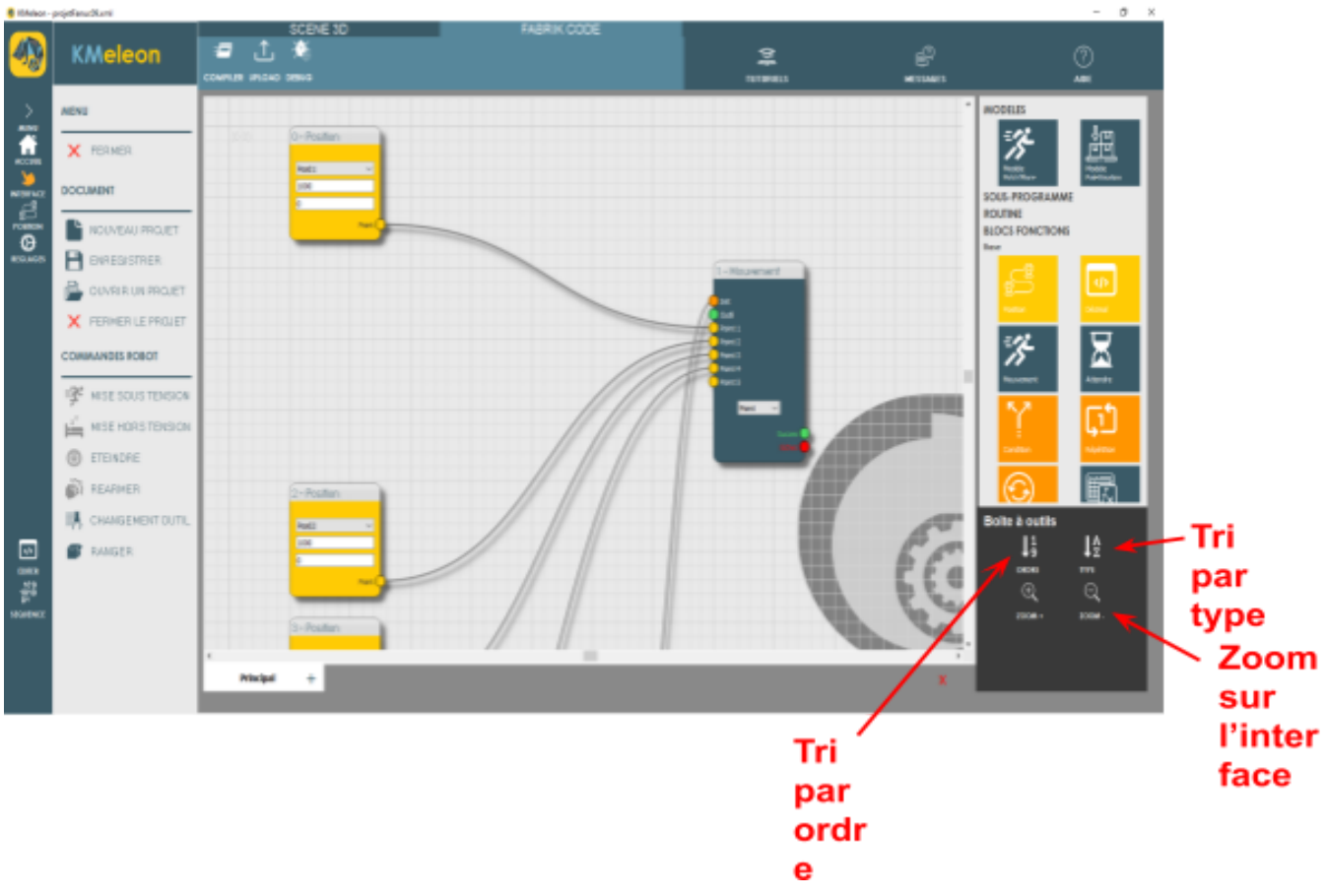


Image 11 : FabrikCode Interface

To create a block, select the block that you want in the list "BLOCS FONCTIONS" on the right of the software. Maintain the click on the block that you want and release the click in the scene to make it appear.

You can now link the blocks and configure them.

Creation of a project

1. Configuration of a project

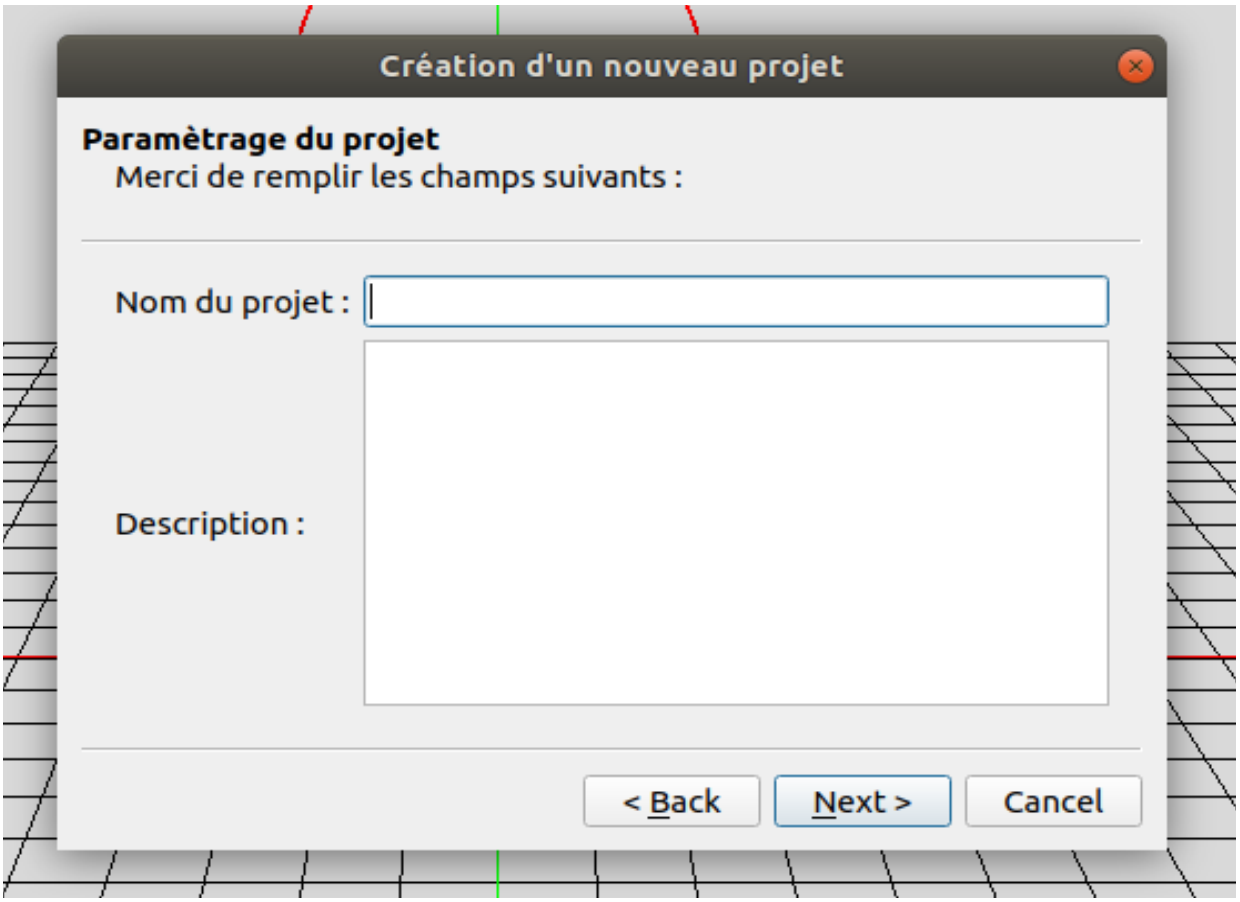


Image 12 : First part of the configuration window of the robot

You have to give a name to your project.

You can give a description to your project to make it more understandable for the others.

2. Configuration of the robot

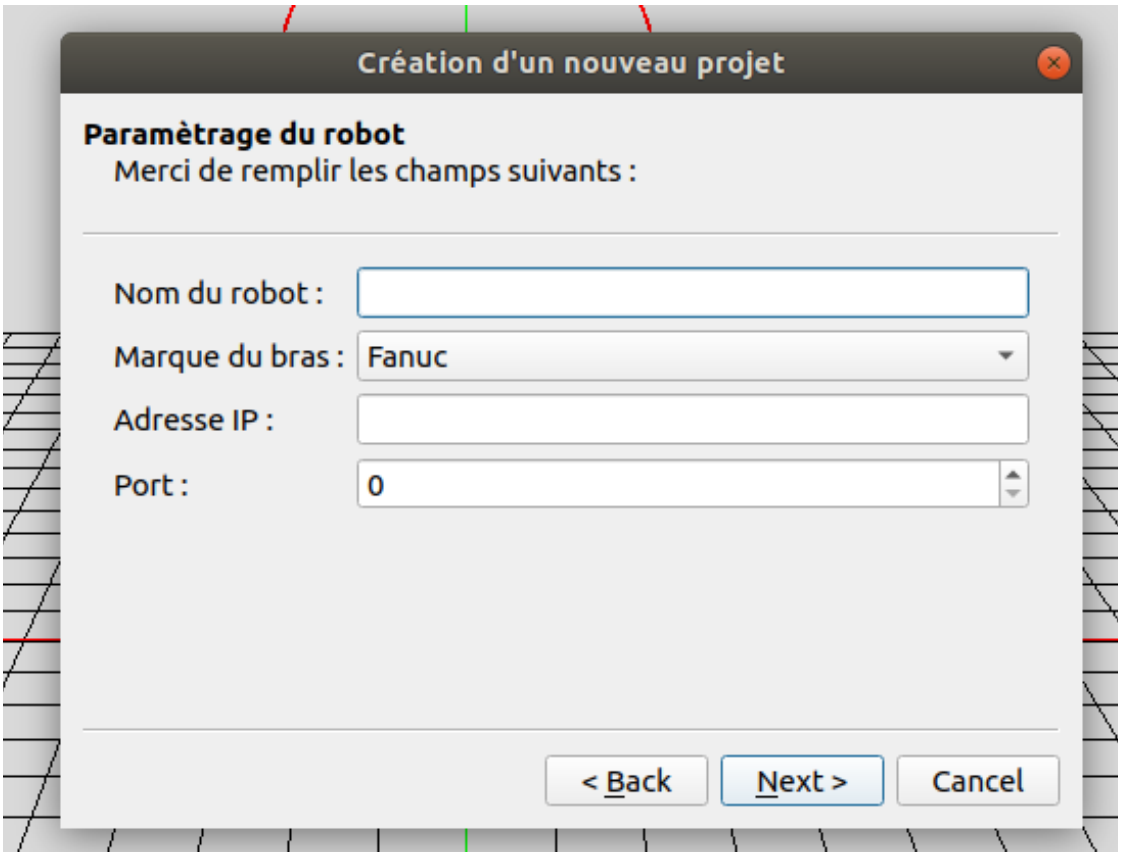


Image 13 : Second part of the configuration window of the robot

This window asks you to configure the robot arm.

- "Nom du robot" corresponds to the name of the bay. This name is not required, you can name it as you want.
- "Marque du bras" allows you to scroll in a selection menu of the robot arm in the chosen brand.
- "Adresse IP" corresponds to the identification address of the robot arm on the network.

- “Port” allows you to identify the service with which the robot communicates.

TIPS : Find the IP address of a Fanuc robot

Go into : "MENU", then "SETUP" and "HOST COM"

The window below appears :

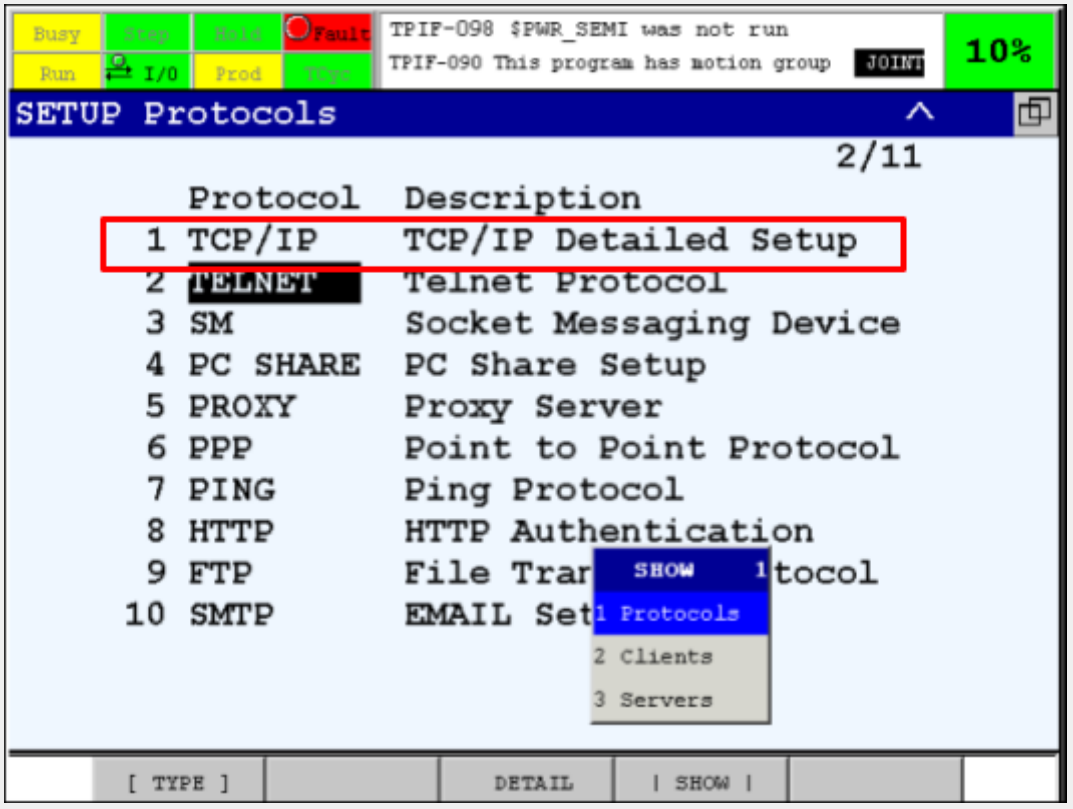


Image 14 : Window HOST FANUC : TCP/IP highlighted

Click on TCP/IP, the next tab opens :

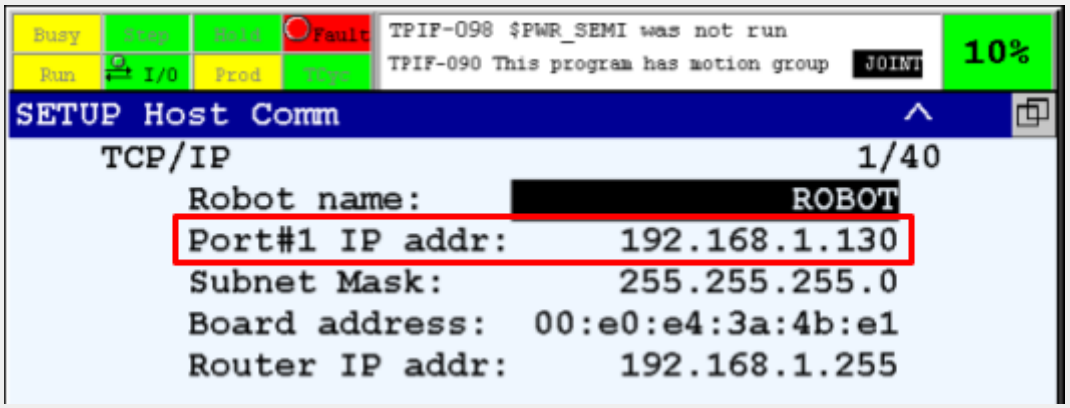


Image 15 : Tab TCP/IP of the menu HOST FANUC : IP address highlighted

"Port#1 IP addr:" corresponds to the IP address of the robot.

Tips : Find the IP address of a UR robot

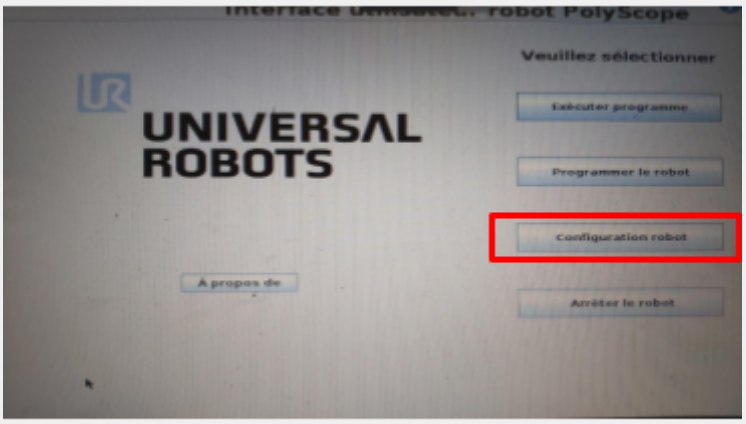


Image 16 : UR Menu - robot setup highlighted

Click on "configuration robot", the following window opens :

Image 17 : Setup menu of an UR robot - network highlighted

Click on "Réseau", the field "Adresse IP" corresponds to the IP address of the robot, as the highlighted field below:

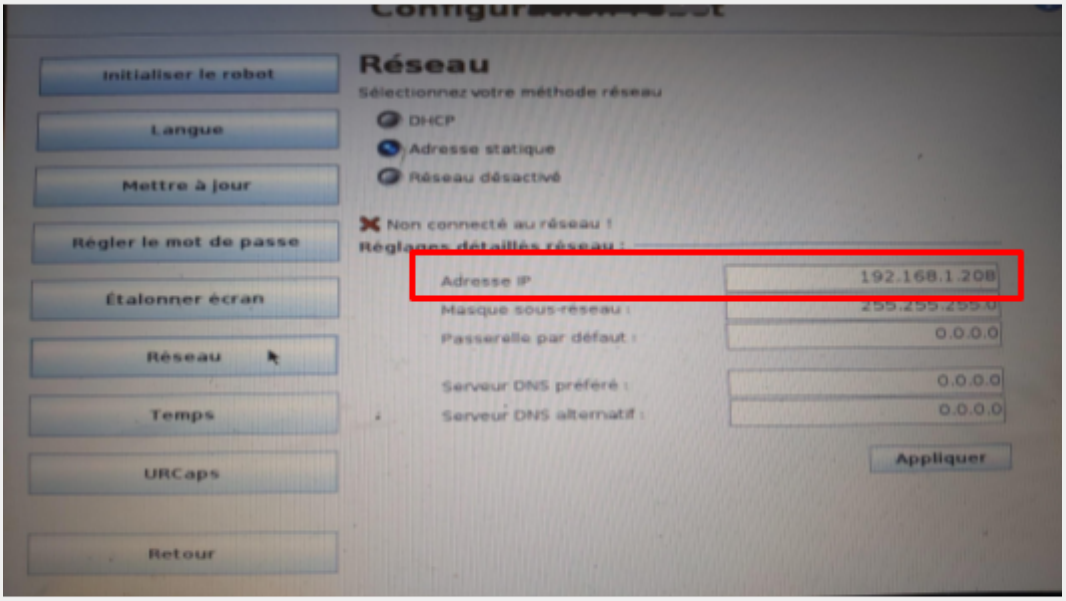


Image 18 : UR network menu - IP - address highlighted

Configure peripheral systems

To access the configuration, click on the button "RÉGLAGES" in the menu on the top of the interface.

Image 19 : Main interface menu on the left

The configuration window opens on the interface below :

Image 20 : Fenêtre de paramétrage

The red highlighted parts represent the functions you have to configure.

Tool configuration

The red part defines the compulsory settings. Below, there is a plan showing you the configuration of the translation and the orientation of the product.

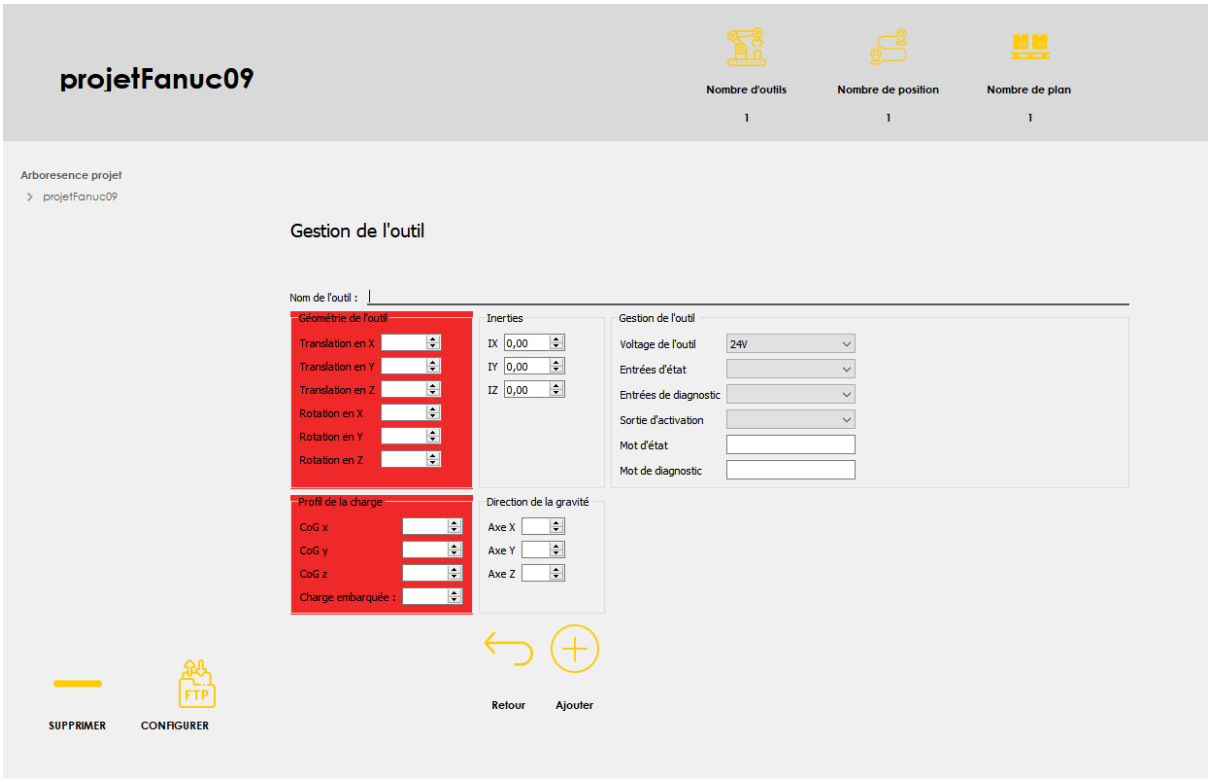


Image 21 : Tool management window

TIPS : Tool configuration

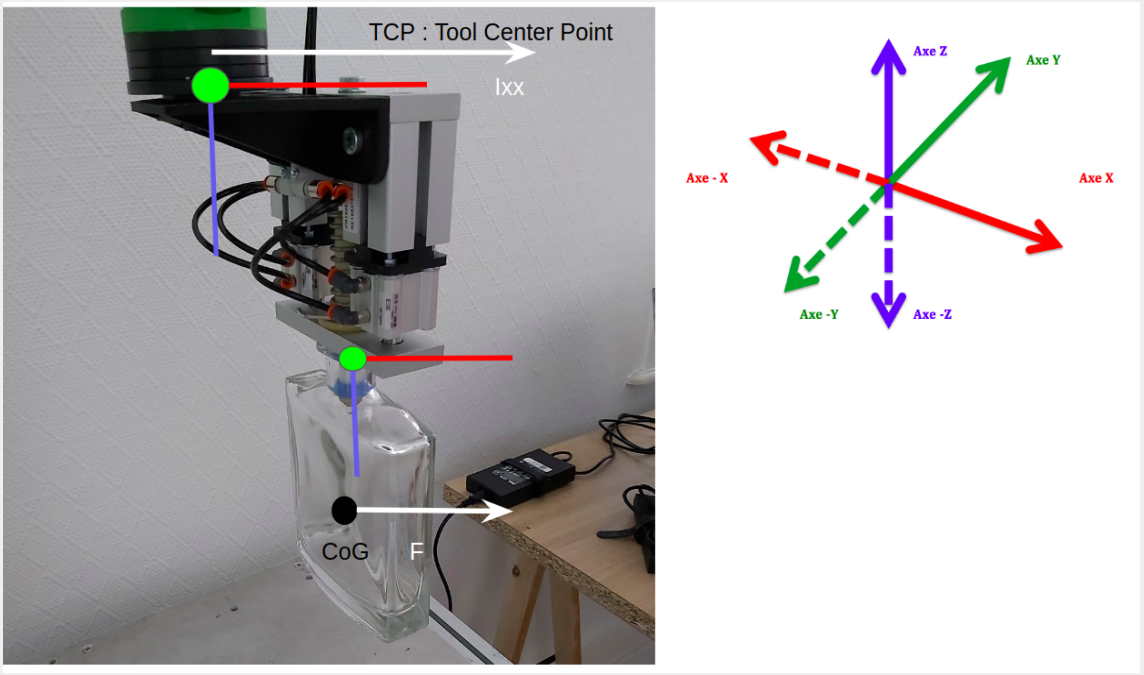


Image 22 : Example of the axis position

The forward-backward axis is named axis X and is represented in red
The right-left axis is named Y axis and is represented in green
The top-bottom axis est named axis Z and is represented in blue

The translation in **X** represents in millimeters the move of your tool forward (ex : +100mm) or backward (ex : -100 mm).
The translation in **Y** represents in millimeters the move of your tool on the right (ex : +100mm) or on the left (ex : -100 mm).
The translation in **Z** represents in millimeters the move of your tool upward (ex : +100mm) or downward (ex : -100 mm).

The rotation represents the inclination of the tool around the axis X, Y ou Z you defined.

The gravity center is the crossroad between the different strengths that allow a thing to balance.

The payload is the mass of the object you want to grab.

1. Configuration of the bay

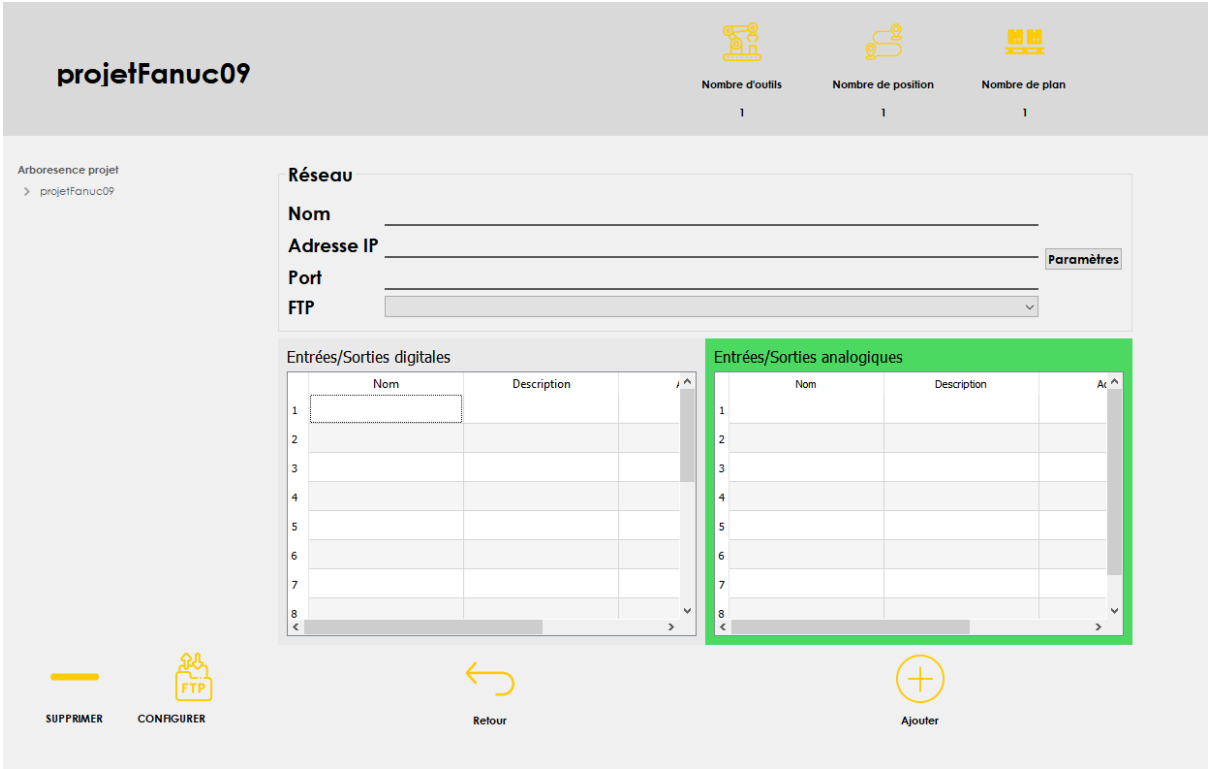


Image 23 : Inputs and outputs configuration window of the robot bay

This window allows you to define the inputs and outputs of the robot bay.

To configure an input, fill in :

- a name
- a description (purpose of this input ?) etc.

- an address (on which terminal is connected the button or the sensor ?).

The sub-window "Entrées/sorties digitales" can be compared to the functioning of switch, on or off. Nevertheless, it does not make precise measures (ex : temperature, humidity, *etc*.).

The sub-window "Entrées/sorties analogique" is the opposite of digital inputs and outputs. It makes a measure or gives a precise information (ex : temperature, humidity, *etc*.). They can be compared to your oven turning knobs.

2. Variables

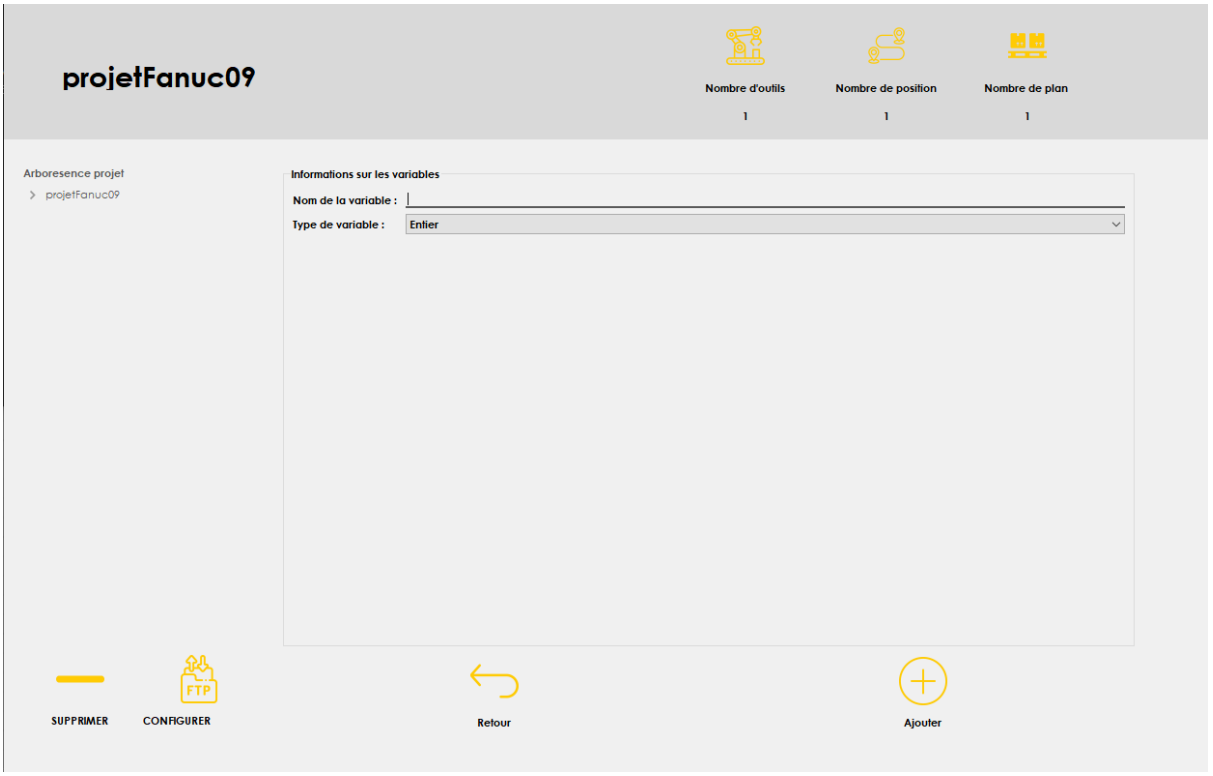


Image 24 : Information window about the variables

The variables* are linked to the software except for the system variables. KMeleon will ask you to fill in:

- Variable's name
- Variable's type, it means:
  - Whole : for a whole number
  - Decimal : for a decimal number
  - Chain : a character string, text
  - Boolean : To create a switch "true" ou "false"
  - System : This type is dedicated to the brand Fanuc. Here, you can report the system variables that you want to configure in your program.

*Variable : *In computing, **variables** are symbols which associate a name (username) to a value (Source : Wikipédia).*

3. Words

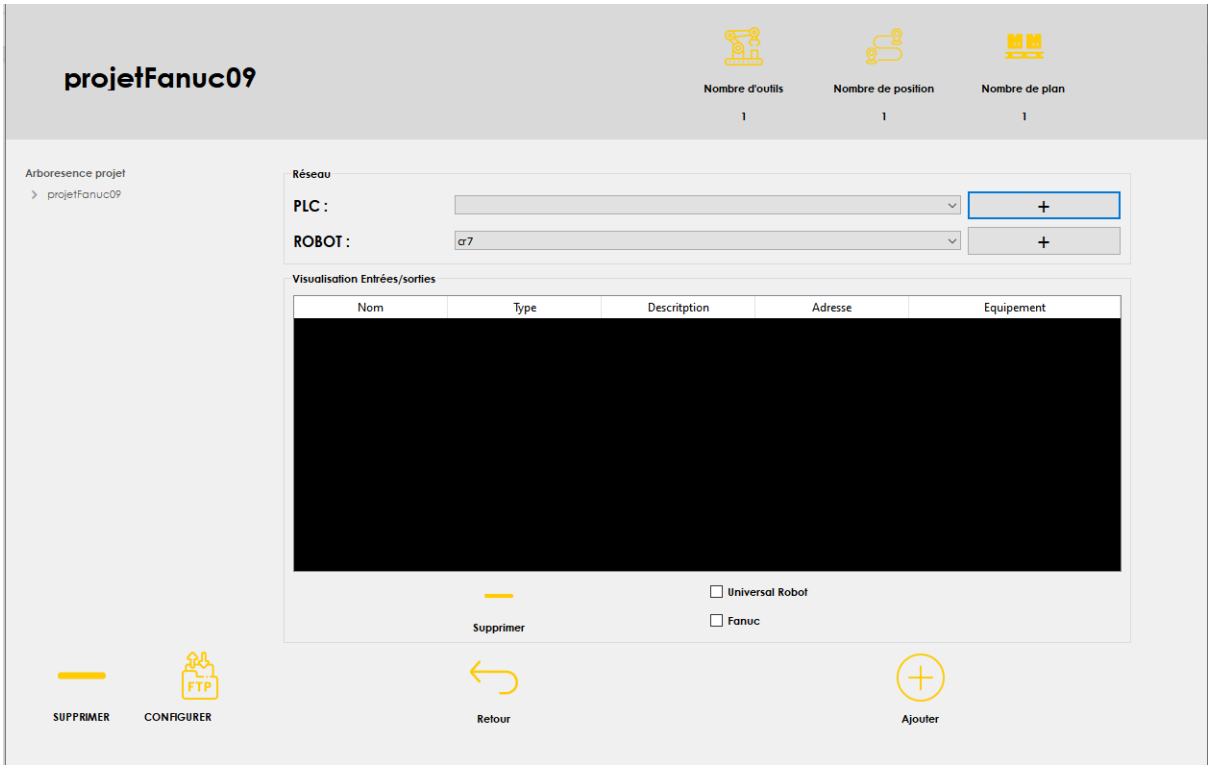


Image 25 : Window on the network

Words* are linked to a robot or an automaton.

The software will ask you to fill in :

- A name.
- A type : the type is defined by default according to a word linked to the automaton or a word linked to the robot.
- A description : you have to describe the data referred to the word.
- An address : corresponds to the address of the word in the automaton or the robot.

- The equipment (defined by default) : it is the equipment on which the word will be linked.

*Word : *In automatism, **words** are symbols which associate a name (username) to a value. They are automaton variables. (Source : Wikipédia).*

4. Conditions

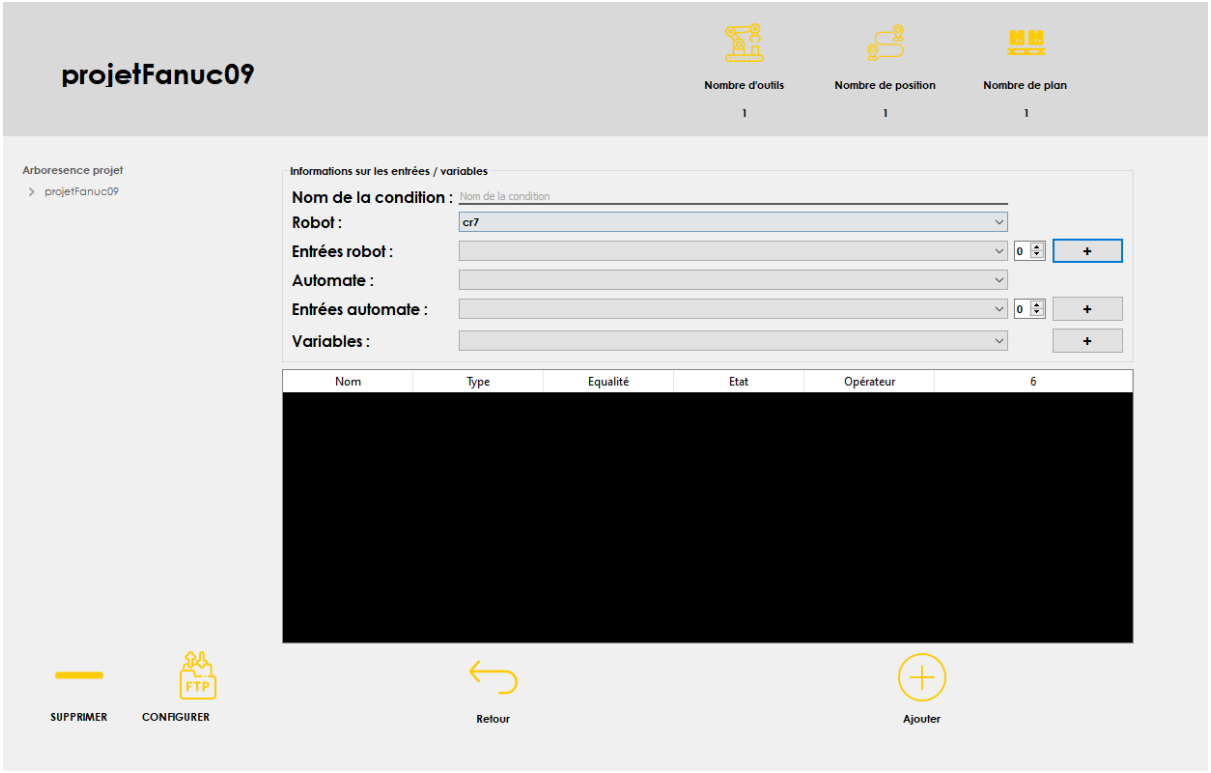


Image 26 : Information window about inputs and outputs variables

Conditions* are linked to the software. For the setup, you have to fill in :

- Condition's name.
- The type is linked to the checked input type:
  - Robot input linked to the robot bay
  - Automaton input linked to an automaton
- Equality :

    - "equal" : allows you to check an equality
    - "different" : allows you to check an inequality
    - "inferior"
    - "superior"
- State:
    - Activated
    - Deactivated
    - If it is a variable, then a value is configured.
- The operator allows you to define another check:
    - "ET" Both checks have to be valid to accept the condition doivent
    - "OU" at least one of the two checks has to be valid to accept the condition.

*Condition : *In computing, **conditions** are a set of equality you have to validate to execute an action (Source : Wikipédia).*

Creation of a program

1. Take positions

⚠	Warning : • The screen must be fixed out of the evolution area of the robot arm. • During the configuration of position or the beginning of the cycle, the operator must be out of the evolution area of the robot to guarantee its safety.

In the KMeleon menu, click on the button "position".

If the real robot is connected, then place your cursor on "réel", if not, keep it on "simulation".

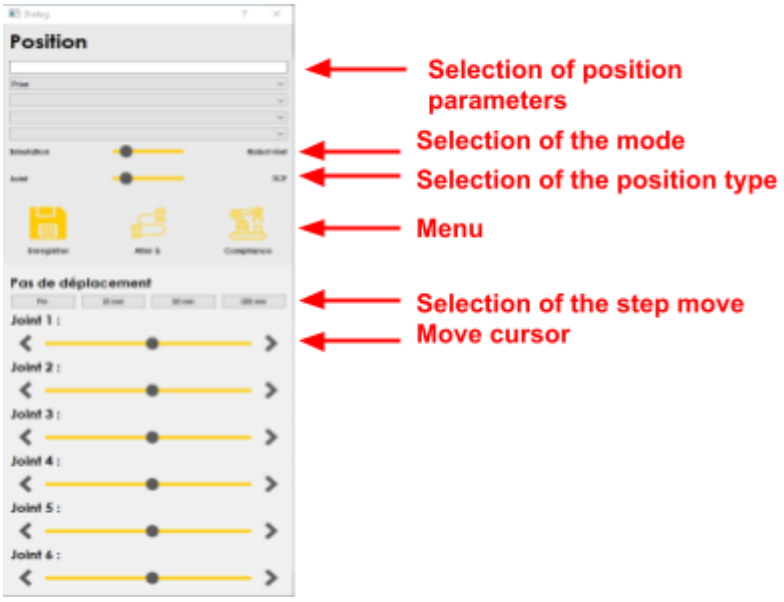


Image 27 : Interface des positions

The robot can me moved by hand :

Select the angular or cartesian position type from the joined cursor / TCP.

If the robot can be moved manually, press "compliance" to free the robot axis, move the robot and press again "compliance".

The position is automatically updated.

Give a name to the position, select the tool, the associated roboto, the type of point (grip, passing, removal) and the landmark.

The robot must be moved from the *teach pendant* :

Select the angular or cartesian position type from the joined cursor / TCP.

Select the step move.

Move the cursors to reach the position that you want. Then, click on the arrow to update the position of the real robot.

The position is automatically updated.

Give a name to the position, select the tool, the associated robot, the type of point (grip, passing, removal) and the landmark.

Move into the simulation mode :

Select the angular or cartesian position type via the joined curseur / TCP.

Select the step move.

Place the cursors to reach the position that you want. Then, click on the arrow to update the robot position.

The position is automatically updated.

Give a name to the position, select the tool, the associated robot, the type of point (grip, passing, removal) and the landmark.

First, go into the FabrikCode interface by clicking on the tab “FABRIK CODE”.

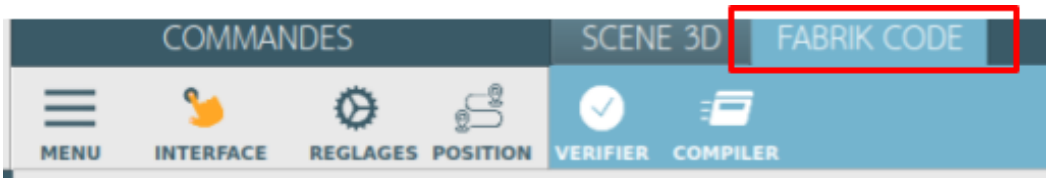


Image 28 : Top of the interface

2. Main program

When you are creating a project, the main program automatically opens.

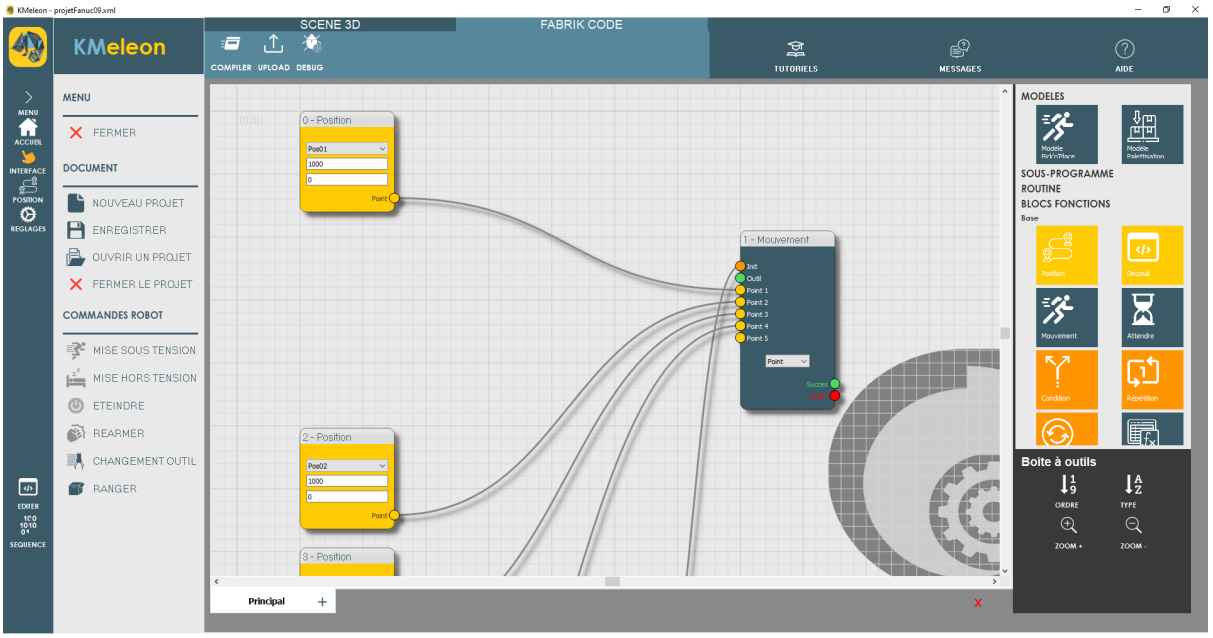


Image 29 : FabrikCode interface

You can then complete it adding blocks.

To create a block, select the block that you want in the list “BLOCS FONCTIONS” on the right. Keep the click on the block that you want and release it in the scene to make the block appear.

You can now link the block and configure it.

3. Sub-program

A sub-program allows you to create a program which will be called at the right time in your main program. This will make the lecture of your main program easier and allows you to use this sub-programs again in different places in your program.

To create a sub-program, click on "+" next to the main tab which is under the FabrikCode scene. Then, click on "nouveau sous-programme".

A window opens and asks you to fill in the sub-program's name.

Some words are prohibited for the sub-program's name:

- Variables
- Succès
- Échec

4. Routine

A routine allows you to execute a program as a background task of your main program. A routine can, for example, execute a temperature measure when the robot is moving.

To create a routine, click on "+" next to the main tab, which is under theFabrikCode scene. Then, click on "nouvelle routine".

A window opens and asks you to fill in the routine's name.

Some words are prohibited for the routine's name:

- Variables
- Succès
- Échec

A new configuration interface opens. The configuration is the same as the main program.

Simulation of a program

When the program is created FabrikCode interface, save it and go back to the simulation interface.

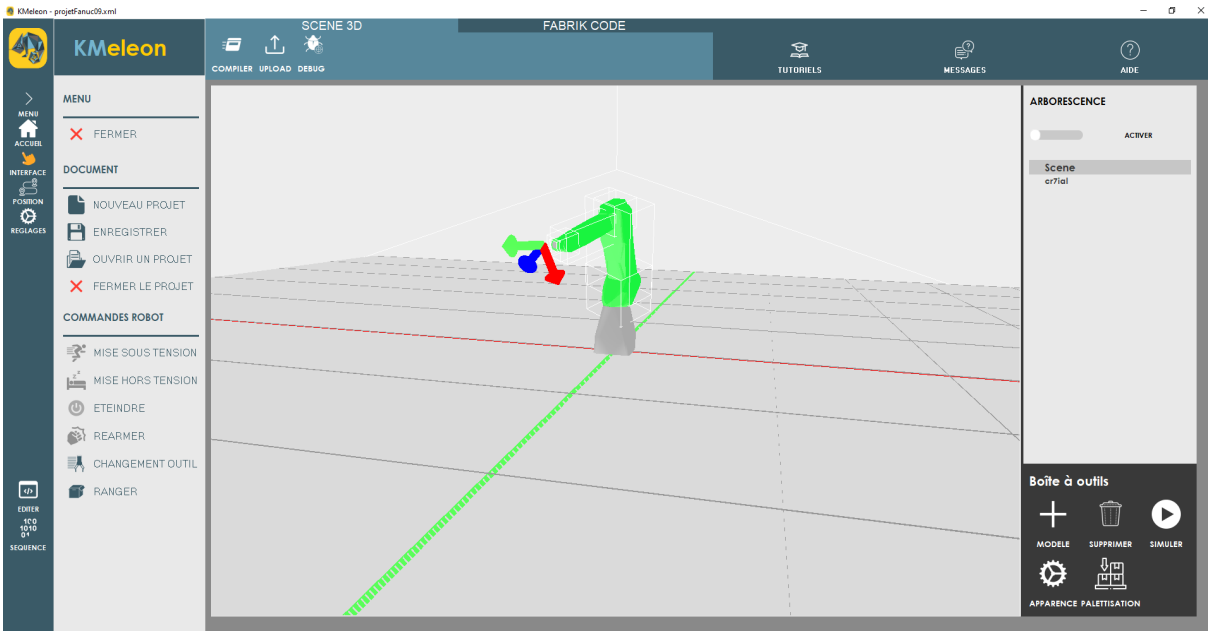


Image 30 : 3D scene interface

Check that a robot is well loaded in the interface.

The, click on “SIMULER” in the tool box simulation interface. The robot will execute the movement you asked for.

Compilation of a program

Automatic compilation of the program

Image 31 : Interface under the FabrikCode with the button “Upload” highlighted

When the program is done, save it, click on the button “compiler”, then, click on the button “ ▶ ” to execute it.*

Manual compilation of the program via KMeleon

When the program is edited and saved, click on the button “COMPILER”.

The program will compile and will be deployed in the KMBox.

Image 32 : Interface under the FabrikCode with the button “Compiler” highlighted

Manual compilation of the program via the powershell terminal

In the search field, write “powershell”.

Open the program “powershell” in the administrator mode. Enter the command “cd CHEMIN_VERS_LE_DOSSIER_DE_VOTRE_PROJET” then, enter the command “mingw32-make clean” puis “mingw32-make”.

Manual deployment of the program

In the windows search field, write "powershell".

Open the program "powershell" in the administrator mode.
Enter the command cd DOSSIER_INSTALLATION_KMELEON/openSSH" then, enter ".\sftp bb@192.168.127.1"
The password corresponds to the last 4 characters of the mac address of the BB400 in the box.
When the connexion is done, enter the command "put -R CHEMIN_VERS_LE_DOSSIER_DE_VOTRE_PROJET/opt/KMApp/bin"

Manual execution of the program

Open your web navigator and then in the search field enter the address 192.168.127.1:9090.

The username is bb and the password corresponds to the last 4 characters of the mac address of the BB400 in the box.

Go into the localhost tab then terminal and enter the following command "cd /opt/KMApp/bin" then, "./KMApp" to start the program.

Then, activate the "startCycle" of the box, with the chosen method.

Edition of a program in c++ language

From the KMeleon's editor program

In the KMeleon menu, click on the button "EDITER". The program editor opens. Click on the button "OUVRIR", then select the file fabrikcode.cpp in [CHEMIN_VERS_LE_DOSSIER_DE_VOTRE_PROJET]. The file opens and you can begin the edition.

When the code is edited click on "ENREGISTRER".

Don't forget to compile the program again.

From sublime-text 3

Go to the spot where is your project file and then, right-click on "fabrikcode.cpp", after click on "ouvrir avec" and select "sublime-text3" in the list.

If the editor is not in c++, go to "view", then, "synthax" and select "c++".

When your code is edited, click on "ENREGISTRER".

Don't forget to compile the program again.

Programming

Blocks connection rules :

A link between two blocks is an edge.
A connexion circle is a dot.

Colours :

Yellow : Variables / Resources (ex : position, decimal)
Orange : Contrôle (Condition, boucle)
Blue : Action (Move, palletizing)
Pink : Sub-program
Purple : Routine

Connexions rules :

- It is prohibited to connect two edges on the same dot.
- It is recommended not to connect chain loops.
- Each dot has a colour and this colour corresponds to the type of block which is allowed on this dot. If these colours are not respected, the code will not work.

To realize your first tests, you can use the different programming models.

Connexion plan :

Pick'n'place loop :

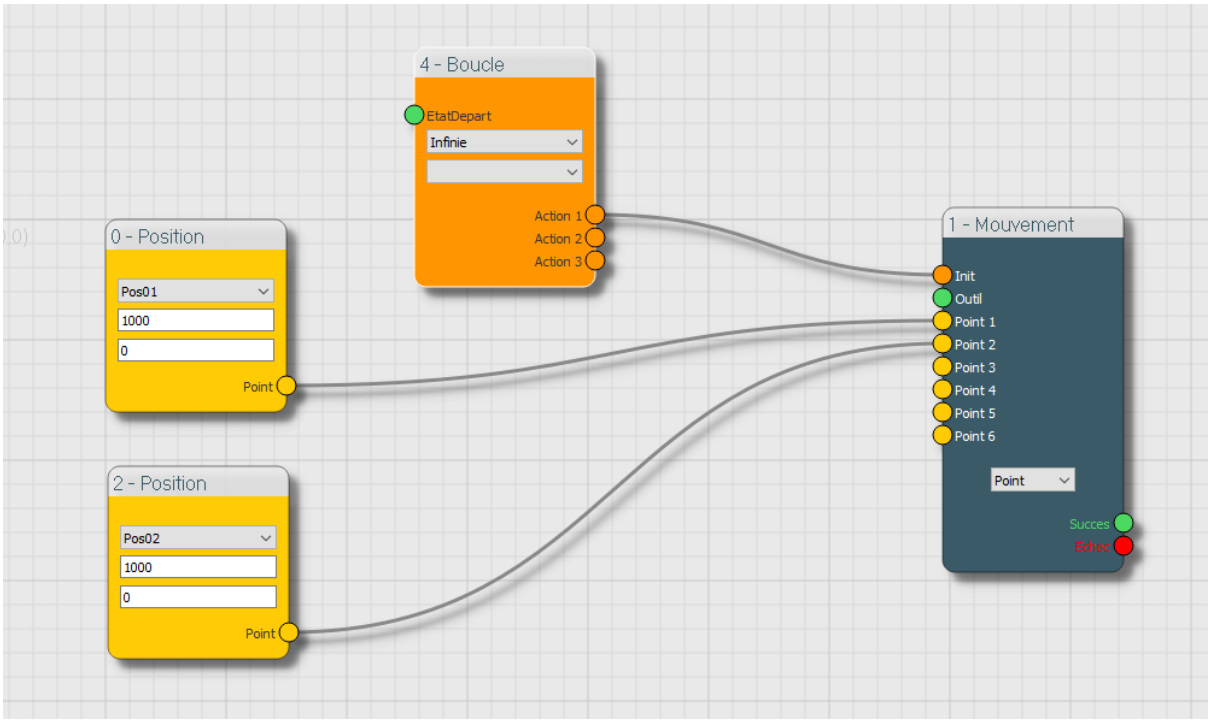


Image 33 : Example Pick'n'place loop plan

In this example, the "position" blocks are connected to the block "movement". You can connect multiple blocks, the number of dots will be adapted with the connexions.

The loop block is connected to the "init" of the movement block. This allows you to place the action in the loop.

Here, the loop block is ordered on "infini" because we want to execute the loop without any interruption.

Pick'n'place with condition :

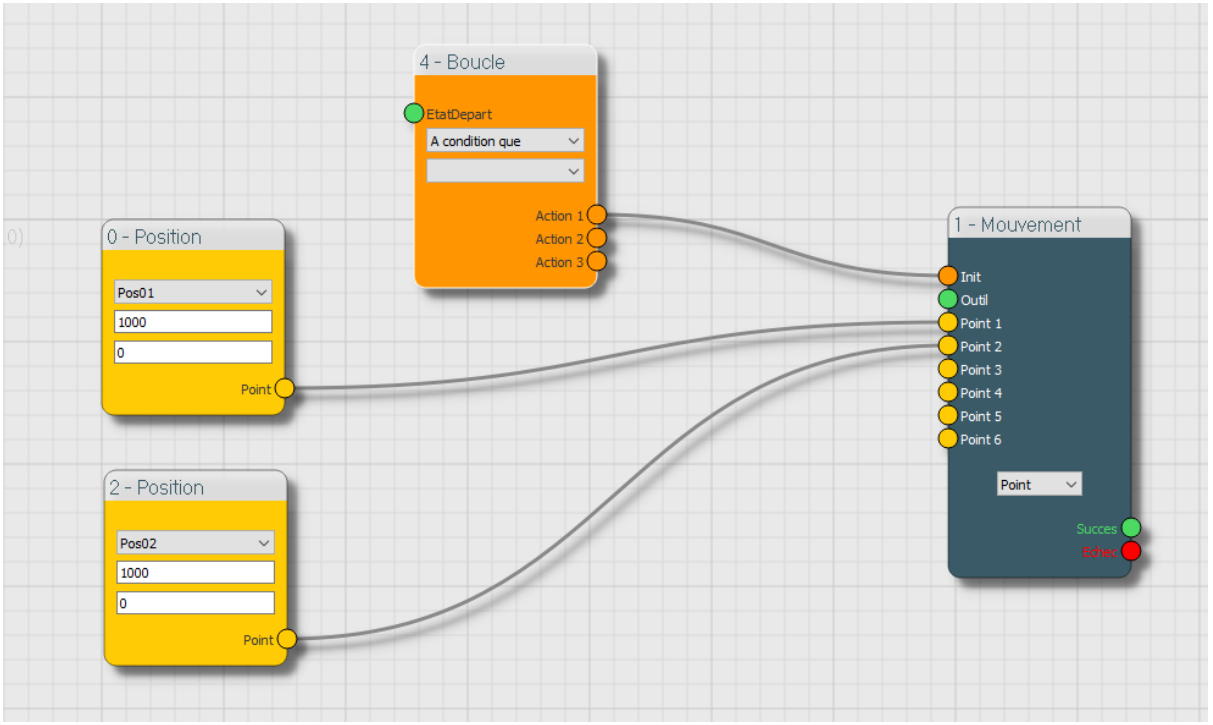


Image 34 : Example of a Pick'n'Place plan with condition

In this example, the blocks "position" are connected to the block "movement". You can connect multiple blocks, the number of dots will be adapted with the connexions.

The loop block is connected to the "init" of the movement block. This allows you to place the action in the loop.

Here, the loop block is ordered on "À condition que " because we want to execute the loop just if one condition is approved.

Pick'n'place action at the end of the move :

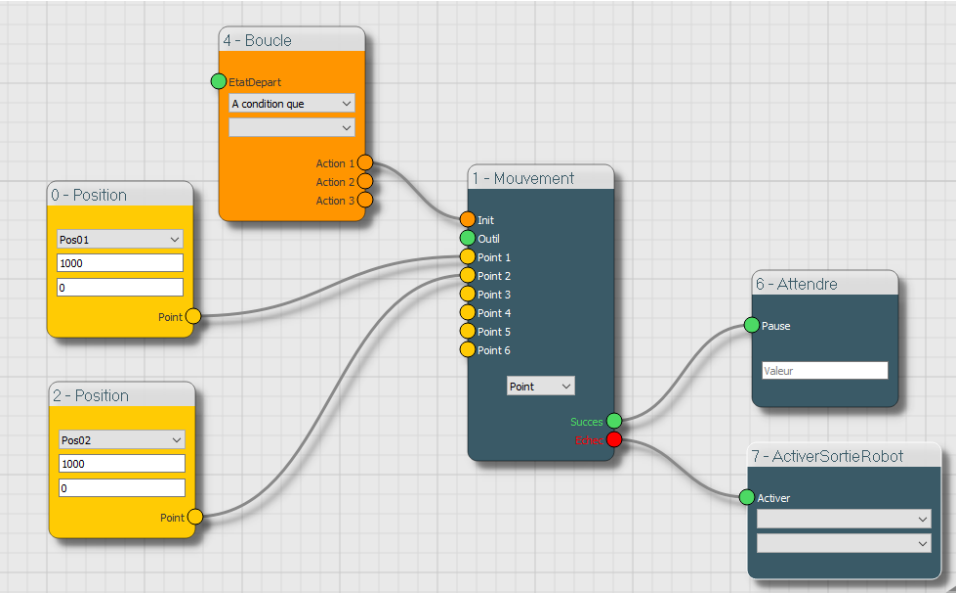


Image 35 : Example of a Pick'n'Place plan with action at the end of the move

In this example, the blocks "position" are connected to the block "movement". You can connect multiple blocks, the number of dots will be adapted with the connexions.

The loop block is connected to the "init" of the movement block. This allows you to place the action in the loop.

Here, the loop block is ordered on "À condition que " because we want to execute the loop just if one condition is approved.

In this example, we added some actions in case of success or failure of the action "movement".

In case of success, the robot will take a little pause. In the case of failure the robot activates an outlet of the robot to mention a default.

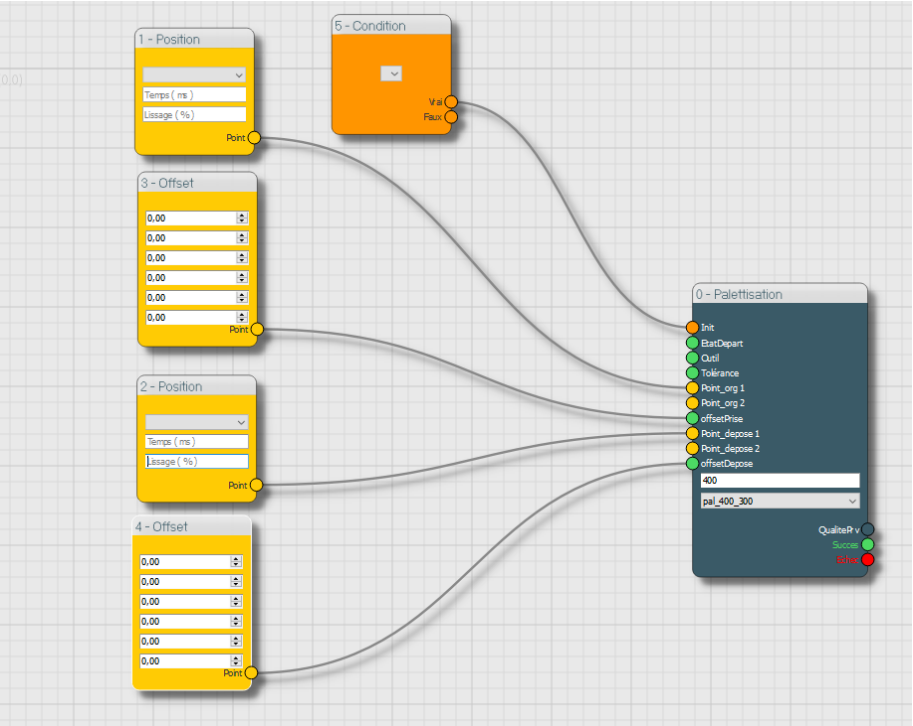


Image 36 : Example of a palletizing plan

In this example, the blocks “position” are connected to the block “palettizing”. You can connect multiple blocks,

You can connect multiple blocks, the number of dots “point_org”, ”point_dépose” will be adapted with the connexions.

The block “condition” is connected to “init” of the block “palletizing”. This allows you to realize the action just if the condition is approved.

The dots “point_org”, ”point_dépose” correspond to the point of engagement and to the point of deposit. Before the grip and the removal, it is possible to add moves, because the dots are incrementing.
The dots “offset prise”,”offset dépose” correspond to the offset made for the grip of the piece and the removal of the piece.

The offset removal is automatically updated according to the height of removal on the pallet.

Sub-program call :

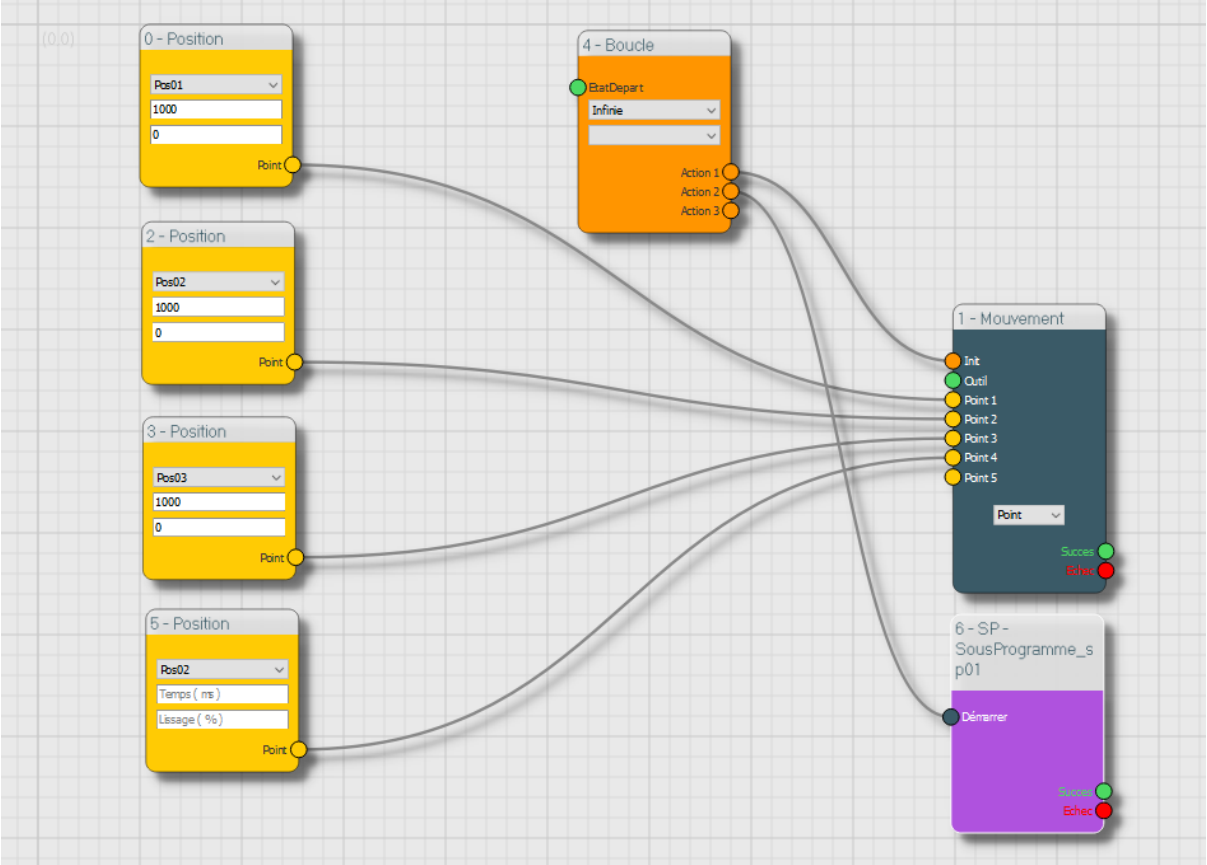


Image 37 : Example of a plan for the sub-program call

You can create a sub-program and call this block in the main program.

Your sub-program must have edges. Moreover, it is strongly recommended not to add loops in the sub-programs to avoid infinite loops. Nevertheless, if it is necessary it remains possible.

The block “loop” and the frame of the program allows you to organize the actions sequence. So, you can add multiple actions, the “action” dots will be adapted with the connexions.

To modify a sub-program, double-click on the block you want to edit.

Glossary

Sensor :

Definition : A sensor is a device which allows you to realize a measure.

Example : *Your oven's temperature sensor is a sensor.*

Actuator:

Definition : An actuator is a device which allows you to make an action.

Example : *Your car's motor is an actuator*

Automaton :

Definition : It is a device which centralizes measure commands which come from sensors and action commands sent to the actuators.

Example : *The system which manages your heating is an automaton. If the heat is too low it activates heating.*

Teach Pendant :

Definition : It is a console which allows you to configure the robot arm.

Inputs :

Definition : An input is a point of connexion between the robot bay or between an automaton and a sensor.

Example : *The thermostat of your heating is linked to an input of the management system.*

Outputs :

Definition : An output is a connexion between the robot bay or an automaton and an actuator.

Example : *Your heatings are linked via an output of the management system.*

Useful contacts

Technical team :

These are the contacts you can call if you have any problem.

Florian DORDAIN
Directeur Technique
florian.dordain@tesseract-solutions.fr

Ouerdia BOURAHLA
Robotic engineer
ouerdia.bouhrala@tesseract-solutions.fr

Nicolas VERDIERE
Robotic engineer
nicolas.verdiere@tesseract-solutions.fr

Commercial team :

These are the contacts you can call if you want to renew your license.

Pénélope TOURNAY
Business Developer
Tesseract solutions
penelope.tournay@tesseract-solutions.fr

Help center :

https://www.tesseract-solutions.fr/knowledge

Any problem or question ?

Contact us