



© Copyright 2019 KUKA France

KUKA FRANCE

Techvallée

6, avenue du Parc

F-91140 Villebon-sur-Yvette

T +33.1.69.31.66.00

F +33.1.69.31.66.01

www.kuka.fr

This documentation or excerpts thereof may not be reproduced or disclosed to third parties without the express permission of KUKA France.

The user has no claims to these functions, however, in the case of a replacement or service work. The user has no claims to these functions, however, in the case of a replacement or service work.

We have checked the content of this documentation for conformity with the hardware and software described.

Differences cannot, however, be totally ruled out. For which reason we are not able to guarantee total conformity. The information in this documentation is checked on a regular basis, however, and necessary corrections will be incorporated in the subsequent edition.

Subject to technical alterations without an effect on the function.

Table of contents

1	Introduction	6
1.1	Scope.....	6
1.2	Industrial robot documentation.....	6
1.3	Representation of comments.....	6
2	Terms used	7
3	Safety	7
4	Product description	9
4.1	The WOP platform	9
4.2	WopCore.....	10
5	Mnemonic writing convention.....	10
6	Installation	11
6.1	Installation of WopCore in WorkVisual	11
6.2	Installing the WopCore option.....	12
7	License activation.....	13
8	Configuration.....	16
8.1	"Cycle / Process" menu	17
8.1.1	"Work type enumeration" parameter	17
8.1.2	"Work status enumeration" parameter	18
8.1.3	"Park request" parameter.....	20
8.1.4	"Request robot stop at end of cycle" parameter	20
8.1.5	"Robot in cycle" parameter.....	21
8.1.6	"Robot in parking position" parameter	21
8.1.7	"Robot in loop position" parameter.....	22
8.2	"Robot Start/Stop" menu.....	23
8.2.1	"Automatic start mode selection" parameter	24
8.2.2	"Robot movement authorization" parameter	27
8.2.3	"Robot cycle start request" parameter.....	28
8.2.4	"Robot ready for start" parameter	28
8.2.5	"Robot cycle immediate stop request" parameter	29
8.2.6	"Power-up request" parameter	29
8.2.7	"Internal cycle start request" parameter.....	30
8.3	"Zone access" menu	31
8.3.1	"Open door and zone access request" parameter	31
8.3.2	"Zone access authorization" parameter	32

8.4	"Messages and errors" menu	33
8.4.1	"General error" parameter	33
8.4.2	"Acknowledgeable message displayed on SmartPAD" parameter	34
8.4.3	"User message return" parameter.....	35
8.4.4	"Number of user messages to return" parameter	35
8.4.5	"Confirm messages" parameter	36
8.5	"Operating mode" menu.....	37
8.5.1	"Robot in automatic mode" parameter	37
8.5.2	"Robot in manual mode" parameter.....	38
8.5.3	"Dry run cycle" parameter.....	38
8.6	"Production management" menu.....	39
8.6.1	"With multiple production management" parameter	40
8.6.2	"Change production request" parameter	40
8.6.3	"Production number requested" parameter	40
8.6.4	"Requested production successfully loaded" parameter	41
8.6.5	"Ready for production change" parameter	42
8.6.6	"Loaded production number" parameter	42
8.6.7	"Production change error number" parameter	43
9	Configuration of messages	44
10	User code.....	46
10.1	"WopUserDat" file: User data	46
10.2	"WopUserSps" file: User program for the background task.....	48
10.2.1	Flag configuration	50
10.2.2	Robot in service.....	50
10.2.3	Robot ready to start.....	51
10.2.4	Robot in manual.....	51
10.2.5	Robot in parking position.....	51
10.2.6	Robot in loop position	52
10.2.7	Robot in Cycle	52
10.2.8	Robot error.....	52
10.2.9	Zone access authorization.....	53
10.2.10	Authorization to execute service trajectories	53
10.2.11	Robot movement authorization	53
10.2.12	Management of the parking request	53
10.2.13	Management of the master program reset request	53
10.2.14	Management of Escape request	53
10.2.15	Management of the start authorization.....	54
10.2.16	Management of the OK start position	54
10.2.17	Robot cycle start request	54
10.2.18	Robot stop request at the end of the cycle.....	54
10.2.19	Request for immediate robot stop	54
10.2.20	Door opening and zone access request.....	54
10.3	"WopUserSrc" file: User code called during trajectory	55
10.3.1	Configure dynamic control	57

■ 11	Generating SRC type error messages.....	59
	11.1 Displaying values in the messages.....	61
	11.2 Program the processing of answers to the dialog messages.....	62
■ 12	Service trajectories.....	64
	12.1 Define the points LoopPos, Parking.....	64
	12.2 Programming a service trajectory.....	65
	12.3 Programming an escape.....	65
■ 13	Productions.....	67
	13.1 Creating a production.....	67
	13.2 Creation of trajectory calls.....	69
	13.3 Creating a trajectory.....	70
	13.4 Associate a trajectory call with a trajectory.....	72
	13.5 Defining the trajectory call conditions.....	73
	13.6 Locking the cycle code equations.....	74
	13.7 Inserting the Job robot form in trajectories.....	75
	13.8 Creating a template for work trajectories.....	77
	13.9 Selecting a production via the SmartPad screen.....	78
■ 14	Management of several productions.....	79
■ 15	Shunt the MoveRobot in progress.....	83
■ 16	List of error codes.....	84
■ 17	KUKA After-sales service.....	87

1 Introduction

1.1 Scope

This documentation is aimed at users with the following knowledge and skills:

- In-depth knowledge of KRL programming.
- In-depth knowledge of the robot control system.



For optimum use of our products, we recommend that our customers take a training course at the KUKA College. Visit our website www.kuka.com or ask one of our branches for any additional information about our training program.

1.2 Industrial robot documentation

The industrial robot documentation consists of the following parts:

- Documentation for the robot mechanical assembly.
- Robot control documentation.
- Service and programming manual for the KUKA System Software.
- Instructions for options and accessories.
- Parts catalog on storage medium.

Each of these sets of instructions is a separate document.

1.3 Representation of comments

Safety

These notes are safety-related, and must therefore be strictly obeyed.



DANGER

These warnings mean that it is certain or highly probable that death or severe injuries will occur, if no precautions are taken.



WARNING

These warnings mean that death or severe injuries may occur, if no precautions are taken.



CAUTION

These warnings mean that minor injuries may occur, if no precautions are taken.

NOTICE

These comments refer to safety-related information or general safety measures. These comments do not refer to specific hazards or safety measures.



These comments refer to important information concerning the use of the application.

Notes

These notes are intended to facilitate your work or refer you to additional information.



Comment to facilitate work or referring to additional information.

2 Terms used

Term	Description
API	PLC (Industrial Programmable Logic Controller)
ES	Emergency Stop.
Auto	Local Automatic mode.
Ext SFC	External automatic mode managed by a local WOP grafcet.
Ext PLC	External automatic mode managed by a remote PLC according to the standard KUKA sequencing cycle.
WOP	Way Of Programming.

3 Safety

This documentation contains safety-relevant notes specifically related to the software described here.

The basic safety-related information for the industrial robot can be found in the chapter "Safety" in the Programming and Operating Instructions for system integrators or in the Programming and Operating Instructions for end users.



Comply with safety-relevant information

The safe use of this product requires knowledge of and compliance with fundamental safety measures. Death, severe injuries or damage to property may otherwise result.

The "Safety" chapter in the operating and programming instructions of the KUKA System Software (KSS) must be observed.



WARNING

Danger to life and limb due to suspended load

When a load is picked up and carried by the gripper, this constitutes a hazardous situation that can lead to death or severe injuries.

- Where possible, perform work outside the space limited by the safeguards.
- If it is necessary to work inside this space, it must be ensured that there are no persons beneath the suspended load

**WARNING****Danger to life and limb due to unforeseen opening of the gripper**

When planning a cell, it must be ensured that the gripper cannot be opened inadvertently. In particular, the gripper must not open automatically, thereby dropping the gripped workpiece, when the power supply is deactivated (e.g. by opening the safety gates). Death, severe injuries or damage to property may otherwise result.

- The system integrator must configure the gripper in such a way that the safe state of the gripper is also maintained in the event of an EMERGENCY STOP, i.e. the gripper remains closed.
- Alternatively, safe inputs and outputs of a PLC can be used so that a gripper output can only be set if the part sensor is actuated:
 - Safe outputs at the actuators on the gripper
 - Safe inputs from sensors on the workpiece support
- Mechanically protect the workspace against falling workpieces.

NOTICE**Material damage due to unchecked gripper behavior**

Once the configuration of all switching states is completed, the real behavior of the gripper must be checked and tested to see if it matches the planned behavior. Damage to property may otherwise result.

Following configuration of the switching states, the behavior of the gripper must be tested by trained personnel

**Preventing system failures**

To prevent hazardous situations and damage, it is necessary to check the condition and functionality of the system regularly.

Death, severe injuries and damage to property may otherwise result.

- Check the system (supply lines, actuators, sensors, etc.) regularly for damage.
- Perform regular system tests.
- Observe the documentation from the system integrator.

**Adopting general safety measures**

When working near the gripper, it must be ensured that body parts, hair, clothing, glasses, jewelry, etc., cannot be caught up in the gripper.

Death, severe injuries or damage to property may result.

- Personal protective equipment must be worn, e.g. safety shoes, safety glasses, etc.
- The local and country-specific accident prevention regulations must be observed.

**Take gripper malfunction into consideration**

If the gripper does not change the switching state, there may be mechanical jamming. In the event of mechanical jamming, it is not possible to predict how the gripper will move.

Death, severe injuries or damage to property may result.



Risk of collision in GhostMode

In GhostMode, it is possible that the programmed switching states will not be set and monitoring functions skipped, resulting in collisions. Death, severe injuries or damage to property may result. Only use GhostMode in operating mode T1 or T2.



Overloading with load management deactivated

Load management prevents overloading of the robot and ensures that the positioning of absolutely accurate robots is correct, even with a load. If load management is inactive, overloading can lead to serious injuries and damage to property.

- Load management must always be activated during operation



For dangerous switching states, the start-up technician can program a critical key. In such a case, the key has to be pressed twice by the operator within a given time before the action is executed.

Conditions that must be fulfilled before an action is executed can also be programmed. The critical key refers only to the 3 status keys that can be assigned.

Modifications to the state via the display of the switching states cannot be programmed as critical. In such a case, the conditions that have to be fulfilled can be programmed.

The following conditions must be met for state modifications via the status keys or the display of the switching states:

- Enabling switch is pressed.
- Program is not active.

4 Product description

4.1 The WOP platform

The WOP platform standardizes and simplifies the deployment of a KUKA robot controller by a KRC4 controller, for any type of application.

A simple and flexible structure in the form of a library lets you select the applications you need and focus on your business process, while allowing the robot to fulfill its potential.

Programming made simple

- The platform offers the user essential functions through automation, online forms, configuration screens, and function keys.

Control and diagnose

- With a user-friendly interface designed with the user's needs in mind, it's easy to learn to operate and to display data on the SmartPad.
- You can program your whole fleet of robots with the same KUKA platform on your SmartPads.

Advantages for integrators

- Reduced and controlled commissioning time.
- Reduced programming time.
- More efficient and responsive support.

Advantages for customers

- Standardized robot structure and behavior.
- A unique training course for maintenance departments.
- Even more responsive KUKA support.
- An improved user experience.

4.2 WopCore

WopCore is the main software package of the WOP platform.

It offers the following functionality and tools:

- Pre-programmed, secure starting of the robot cycle.
- Assistance in the definition of exchanges with the PLC and the environment.
- Assistance in writing the logic for the execution of movements and operations. Programming interface for the trajectory execution conditions
- Assistance in creating specific sets of movements and operations for the production of each part
- Tools to display diagnostic or interaction messages without programming.
- Preprogrammed behaviors that simplify commissioning for the integrator. e.g.: Dynamic control, service trajectory, program initialization, etc.
- Maintenance becomes more efficient thanks to the way folders and files are organized, which is reproduced on several robots.

5 Mnemonic writing convention

Mnemonics of variables are written as follows:

Type_**What**_For whom

Where:

- **Type**: type of variable.
- **What**: name of the variable.
- For whom: for which machine.

For example

- I_ButeeAvancee_Conv1
- O_AvanceButee_Conv1



- Underscore "_" is used as the separator. No more than two underscores per mnemonic.
- All instructions and KUKA system variables are in capitals.

The following table presents the meaning of each type of variable:

Types of variable	Meanings
I_	Inputs
O_	Outputs
M_	Memory residing in the same interpreter (.SRC or .SUB)
X_	Memory exchanged between the two interpreters
ML_	Indicator memory
OL_	Indicator output
C_	Constants
US_	Variables exchanged with a user screen
EP_	Proconos inputs
SP_	Proconos outputs
MP_	Memories exchanged with Proconos
SK_	Memories written by a status key

Programming example

```
IF NOT$IN[I_ButeeAvancee_Conv1[1,1]] AND$T1AND I_BPAvanceButee_Conv1
THEN
$OUT[O_AvanceButee_Conv1[1,1]]=TRUE
ENDIF
```

6 Installation

6.1 Installation of WopCore in WorkVisual

This step makes the WopCore option available in WorkVisual.

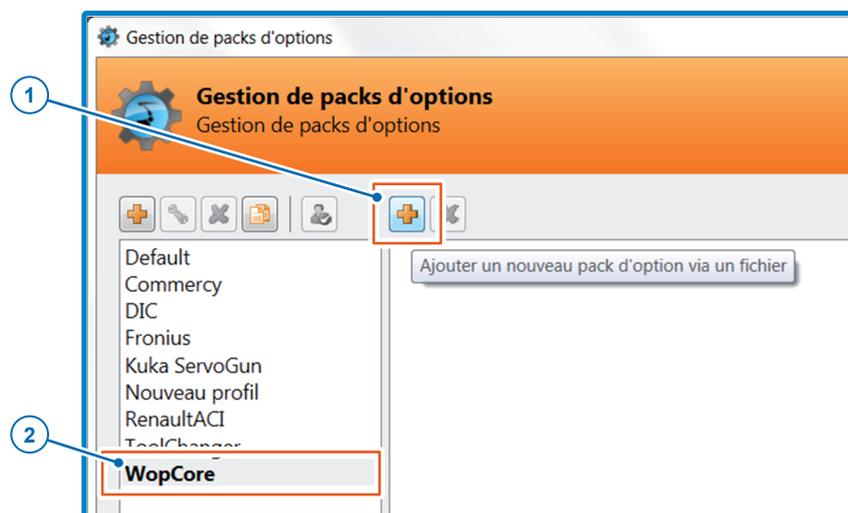
This is necessary before the first installation of the WopCore option in a robot project.

Condition

- KSS in version 8.5.
- WorkVisual V5.0 or higher.
- Must have the WopCore KOP file.

Procedure

- Open WorkVisual
- Go into the menu "Tools\Optional pack management".



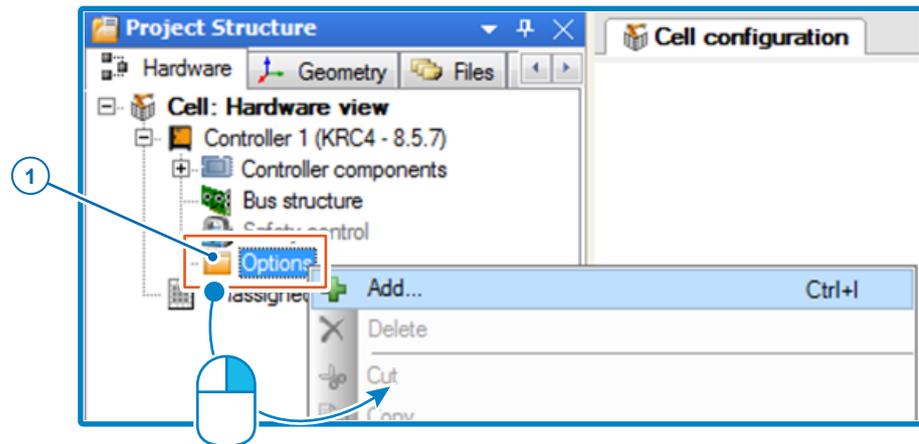
- 1 "Add new option pack via a file" button
 - 2 WopCore KOP file.
- Click "Add new option pack via a file"
 - Select the WopCore KOP file and then click "Open".
 - Confirm each of the steps to finalize the installation.



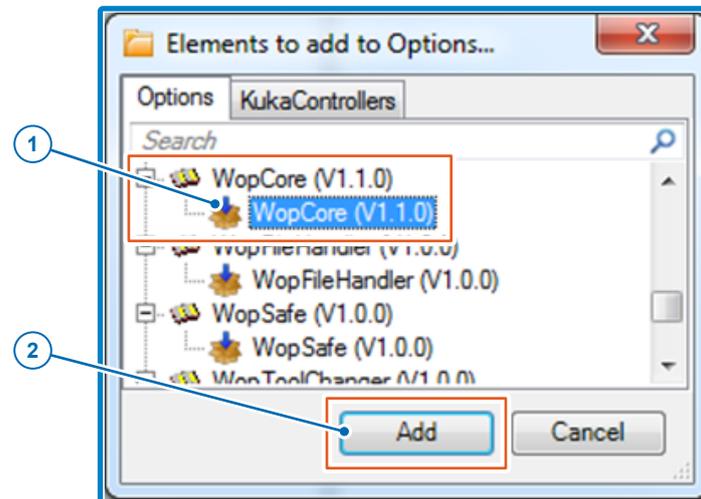
For the sake of convenience, it might be helpful to create a profile of options dedicated to the WOP platform.
To do this, refer to the WorkVisual documentation.

6.2 Installing the WopCore option

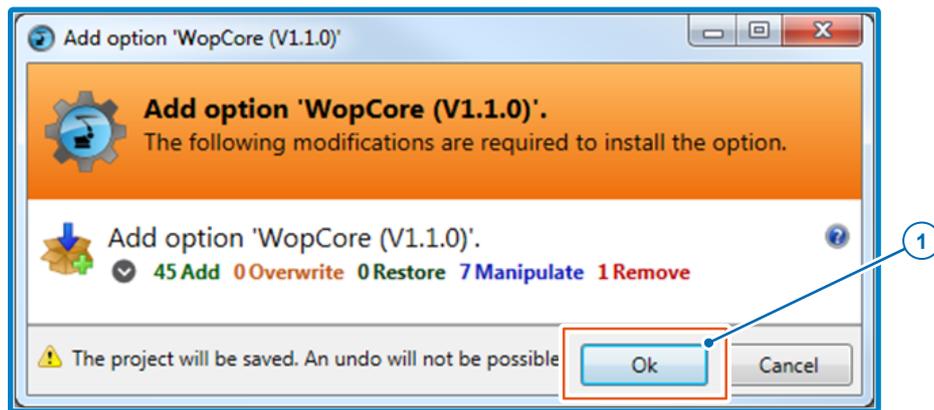
- Download the robot project.



- 1 "Options"
- Select "Options" in the project tree.
 - Right-click to add a new option.



- 1 "WopCore" option
 - 2 "Add" button
- Select "WopCore" from the list of options, and click the "Add" button.



1 "OK" button

- Click the "OK" button to confirm the addition of the WopCore option to the project. The option is now integrated into the project.
- Switch the robot to Expert mode.



1 "Transfer" button

- Click the "Transfer" button to transfer the project to the robot.
- On the robot, when asked "Voulez-vous autoriser l'activation du projet ?" ["Authorize project activation?"], answer "Oui" [Yes].
- Answer "Oui" [Yes] to project modifications.

The robot restarts.

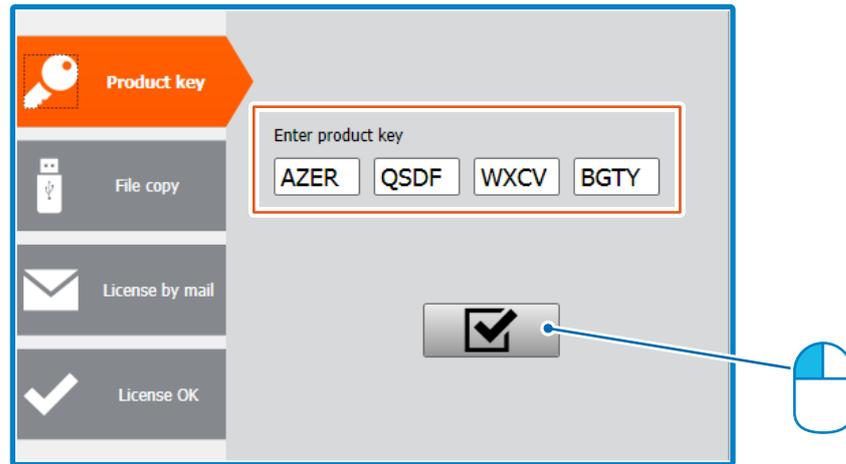
7 License activation

WopCore requires the installation of a license file on the robot. Without it, the robot will be limited to 30% speed in automatic mode, and a screen will appear on the robot every 15 minutes.

Condition

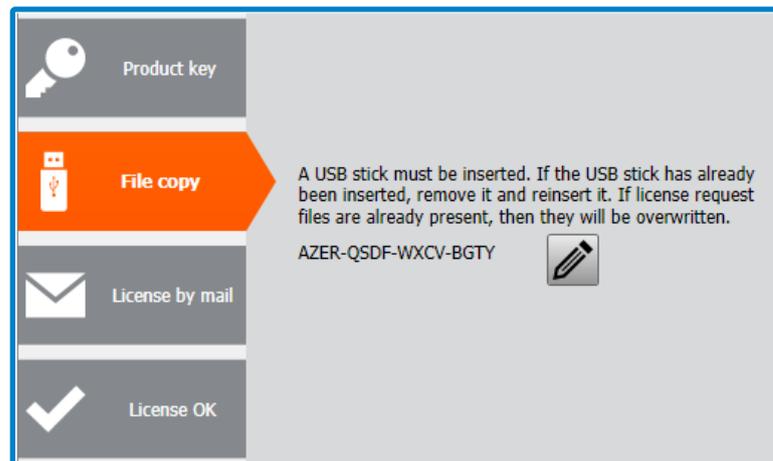
- You must have the correct WopCore Product Key for the robot. If you don't have it, ask your KUKA contact to give it to you.

Procedure



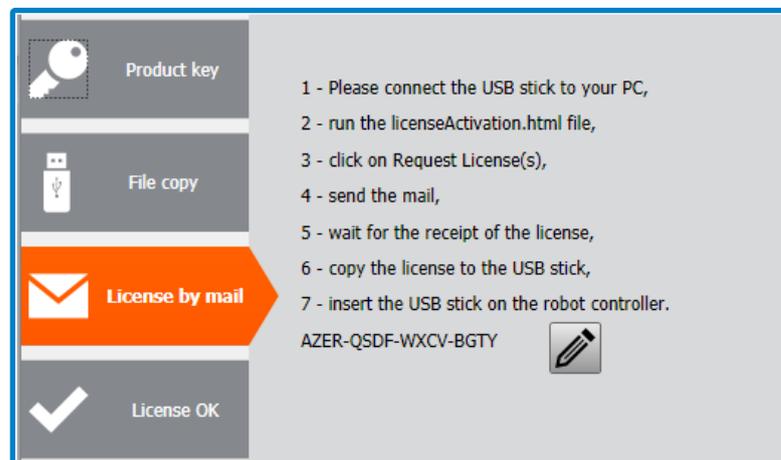
- When the robot starts up for the first time with WopCore, a screen appears and asks you to enter the WopCore Product Key.
- If you have closed that screen, open the menu "WOP – Display\WOP\License WopCore".
- Click the confirmation icon.

A new screen appears, asking you to insert a USB key.



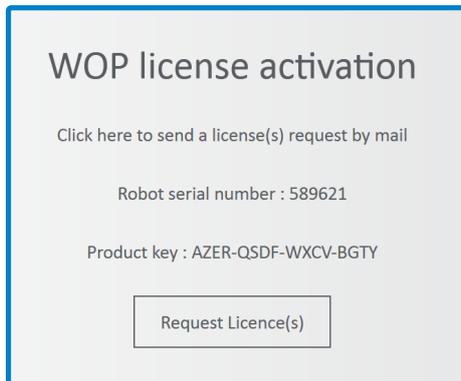
- Insert the USB key into the robot cabinet.
- Some files are automatically copied onto the key.

A new screen appears.



- Remove the USB key.
- Use a PC with an internet connection.

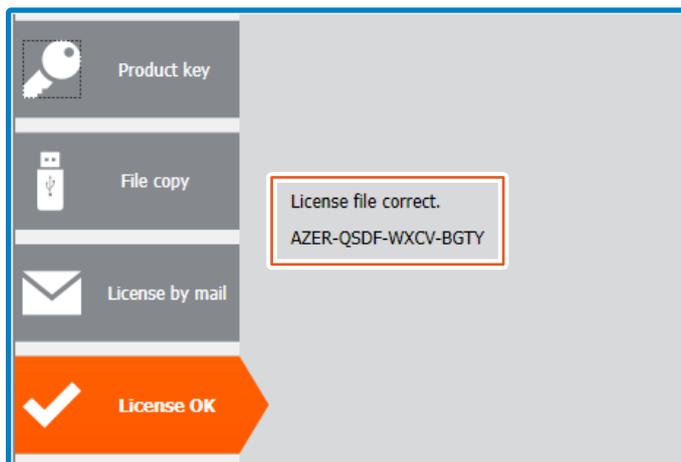
- On the USB key, go into the directory WopLicenses\Robot number.
- Double-click the file licenseActivation.html.
- A web page opens. Click "Request License".



- An e-mail is generated automatically.
- Send the e-mail.
- A few minutes later, you will receive an e-mail with the robot license file.
- Copy this file onto the USB key.
- Connect the key to the robot cabinet.

The files are automatically copied from the key to the robot.

A new screen appears, informing you that the license has been registered.



8 Configuration

Description

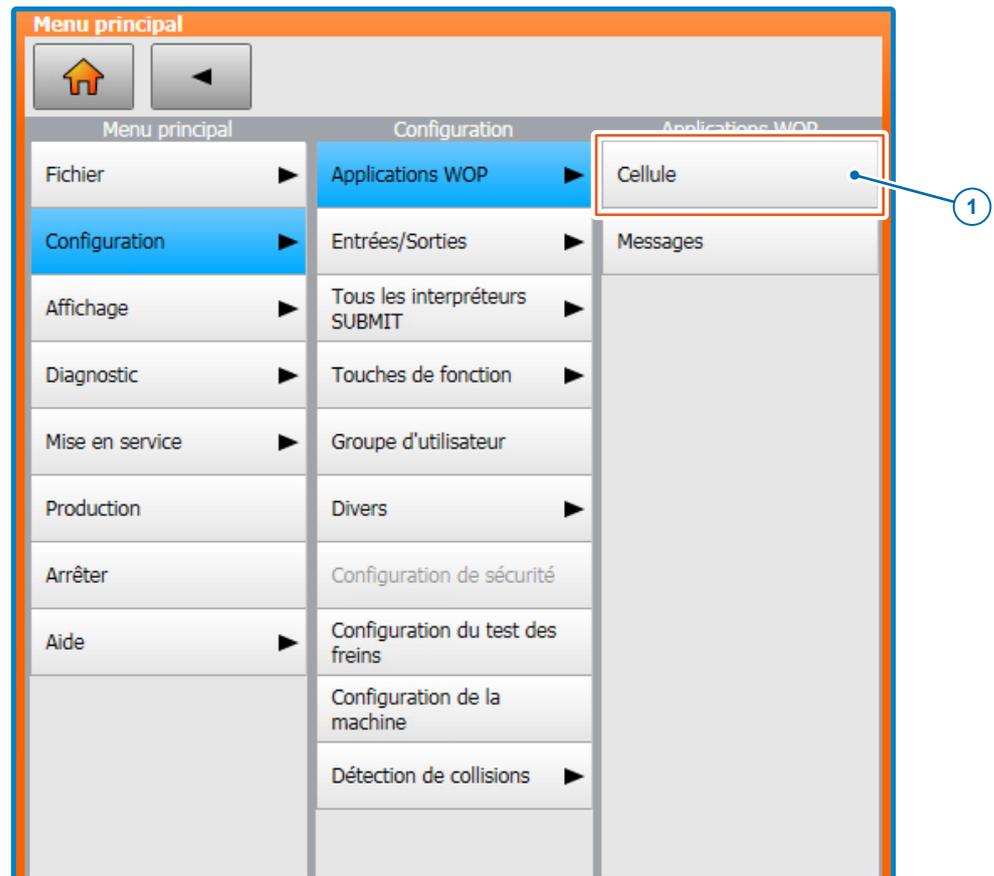
This menu lets you configure the various input and output signals.

It gives access to the configuration screen used to set the various signals and parameters.

Condition

- User group: Expert.

Procedure

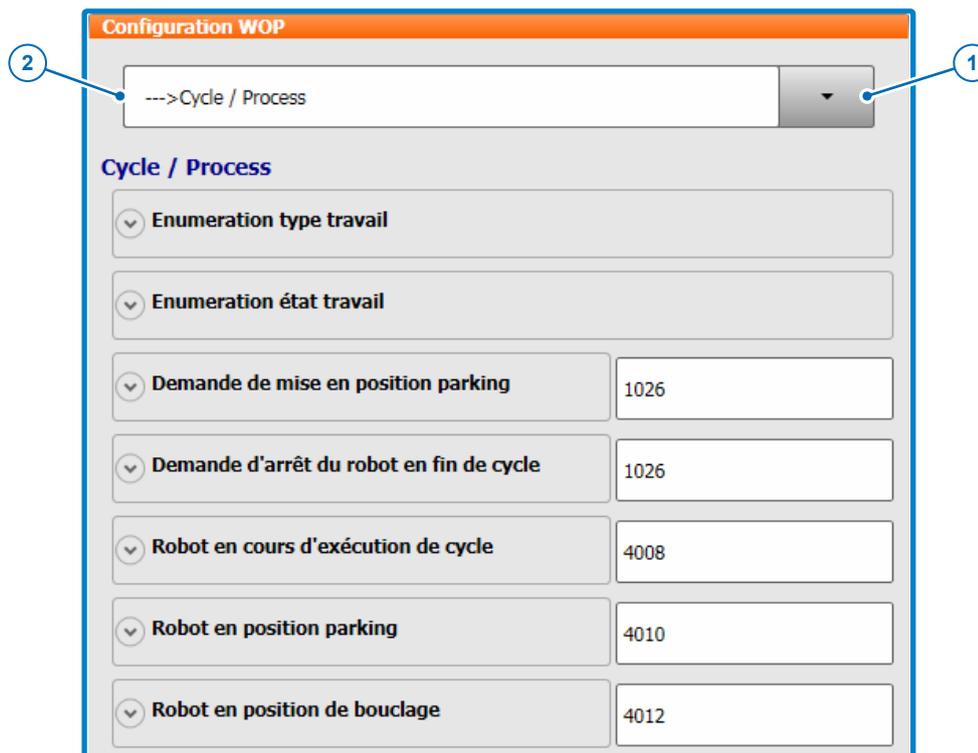


1 "Cell" button

Select the following menu:

- Main menu > Configuration > WOP applications > Cell

8.1 “Cycle / Process” menu



- 1 List button
- 2 “Cycle / Process” menu

- Click the List button (1) to display the “Cycle / Process” menu.
- List of available parameters:

Parameter	Variable
Work type enumeration	EnumWorkRobot
Work state enumeration	EnumStateRobot
Park request	I_Parking_Request
Request robot stop at end of cycle	I_EndCycle_Request
Robot in cycle	O_Rob_InCycle
Robot in parking position	O_Rob_Parking
Robot in loop position	O_Rob_LoopPos1

8.1.1 “Work type enumeration” parameter

Description

This parameter is optional.

This list enumerates the robot's tasks. It is used by an online form to indicate the task in progress.



Expert information

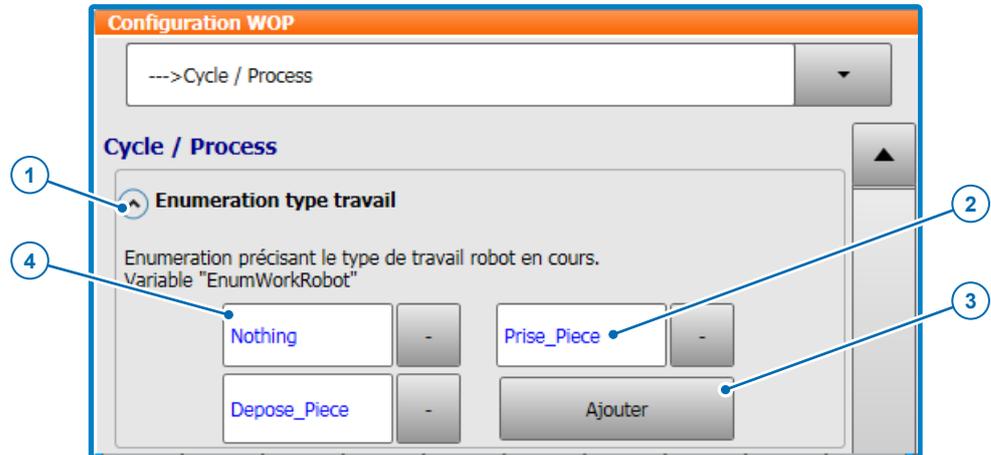
The variable associated with this parameter is: EnumWorkRobot.

- Variable used: EnumWorkRobot.

Condition

- User group: Expert.

Procedure



- 1 "Work type enumeration" parameter
- 2 Input field
- 3 "Add" button
- 4 "Nothing" enumeration

- In the input field (2), input a description of the type of work.
- To add an enumeration, click the "Add" button (3).



If an unauthorized character is input, the border of the input field turns red.

Example of a work declaration: `Prise_Piece`, `Depose_Piece`.



Important Note:

- The declaration "Nothing" must be left on the declaration of enumerations (4). Forgetting this "Nothing" declaration (4) causes an error in the WopUserDat file.

8.1.2 "Work status enumeration" parameter

Description

This list enumerates the robot's task statuses. It is used by an online form to indicate the work state in progress.



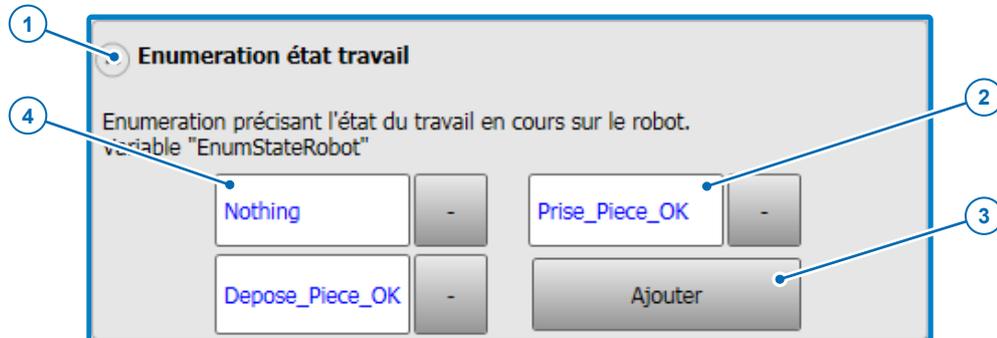
Expert information

The variable associated with this parameter is: EnumStateRobot.

Condition

- User group: Expert.

Procedure



- 1 "Work status enumeration" parameter
- 2 Input field
- 3 "Add" button
- 4 "Nothing" enumeration

- In the input field (2), input a description of the status of the work in progress on the robot.
- To add an enumeration, click the "Add" button (3).



If an unauthorized character is input, the border of the input field turns red.

Example of a work state declaration: `Prise_Piece_OK`.



Important Note:

- The declaration "Nothing" must be left on the declaration of enumerations (4). Forgetting this "Nothing" declaration (4) causes an error in the WopUserDat file.

8.1.3 “Park request” parameter

Description

This setting lets you define the address of the parking request input.

This reformulation applies to all inputs and outputs!

The service request function is already programmed in WopCore.



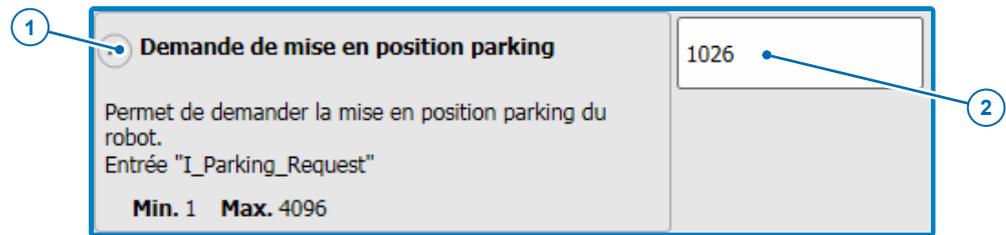
Expert information

The variable associated with this parameter is: I_Parking_Request.

Condition

- User group: Expert.

Procedure



- 1 “Park request” parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1026.



If an unauthorized character is input, the border of the input field turns red.

8.1.4 “Request robot stop at end of cycle” parameter

Description

This parameter is used to ask for the robot cycle to stop at the end of the current trajectory.



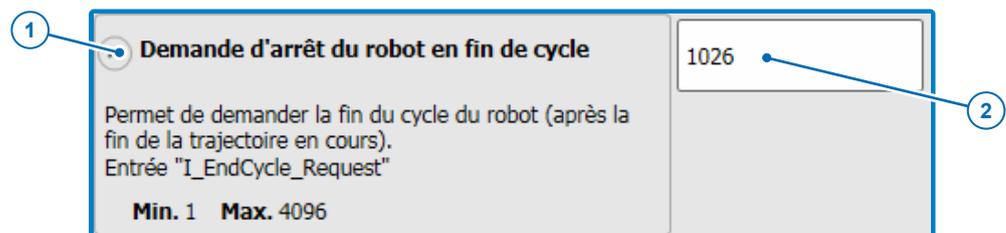
Expert information

The variable associated with this parameter is: I_EndCycle_Request.

Condition

- User group: Expert.

Procedure



- 1 “Request robot stop at end of cycle” parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1026.

 If an unauthorized character is input, the border of the input field turns red.

8.1.5 “Robot in cycle” parameter

Description

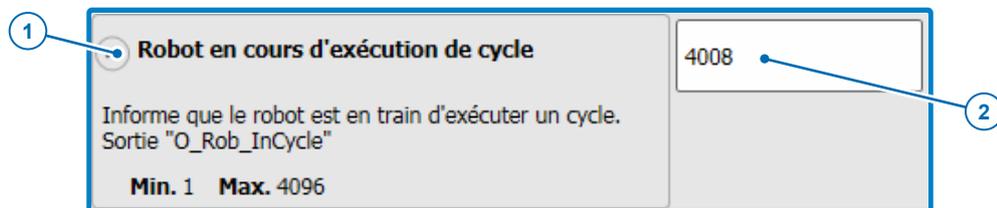
This parameter gives notification that the robot is currently executing a cycle.

 **Expert information**
The variable associated with this parameter is: O_Rob_InCycle.

Condition

- User group: Expert.

Procedure



- 1 “Robot in cycle” parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4008.

 If an unauthorized character is input, the border of the input field turns red.

8.1.6 "Robot in parking position" parameter

Description

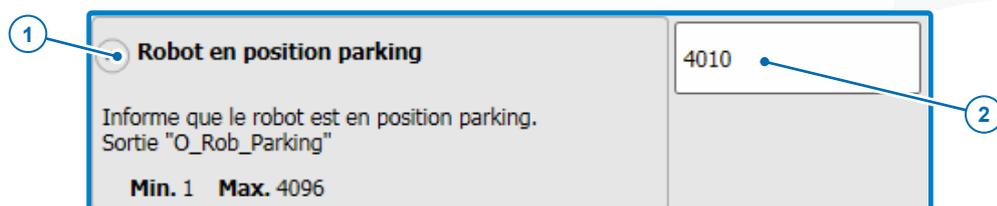
This parameter indicates that the robot is in parking position.

 **Expert information**
The variable associated with this parameter is: O_Rob_Parking.

Condition

- User group: Expert.

Procedure



- 1 "Robot in parking position" parameter

2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4010.

 If an unauthorized character is input, the border of the input field turns red.

8.1.7 "Robot in loop position" parameter

Description

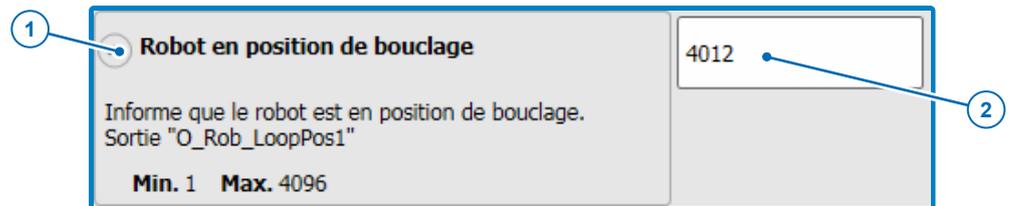
This parameter indicates that the robot is in loop position.

 **Expert information**
The variable associated with this parameter is: O_Rob_LoopPos1.

Condition

- User group: Expert.

Procedure

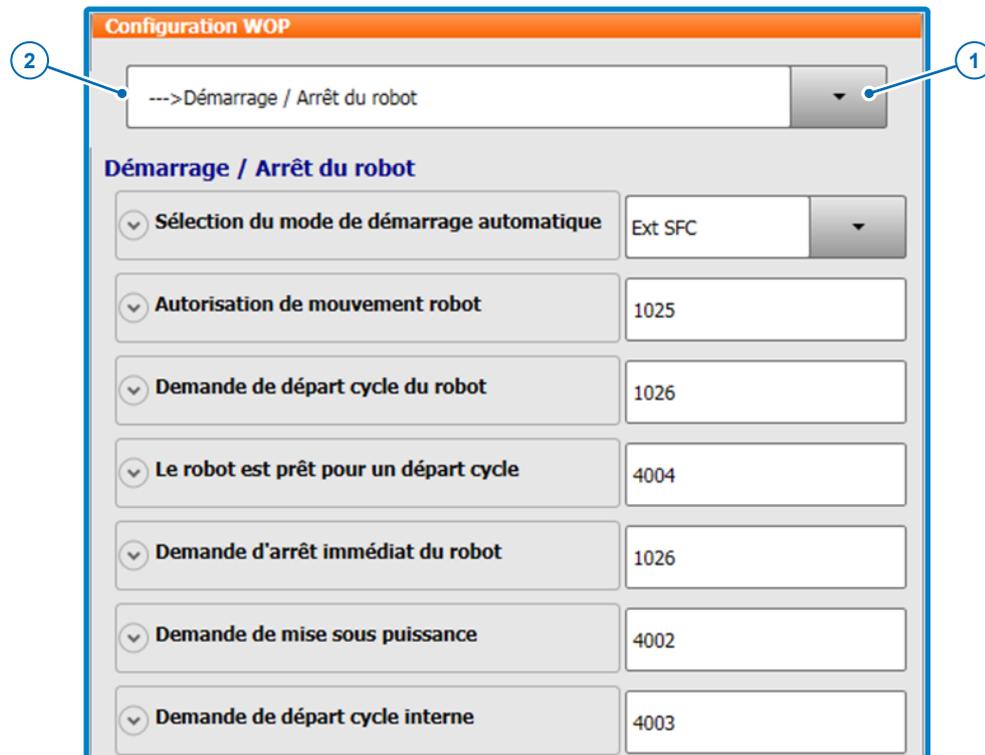


- 1 "Robot in loop position" parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4012.

 If an unauthorized character is input, the border of the input field turns red.

8.2 “Robot Start/Stop” menu



- 1 List button
- 2 “Robot Start/Stop” menu

- Click the List button (1) to display the “Robot Start/Stop” menu.
- List of available parameters:

Parameter	Variable
Automatic start mode selection	Mode: Auto / Ext SFC / Ext PLC
Robot movement authorization	I_MoveRobot_Ok
Robot cycle start request	I_Cycle_Start
Robot ready for start	O_ReadyFor_Start
Robot immediate stop request	I_StopCycle_Request
Power-up request *	O_Drives_On
Internal cycle start request *	O_External_Start



(*) Valid only if the robot automatic start mode is set to “Ext SFC”.

8.2.1 “Automatic start mode selection” parameter

Description

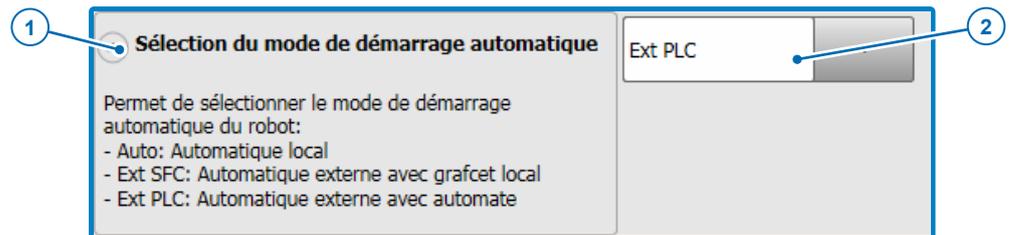
This parameter is used to select the robot's automatic start mode.

- Auto : Local automatic.
- Ext SFC : External automatic managed by a local WOP grafcet (SFC)
- Ext PLC : External automatic managed by a remote PLC according to the standard KUKA sequencing cycle.

Condition

- User group: Expert.

Procedure



- 1 “Automatic start mode selection” parameter
- 2 Selection list

- Select the robot's automatic start mode from the list (2).

Launch with API signal management

- X_Start_Mode=#Ext_Plc

The following system output signals must be configured directly in the standard external automatic part of the robot.

- \$DRIVES_ON
- \$CONF_MESS
- \$EXT_START
- \$PERY_RDY
- \$STOPMESS
- \$PRO_ACT

For a trouble-free robot startup, you must strictly apply the chronogram in the KSS documentation.

Launch with signal management by the robot

- X_Start_Mode=#Ext_SFC

In this case, the robot startup process is completely managed by the robot.

You must configure the following signal input number:

- I_Cycle_Start
- O_Conf_Mess
- O_Drives_On
- O_External_Start
- \$DRIVES_ON
- \$CONF_MESS
- \$EXT_START

Connect the WOP outputs to the system inputs, either physically on the input and output boards, or directly in the PLC:

- O_Conf_Mess -> \$CONF_MESS
- O_Drives_On -> \$DRIVES_ON
- O_External_Start -> \$EXT_START

A 500-ms pulse on the I_Cycle_Start user input allows the robot to be started up by an internal launch grafcet, which then follows the robot startup chronograms.

Launch in internal automatic mode

- X_Start_Mode=#Aut

In this case, the robot startup process is managed entirely manually.

With X_Start_Mode=#Aut, powering up and starting the robot are accessible from the SmartPad.

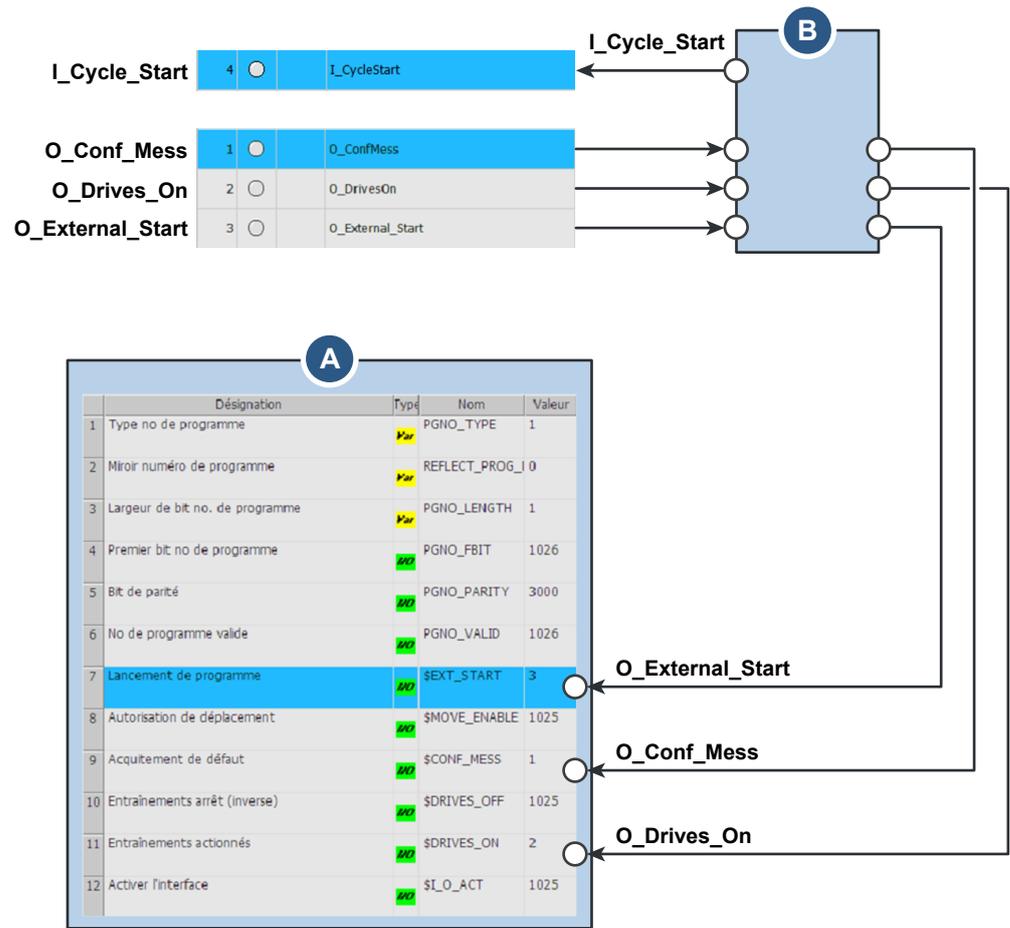


Important Note:

- If the chosen start mode is **X_Start_Mode=#Aut**, the system input **\$MOVE_ENABLE** must be set to the value 1025, or to an external input from a PLC, for example.
- The system variable **STEUOPTION.DAT\CHCK_MOVENA** must be in the state FALSE.

Connection via a PLC

Example :



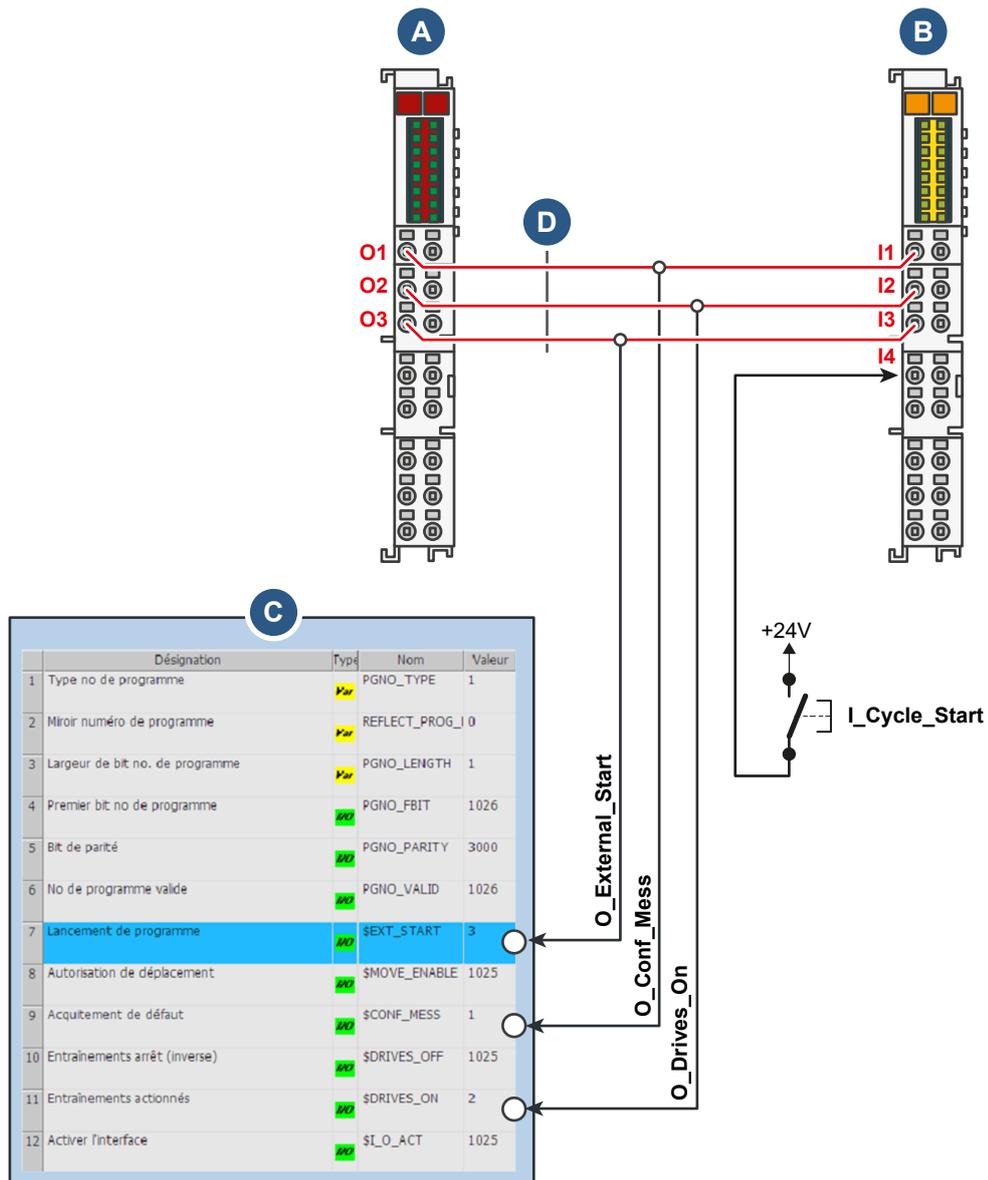
- A External automatic configuration
- B PLC

Connection without PLC

Example :



- Connect output 1 to input 1.
- Connect output 2 to input 2.
- Connect output 3 to input 3.
- Connect I_Cycle_Start to input 4.



- A Outputs
- B Inputs
- C External automatic configuration
- D Bypass

8.2.2 "Robot movement authorization" parameter

Description

This parameter lets you authorize robot movements.
Parameter active in modes T1, T2, AUT and EXT.



Expert information

The variable associated with this parameter is: I_MoveRobot_Ok.

Condition

- User group: Expert.

Procedure

- 1 "Robot movement authorization" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1025.

 If an unauthorized character is input, the border of the input field turns red.

8.2.3 “Robot cycle start request” parameter

Description

This parameter lets you start the robot cycle.

 **Expert information**
The variable associated with this parameter is: I_Cycle_Start.

Condition

- User group: Expert.

Procedure

- 1 “Robot cycle start request” parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1026.

 If an unauthorized character is input, the border of the input field turns red.

8.2.4 “Robot ready for start” parameter

Description

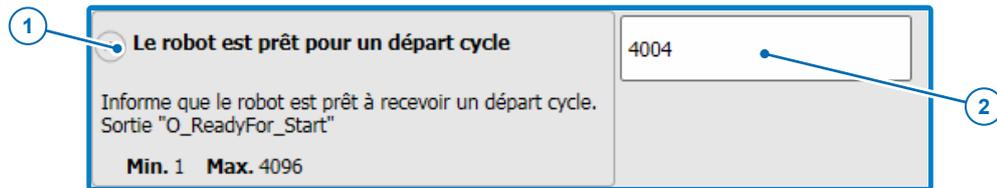
This parameter gives notification that the robot ready for a cycle start command.

 **Expert information**
The variable associated with this parameter is: O_ReadyFor_Start.

Condition

- User group: Expert.

Procedure



1 Le robot est prêt pour un départ cycle

Informe que le robot est prêt à recevoir un départ cycle.
Sortie "O_ReadyFor_Start"

Min. 1 Max. 4096

4004

2

- 1 "Robot ready for start" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4004.

 If an unauthorized character is input, the border of the input field turns red.

8.2.5 "Robot cycle immediate stop request" parameter

Description

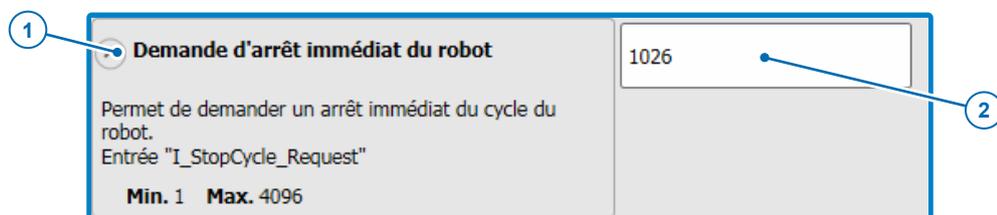
This parameter is used to request the immediate halting of the robot cycle.

 **Expert information**
The variable associated with this parameter is: I_StopCycle_Request.

Condition

- User group: Expert.

Procedure



1 Demande d'arrêt immédiat du robot

Permet de demander un arrêt immédiat du cycle du robot.
Entrée "I_StopCycle_Request"

Min. 1 Max. 4096

1026

2

- 1 "Robot cycle immediate stop request" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1026.

 If an unauthorized character is input, the border of the input field turns red.

8.2.6 "Power-up request" parameter

Description

This parameter lets the robot manage power-up when the automatic startup mode is set to "Ext SFC". The output must be physically connected to the DRIVE_ON input.

 **Expert information**
The variable associated with this parameter is: O_Drives_On.

Condition

- User group: Expert.

Procedure

- 1 "Power-up request" parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4002.

 If an unauthorized character is input, the border of the input field turns red.

8.2.7 "Internal cycle start request" parameter

Description

This parameter lets the robot manage the cycle start when the automatic startup mode is set to "Ext SFC". The output must be physically connected to the EXT_START input.

 **Expert information**
The variable associated with this parameter is: O_External_Start.

Condition

- User group: Expert.

Procedure

- 1 "Internal cycle start request" parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4003.

 If an unauthorized character is input, the border of the input field turns red.

8.3 "Zone access" menu

Description

The access request allows the operator to inform the robot that he/she wants to enter the secure enclosure. The system then steers the robot to its parking spot when the work in progress is finished. The system then stops the program, disables the power, and authorizes the opening of the zone.

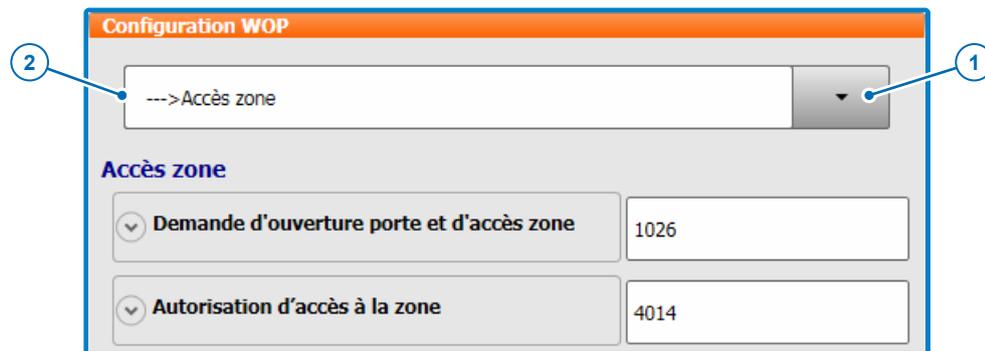
Access to the robot area is authorized via the O_Access_Allowed robot output.

If the robot is in manual T1 or T2 or emergency stop, access to the robot zone is continuously authorized.

The access request function is already programmed in WopCore.

Simply enter the input and output signals concerning the access request.

Procedure



- 1 List button
- 2 "Access zone" menu

- Click the List button (1) to display the "Zone access" menu.
- List of available parameters:

Parameter	Variable
Open door and zone access request	I_Access_Request
Zone access authorization	O_Acces_Allowed

8.3.1 "Open door and zone access request" parameter

Description

This parameter is used to request opening of the zone during the automatic cycle.



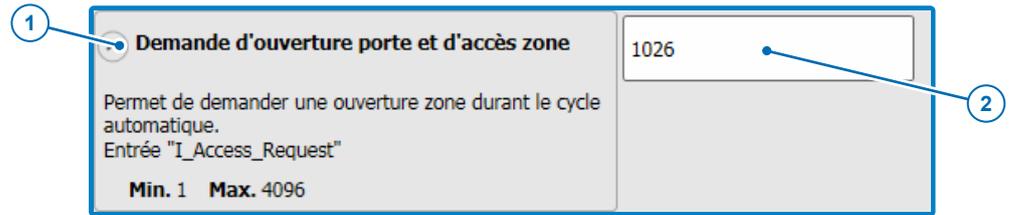
Expert information

The variable associated with this parameter is: I_Access_Request.

Condition

- User group: Expert.

Procedure



- 1 "Open door and zone access request" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1026.

 If an unauthorized character is input, the border of the input field turns red.

8.3.2 "Zone access authorization" parameter

Description

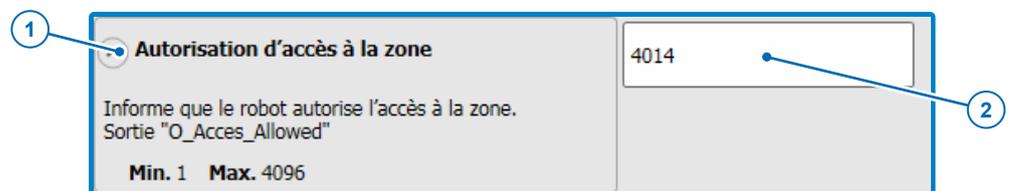
This parameter indicates that the robot authorizes access to the zone

 **Expert information**
The variable associated with this parameter is: O_Access_Allowed.

Condition

- User group: Expert.

Procedure



- 1 "Zone access authorization" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4014.

 If an unauthorized character is input, the border of the input field turns red.

8.4 “Messages and errors” menu



- 1 List button
- 2 “Messages and errors” menu

- Click the List button (1) to display the “Messages and errors” menu.
- List of available parameters:

Parameter	Variable
General error	O_GeneralError_Robot
Acknowledgeable message displayed on SmartPAD	O_DlgMessage_OnRobot
User message return	M_FirstBitMessage_ToPlc
Number of user messages to return	M_LengthMessage_ToPlc
Message acknowledgement request	O_Conf_Mess

8.4.1 “General error” parameter

Description

This parameter is used to inform an external system that a general error is present on the robot.



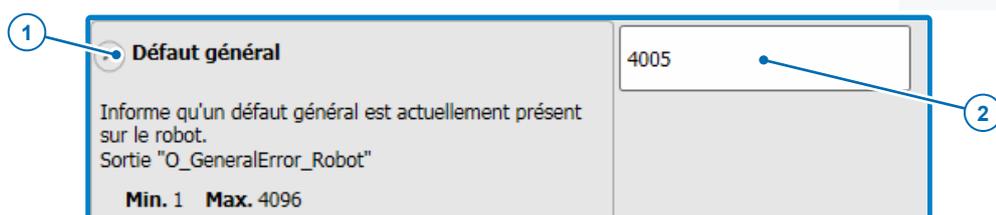
Expert information

The variable associated with this parameter is: O_GeneralError_Robot.

Condition

- User group: Expert.

Procedure



- 1 “General error” parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4005.

 If an unauthorized character is input, the border of the input field turns red.

8.4.2 “Acknowledgeable message displayed on SmartPAD” parameter

Description

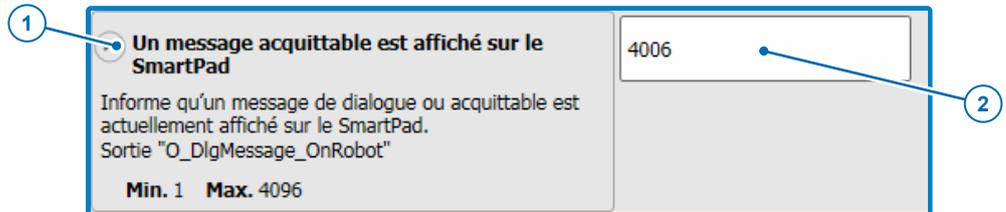
This parameter gives notification that a dialog message or acknowledgeable message is displayed on the SmartPAD.

 **Expert information**
The variable associated with this parameter is: O_DlgMessage_OnRobot.

Condition

- User group: Expert.

Procedure



- 1 “Acknowledgeable message displayed on SmartPAD” parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4006.

 If an unauthorized character is input, the border of the input field turns red.

8.4.3 “User message return” parameter

Description

This parameter is used to define the first output for the return of user messages displayed on the SmartPAD.



Expert information

The variable associated with this parameter is: M_FirstBitMessage_ToPlc.

Configuration example:

- First_Bit_FaultCode =50
- M_LengthMessage_ToPlc =3

When Error 1 is displayed, Output 50 is set to the state TRUE.

When Error 2 is displayed, Output 51 is set to the state TRUE.

When Error 3 is displayed, Output 52 is set to the state TRUE.

If only Errors 1 and 3 are displayed, the state of Outputs 50 and 52 is TRUE.



If M_LengthMessage_ToPlc has a shorter length than the number of faults configured, only faults corresponding to the length declared in FaultCode_Length will be sent.

Condition

- User group: Expert.

Procedure



- 1 “User message return” parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4021.



If an unauthorized character is input, the border of the input field turns red.

8.4.4 “Number of user messages to return” parameter

Description

This parameter is used to define the number of user messages to return.



Expert information

The variable associated with this parameter is: M_LengthMessage_ToPlc.

See configuration example in Section 8.4.5.

Condition

- User group: Expert.

Procedure

- 1 "Number of user messages to return" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 64.
 - Default value: 16.

 If an unauthorized character is input, the border of the input field turns red.

8.4.5 "Confirm messages" parameter

Description

This parameter lets the robot automatically acknowledge messages when the automatic startup mode is set to "Ext SFC". The output must be physically connected to the \$CONF_MESS input. Not all messages are acknowledgeable.

 **Expert information**
The variable associated with this parameter is: O_Conf_Mess.

Condition

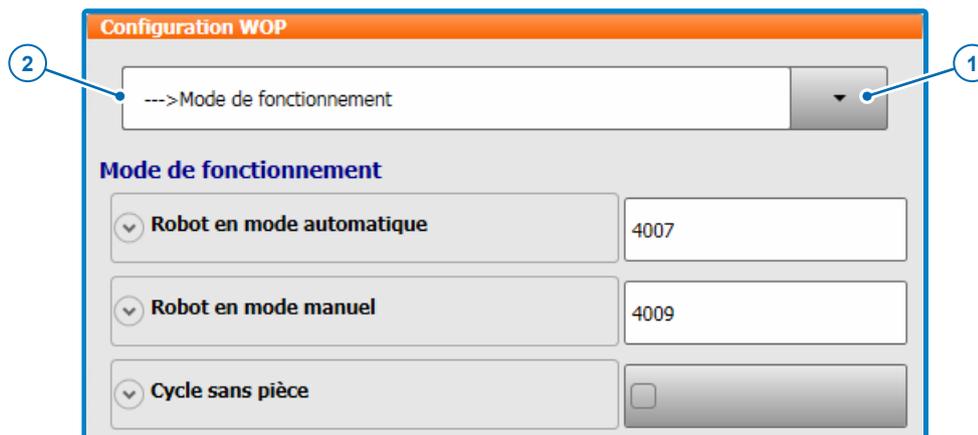
- User group: Expert.

Procedure

- 1 "Confirm messages" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4001.

 If an unauthorized character is input, the border of the input field turns red.

8.5 "Operating mode" menu



- 1 List button
- 2 "Messages and errors" menu

- Click the List button (1) to display the "Operating mode" menu.
- Available parameters:

Parameter	Variable
Robot in automatic mode	O_Rob_RunInAuto
Robot in manual mode	O_Rob_Manu
Dry run cycle	X_Cycle_DryRun

8.5.1 "Robot in automatic mode" parameter

Description

This parameter is used to inform an external system that the robot is in automatic operating mode, powered, and running a program.



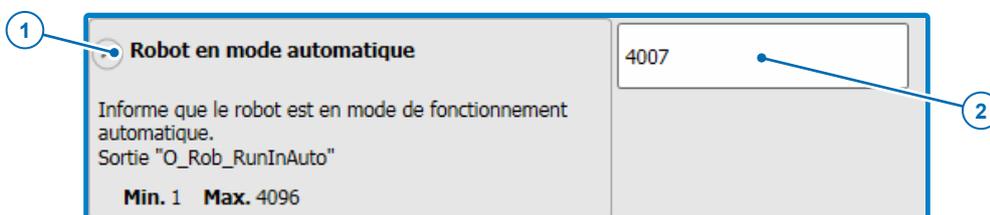
Expert information

The variable associated with this parameter is: O_Rob_RunInAuto.

Condition

- User group: Expert.

Procedure



- 1 "Robot in automatic mode" parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4007.



If an unauthorized character is input, the border of the input field turns red.

8.5.2 "Robot in manual mode" parameter

Description

This parameter is used to inform an external system that the robot is in manual operating mode (selector set to T1 or T2).



Expert information

The variable associated with this parameter is: O_Rob_Manu.

Condition

- User group: Expert.

Procedure

1 "Robot in manual mode" parameter

2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4009.



If an unauthorized character is input, the border of the input field turns red.

8.5.3 "Dry run cycle" parameter

Description

This parameter is used to run a cycle without checks on the part. This lets you bypass the checks in certain WOP applications, such as "Gripper" and "Vacuum".



Expert information

The variable associated with this parameter is: X_Cycle_DryRun.

Condition

- User group: Expert.

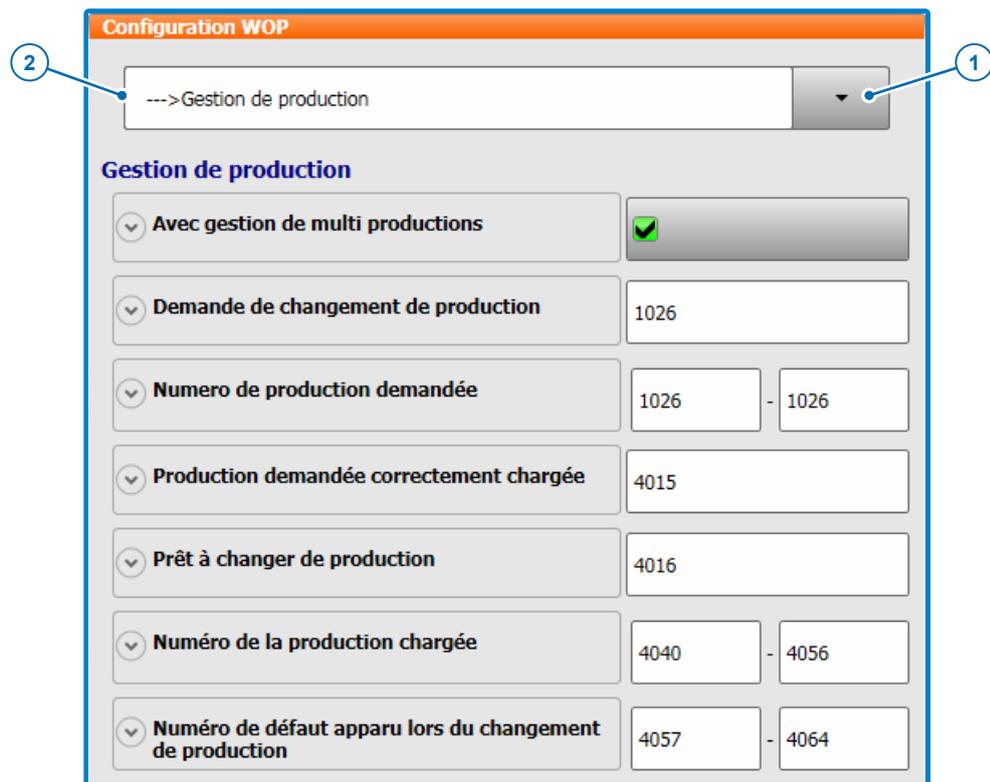
Procedure

1 "Dry run cycle" parameter

2 Checkbox

- Check the box (2) to select the parameter.
 - Default value: FALSE.

8.6 “Production management” menu



- 1 List button
- 2 “Production management” menu

- Click the List button (1) to display the “Production management” menu.
- Available parameters:

Parameter	Variable
With multiple production management	X_ProductionMode_Enable
Change production request*	I_LoadProd_Request
Production number requested*	I_NumLoadProd_Request
Requested production successfully loaded*	O_LoadProd_Loaded
Ready for production change*	O_ReadyFor_LoadProd
Loaded production number*	O_NumLoadProd_Loaded
Error number displayed when changing production*	O_LoadProd_Error



(*) Valid only if the management of several productions is selected.

8.6.1 “With multiple production management” parameter

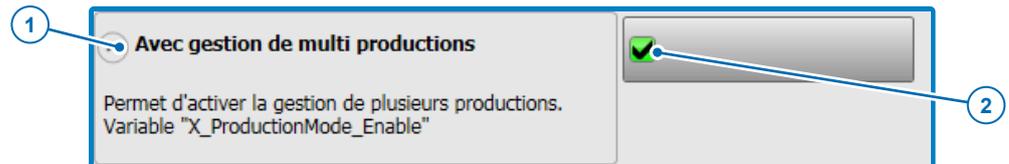
Description

This parameter is used to enable the management of several productions (see Section 14, page 79).

Expert information
 The variable associated with this parameter is: X_ProductionMode_Enable.

- User group: Expert.

Procedure



- 1 “With multiple production management” parameter
- 2 Checkbox

- Check the box (2) to select the parameter.
 - Default value: FALSE.

8.6.2 “Change production request” parameter

Description

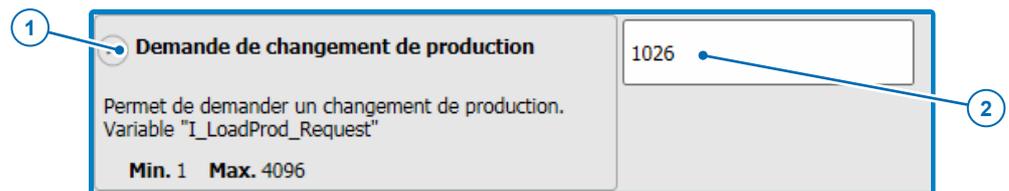
This parameter is used to request a production change.

Expert information
 The variable associated with this parameter is: I_LoadProd_Request.

Condition

- User group: Expert.

Procedure



- 1 “Change production request” parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1026.

Expert information
 If an unauthorized character is input, the border of the input field turns red.

8.6.3 “Production number requested” parameter

Description

This parameter is used to specify the desired production number.

Expert information
 Signal on 8 bits at least (maximum 32 bits).

**Expert information**

The variable associated with this parameter is: I_NumLoadProd_Request.

Condition

- User group: Expert.

Procedure

- 1 "Production number requested" parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 1026 - 1026.



If an unauthorized character is input, the border of the input field turns red.

8.6.4 "Requested production successfully loaded" parameter**Description**

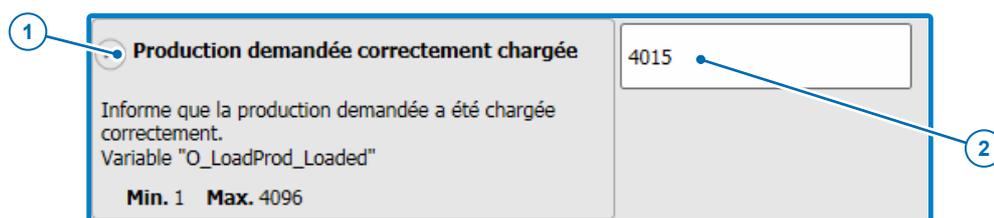
This parameter gives notification that the requested production was correctly loaded.

**Expert information**

The variable associated with this parameter is: O_LoadProd_Loaded.

Condition

- User group: Expert.

Procedure

- 1 "Requested production successfully loaded" parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4015.



If an unauthorized character is input, the border of the input field turns red.

8.6.5 “Ready for production change” parameter

Description

This parameter gives notification that the robot is ready for a production change.



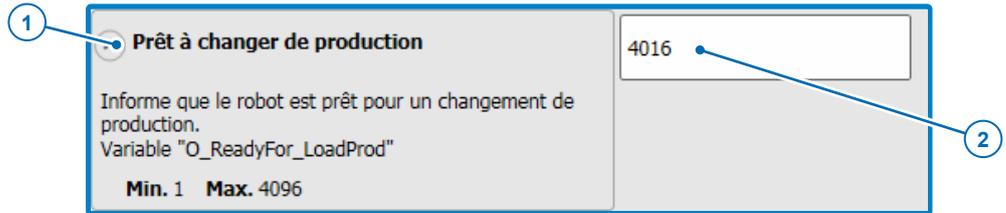
Expert information

The variable associated with this parameter is: O_ReadyFor_LoadProd.

Condition

- User group: Expert.

Procedure



- 1 “Ready for production change” parameter
- 2 Input field

- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4016.



If an unauthorized character is input, the border of the input field turns red.

8.6.6 “Loaded production number” parameter

Description

This parameter returns the number of the production that was loaded.



Signal on 8 bits at least (maximum 32 bits).



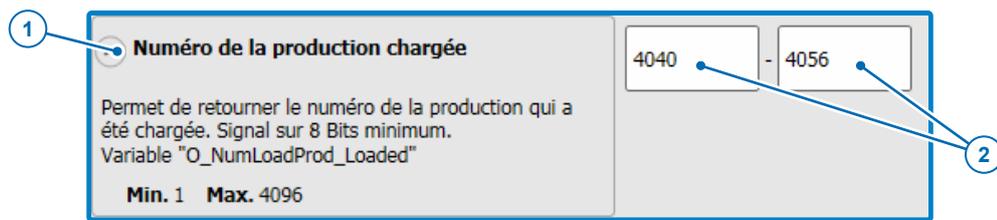
Expert information

The variable associated with this parameter is: O_NumLoadProd_Loaded.

Condition

- User group: Expert.

Procedure



- 1 "Loaded production number" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4040 - 4056.

If an unauthorized character is input, the border of the input field turns red.

8.6.7 "Production change error number" parameter

Description

This parameter returns the number of the error that occurred during the production change.

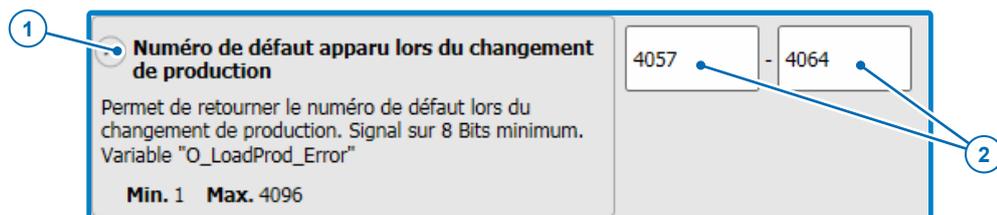
Signal on 8 bits at least (maximum 32 bits).

Expert information
The variable associated with this parameter is: O_LoadProd_Error.

Condition

- User group: Expert.

Procedure



- 1 "Production change error number" parameter
 - 2 Input field
- In the input field (2), enter a value for the variable.
 - Min. value: 1.
 - Max. value: 4096.
 - Default value: 4057 - 4064.

If an unauthorized character is input, the border of the input field turns red.

9 Configuration of messages

Description

WopCore makes it easy to generate trajectory and background task error messages. 100 trajectory messages and 100 messages for the background task can be configured.



Expert information

- ▶ Messages edited directly in the database

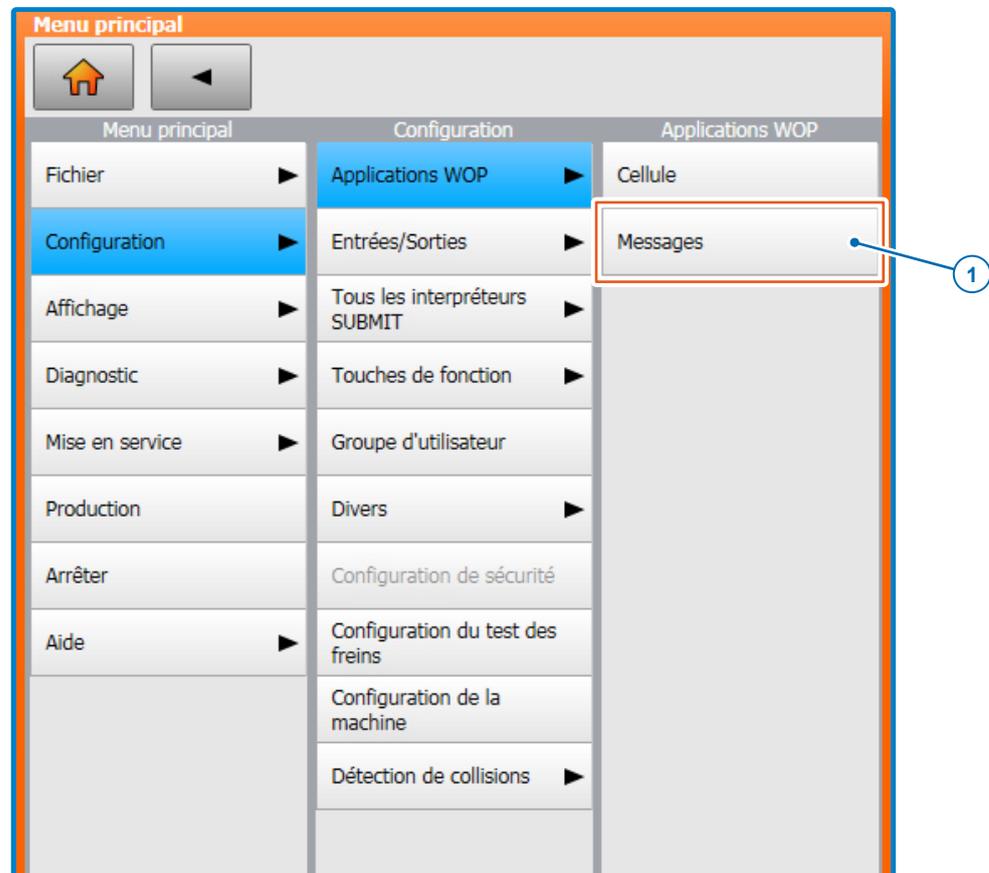
Messages can be edited directly in the file, WopUser.kxr (C:\KRC\TP\WopCore\Kxr).

Note: after the file has been modified, the rack must be restarted before using the message configuration screen or any online message display forms.

Condition

- User group: Expert.

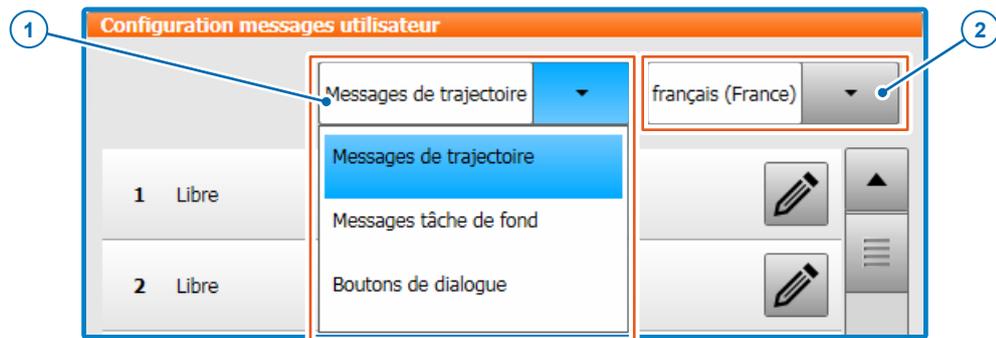
Procedure



- 1 "Messages" button

Select the following menu:

- Main menu > Configuration > WOP applications > Messages



- 1 Type of messages
- 2 Choice of language

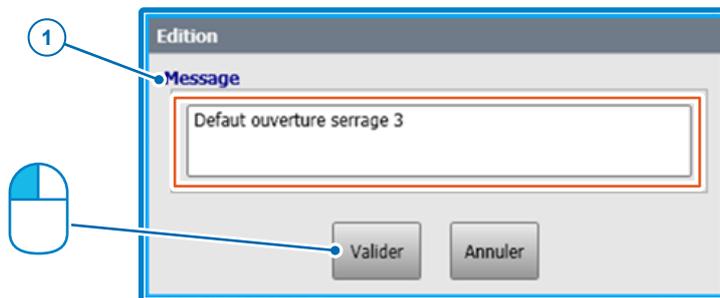
- Choose the type of message (1) you want to configure.
 - Trajectory messages (display during trajectory via an online form).
 - Background task messages (display by the background task in a configurable message on the screen).
 - Dialog buttons.
- Select language (2).



- 1 Configuration icon

- Click the icon (1) of the message you wish to configure.

Trajectory messages



- 1 Trajectory messages

- If you have chosen a trajectory message, you can enter the text of the message and click "confirm".

Background tasks messages

1 Background tasks messages

- If you have chosen a background task message, enter the text of the message and the equation that will trigger its display, and then click "confirm".
- The delay before it is displayed is equal to the time between the equation being set to TRUE and the display of the message.

10 User code

10.1 "WopUserDat" file: User data

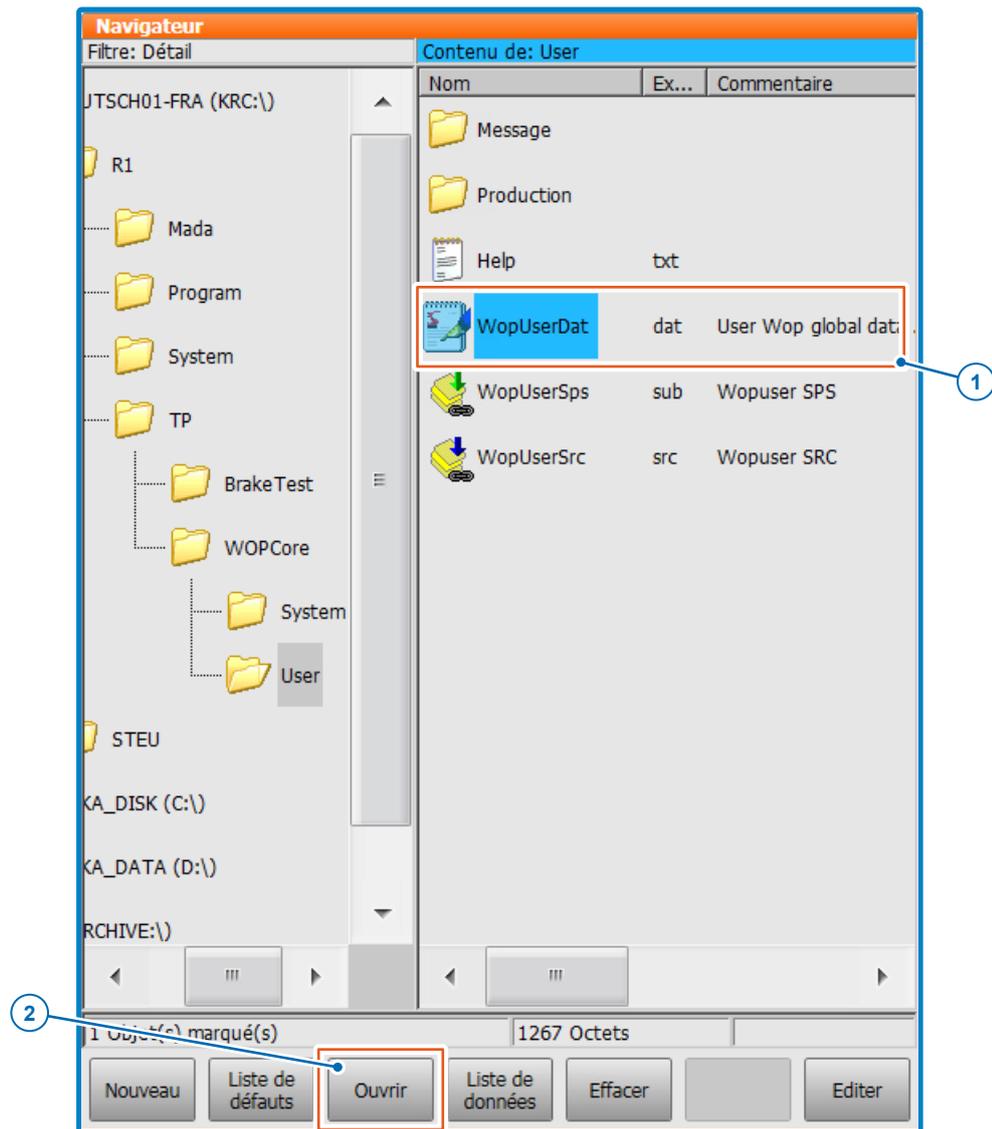
Description

In this file, the user is advised to declare user variables, inputs, outputs, and enumerations.

Condition

- User group: Expert.
- Mode T1.
- No program selected.

Procedure



- 1 "WopUserDat" file
- 2 "Open" button

Select the file "WopUserDat" in the project tree:

- R1 > TP > WOPCore > User > WopUserDat.
- Click the "Open" button to open the file.

```

4 Structure of robot work management
5
6
7 --- User Customer Declaration ---
8 ; !!! Please, use only global declaration in this
  ↓ dat file !!!
9 ;--- Inputs signals ---
10
11 ;--- Outputs signals ---
12
13 ;--- Variables ---
14
15
16

```

1 "WopUserDat" file

10.2 "WopUserSps" file: User program for the background task

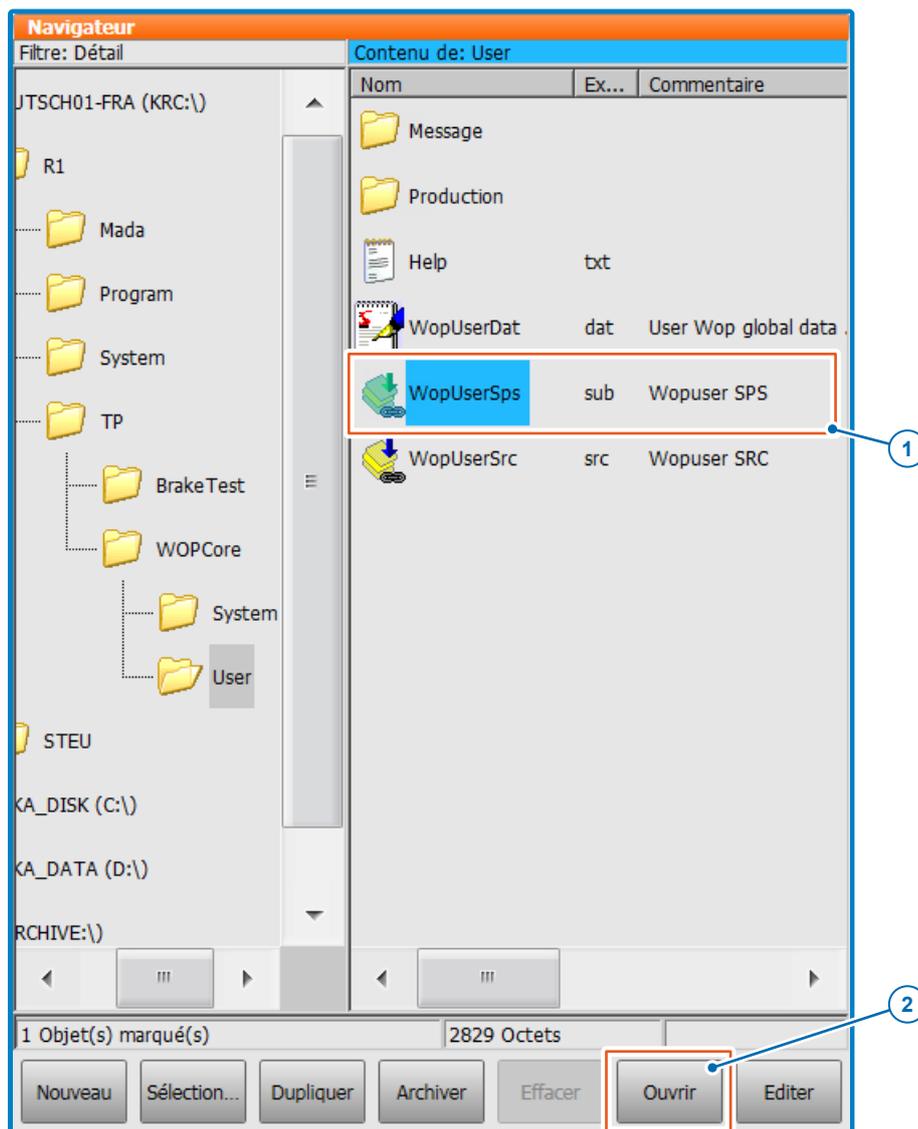
Description

This file is available to program the background task functionality of the client project.

Condition

- User group: Expert.
- Mode T1.
- Program aborted.

Procedure



- 1 "WopUserSps" file
- 2 "Open" button

Select the file "WopUserSps" in the project tree:

- R1 > TP > WOPCore > User > WopUserSps.
- Click the "Open" button to open the file.

The program contains the following folders:

- Fold Command Called when no program selected
Insert in this folder all instructions that must be executed when no program is selected.
- Fold Command Called when Master.src is in reset
Insert in this fold all instructions that must be executed when the program Master.src is reset.
- Fold Flag configuration
Used to adapt certain WOP variables and signals. See page 57.
- Fold Submit Task : Services calls condition
Used to manage service trajectories See page 65.
- Fold Submit Task User task init
Insert in this folder all instructions that must be executed on initialization of the background task.

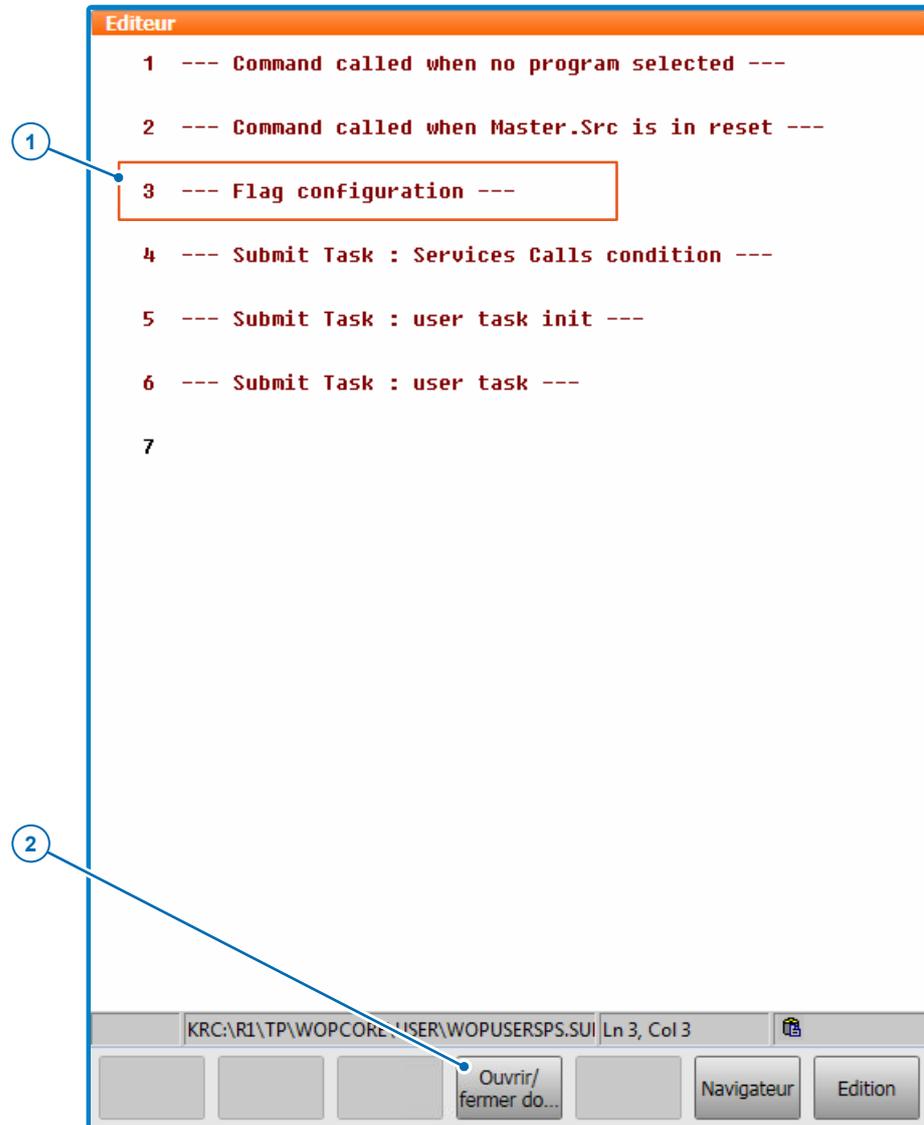
- Fold Submit Task : User task

Insert in this folder all the calls for subprograms and functions created for the background task.



Any creation of subprograms or functions in the background task linked to the customer process must be done in this file, after the last folder.
Remember to call them in the Submit Task fold: User Task.

10.2.1 Flag configuration



- 1 "Flag configuration" fold
 - 2 "Open/Close fold" button
- Click the fold "Flag configuration" and then press "Open/Close fold".

10.2.2 Robot in service

Description

O_Rob_RunInAuto informs an external system that the robot is in service.

'Robot in service' means the following robot state:

- No error on the program name.
- External mode enabled powered.
- MASTER.SRC program selected and running.
- No errors displayed.

- Base initialization complete.

You can add conditions to the equation issuing the O_Rob_RunInAuto signal to adapt the behavior of the robot according to your installation.

- Modify the following equation:
 - $\$OUT[O_Rob_RunInAuto]=X_Rob_RunInAuto \text{ AND condition1 AND Condition 2}$



Do not delete anything from the equation O_Rob_RunInAuto. This could affect the safety of WOP operation. You must ensure that the safety level of the equations is greater than the master equation.

10.2.3 Robot ready to start

Description

O_ReadyFor_Start informs an external system that the robot is ready to receive a start (Cycle start).

'Robot ready to start' means the following robot state:

- No error on the program name.
- Global release of movements (\$MOVE_ENABLE) is enabled.
- Robot on the trajectory.
- External mode enabled.
- MASTER.SRC program selected.
- No errors displayed.

You can add conditions to the equation issuing the O_ReadyFor_Start signal to adapt the behavior of the robot according to your installation.

- $\$OUT[O_ReadyFor_Start]= M_ReadyFor_Start \text{ AND condition1 AND Condition 2}$



Do not delete anything from the equation O_ReadyFor_Start. This could affect the safety of WOP operation. You must ensure that the safety level of the equations is greater than the master equation.

10.2.4 Robot in manual

Description

O_Rob_Manu informs an external system that the robot is in manual mode.

'Robot in manual' means the robot state with mode T1 or T2 enabled.

You can add conditions to the equation issuing the O_Rob_Manu signal to adapt the behavior of the robot according to your installation.

- $\$OUT[O_Rob_Manu]=X_Rob_InManu \text{ AND condition1 AND Condition 2}$



Do not delete anything from the equation O_Rob_InManu. This could affect the safety of WOP operation. You must ensure that the safety level of the equations is greater than the master equation.

10.2.5 Robot in parking position

Description

O_Rob_Parking informs an external system that the robot is in parking position.

During robot operation, the operator presses the Parking request button (pulse).

As soon as possible, the robot processes this request and positions itself at the Parking point.

The service request function is already programmed in WopCore.

'Robot in parking position' means the following robot state:

- Robot physically in the parking position \$IN_HOME5.

You can add conditions to the equation issuing the O_Rob_Parking signal to adapt the behavior of the robot according to your installation.

- \$OUT[O_Rob_Parking]=X_Rob_Parking AND condition1 AND Condition 2



Do not delete anything from the equation O_Rob_Parking. This could affect the safety of WOP operation. You must ensure that the safety level of the equations is greater than the master equation.

10.2.6 Robot in loop position

Description

O_Rob_LoopPos1 informs an external system that the robot is in loop position.

'Robot in loop position' means the following robot state:

- Robot physically in the loop position \$IN_HOME1.

You can add conditions to the equation issuing the O_Rob_Loop signal to adapt the behavior of the robot according to your installation.

- \$OUT[O_Rob_LoopPos1]=X_Rob_Loop AND condition1 AND Condition 2



Do not delete anything from the equation O_Rob_LoopPos1. This could affect the safety of WOP operation. You must ensure that the safety level of the equations is greater than the master equation.

10.2.7 Robot in Cycle

Description

O_Rob_InCycle informs an external system that the robot is in Cycle.

'Robot in cycle' means the following robot state:

- Robot in automatic and in trajectory confirmed by a cycle code equation.

You can add conditions to the equation issuing the O_Rob_InCycle signal to adapt the behavior of the robot according to your installation.

- \$OUT[O_Rob_InCycle]=X_Rob_InCycle AND condition1 AND Condition 2



Do not delete anything from the equation O_Rob_InCycle. This could affect the safety of WOP operation. You must ensure that the safety level of the equations is greater than the master equation.

10.2.8 Robot error

Description

O_GeneralError_Robot informs an external system that the robot is in an error state.

'Robot error' means the following robot state:

- Presence of a system error message or robot stopped.

You can add conditions to the equation issuing the O_GeneralError_Robot signal to adapt the behavior of the robot according to your installation.

- \$OUT[O_GeneralError_Robot]=\$STOPMESSAND condition1 AND Condition 2



Do not delete anything from the equation O_GeneralError_Robot. This could affect the safety of WOP operation. You must ensure that the safety level of the equations is greater than the master equation.

10.2.9 Zone access authorization

Description

O_Acces_Allowed informs an external system that the robot authorises access to the protected zone.

You can add conditions to the equation issuing the O_Acces_Allowed signal to adapt the behavior of the robot according to your installation.

- $\$OUT[O_Acces_Allowed]=(\$T1 \text{ OR } \$T2 \text{ OR NOT } \$ALARM_STOP \text{ OR } (M_StepSfc_AccessArea==20))\text{AND condition1 AND Condition 2}$



Do not delete anything from the equation O_Acces_Allowed.
This could affect the safety of WOP operation.

You must ensure that the safety level of the equations is greater than the master equation.

10.2.10 Authorization to execute service trajectories

Description

The variable X_Allow_ParkingService is used to authorize the execution of service trajectories (Parking, maintenance, access).

You can personalize this authorization by adding conditions.

By default: X_Allow_ParkingService=TRUE

10.2.11 Robot movement authorization

Description

The variable M_MoveRobot_User is used to stop the robot during the trajectory.

You can personalize this authorization by adding conditions.

By default, M_MoveRobot_User=TRUE

10.2.12 Management of the parking request

Description

The variable X_UserParking_Request is used to make one parking request per equation in addition to the dedicated input.

You can personalize this authorization by adding conditions.

By default: X_UserParking_Request=X_Access_Request

10.2.13 Management of the master program reset request

Description

The variable X_MasterReset_User is used to make a reset request for the program Master.src.

You can personalize this authorization by adding conditions.

By default, X_MasterReset_User=FALSE

10.2.14 Management of Escape request

Description

The variable X_UserEscape_Request is used to make the escape request by executing the program W_Escape.SRC. Execution of the W_Escape program has priority over the work and service paths.

X_UserEscape_Request=FALSE

10.2.15 Management of the start authorization

Description

The variable M_ReadyForStart_User is used to authorize external automatic robot startup.

You can personalize this authorization by adding conditions.

By default, M_ReadyForStart_User=TRUE

10.2.16 Management of the OK start position

Description

The variable X_PosRobot_OK indicates that the robot is in a position corresponding to the loop or parking position. If the variable is set to FALSE when the master program is launched, the robot displays a message and executes the trajectory W_CheckHome.src.

You can personalize this authorization by adding conditions.

By default, X_PosRobot_OK=\$IN_HOME5 OR \$IN_HOME1

10.2.17 Robot cycle start request

Description

A True pulse on the variable X_UserStartCycle_Request starts the robot cycle. As a PLC input, its state must return to False in order to restart again the robot cycle.

Association to a GUI

This command can be associated with a button on a GUI such as Kuka.HmiEasy



The cycle start request is only useful if the robot's autostart mode is set to "Ext SFC".

10.2.18 Robot stop request at the end of the cycle

Description

The X_UserEndCycle_Request variable is used to request to stop the robot cycle at the end of the current trajectory.

Association to a GUI

This command can be associated with a button on a GUI such as Kuka.HmiEasy

10.2.19 Request for immediate robot stop

Description

The X_UserStopCycle_Request variable is used to request an immediate stop of the robot cycle.

Association to a GUI

This command can be associated with a button on a GUI such as Kuka.HmiEasy

10.2.20 Door opening and zone access request

Description

The variable X_UserAccess_Request allows you to request a cell opening during the automatic cycle.

Association to a GUI

This command can be associated with a button on a GUI such as Kuka.HmiEasy

10.3 “WopUserSrc” file: User code called during trajectory

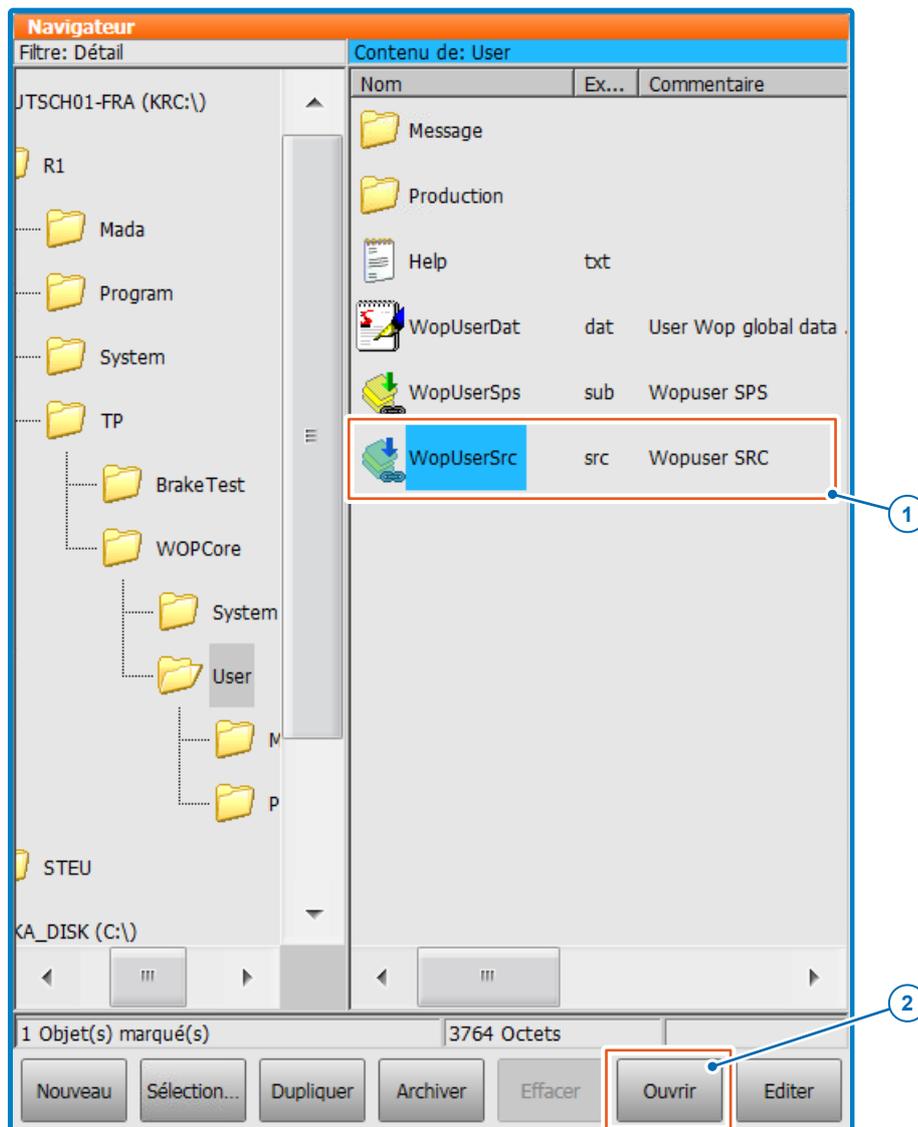
Description

This file is available to program the robot movement task functionality of the client project.

Condition

- User group: Expert.
- Mode T1.
- Program aborted.

Procedure



- 1 "WopUserSrc" file
- 2 "Open" button

Select the file “WopUserSrc” in the project tree:

- R1 > TP > WOPCore > User > WopUserSrc.
- Click the “Open” button to open the file in Edit mode.

The program contains the following folders:

- Fold Command Called before check position

Insert in this fold all instructions that must be executed on initialization of the Master program before 'check home position'.

- Fold Command Called after check position

Insert in this fold all instructions that must be executed on initialization of the Master program after 'check home position'.

- Fold Command Called at the beginning of trajectory

Insert in this folder all instructions that must be executed on initialization of the trajectories, even when they are selected manually (when learning the points, for example).

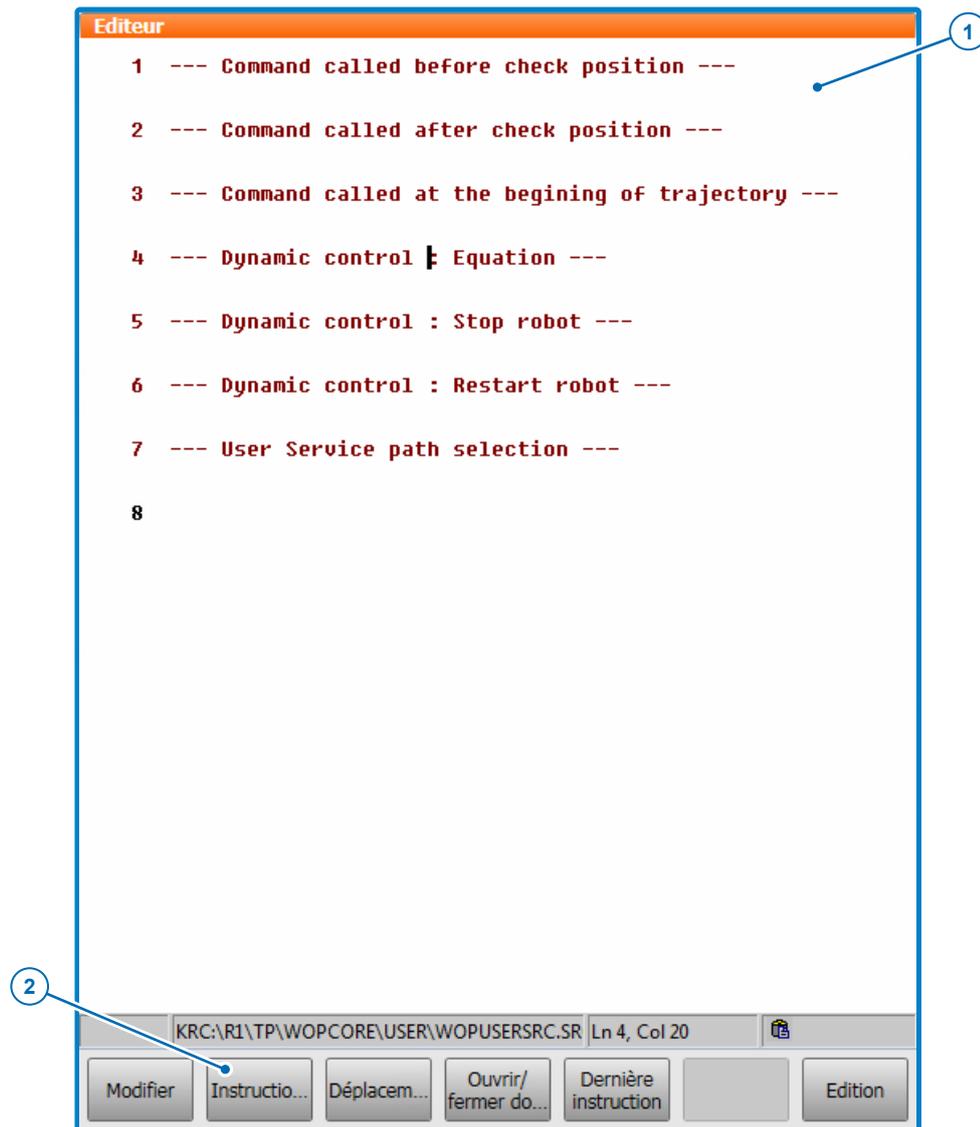
- Folds Dynamic control

These folds are used to manage dynamic control.

- Fold User Service Path Selection lets you add service trajectories. See page 65.



Any creation of SRC subprograms or SRC functions linked to the customer process must be done in this file, after the last fold.



- 1 "WopUserSrc" file
- 2 "Instruction" button

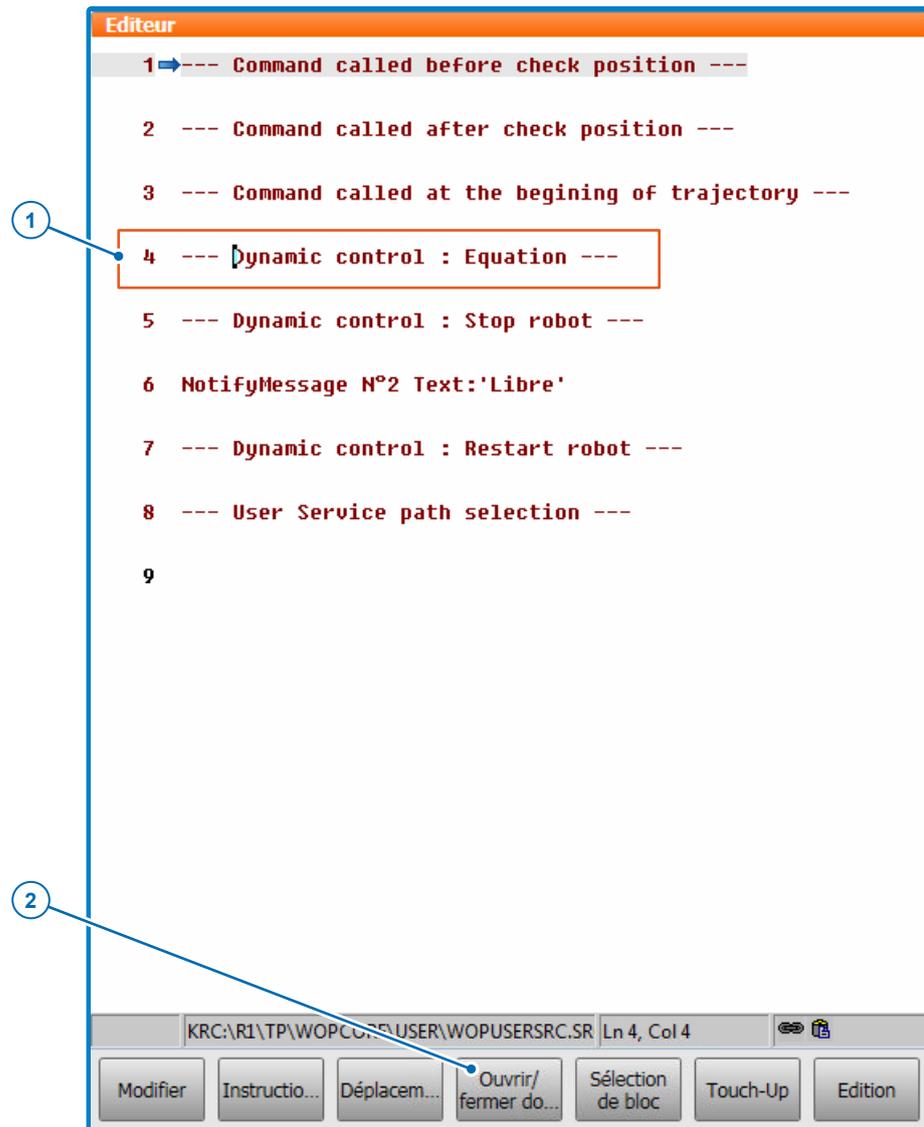
10.3.1 Configure dynamic control

Description

Dynamic control is used to stop the robot if an element of the process develops a fault; for example, when a part is lost during transportation from a point A to a point B.

To generate a robot stop, WopCore has a preprogrammed function. This function must be completed by the integrator according to the following method.

Procedure



- 1 "Dynamic Control : Equation" fold
- 2 "Open/Close fold" button

- Open the file "WopUserSrc.src".
- Press "Open/Close fold" to open the "Dynamic Control: Equation" fold.
- Modify the following equation by adding the conditions for the actuators to be continuously monitored:
 - \$CYCFLAG[C_CycFLagUser]=FALSE
 - For example : \$CYCFLAG[C_CycFLagUser]=NOT I_MachinePrete



The \$CYCFLAG\[C_CycFLagUser] must always be set to the state FALSE.
Dynamic control is triggered when \$CYC-FLAG\[C_CycFLagUser] is set to the state TRUE.

- This equation is monitored by a higher priority interrupt that calls on an interrupt subprogram that generates the robot stop.
- Stop actions can be configured in the "Dynamic control: Stop robot" fold.
 - In this fold, insert all the conditions that should stop the robot and prevent it from restarting its cycle.
 - Restart actions can be configured in the "Dynamic Control: Restartrobot" fold.
 - In this fold, insert the actions that must be restarted with the robot.

11 Generating SRC type error messages

Description

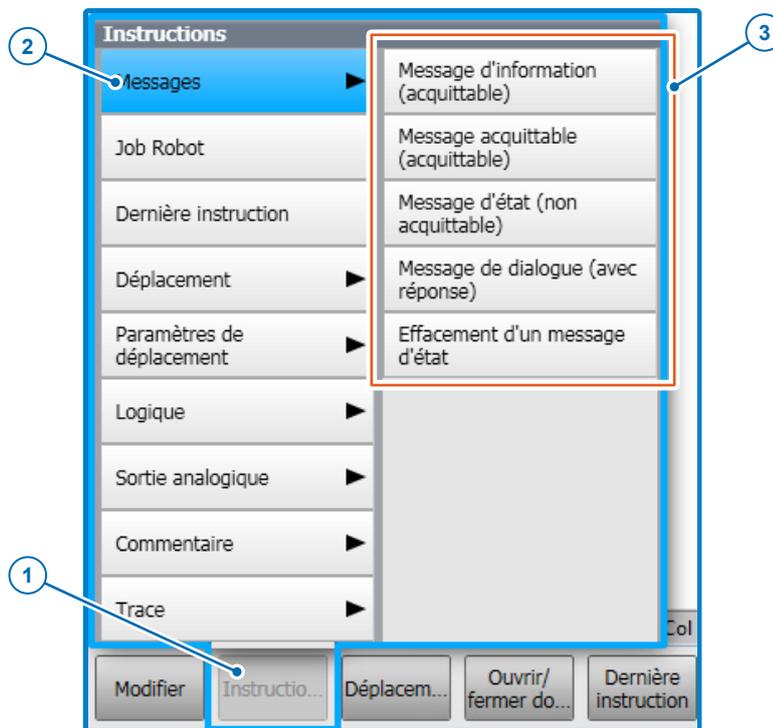
WopCore lets you easily generate error messages for the robot movement task.

These messages are generated by the robot process, i.e. by the trajectories and any other program with the extension .SRC.

Messages can generate:

- A simple warning message.
- A "STOP" message that halts the program and waits for user acknowledgement before resuming execution.
- A dialog message that stops the robot and displays a configurable multiple-choice list.

Procedure



- 1 "Instruction" button
- 2 "Messages" menu
- 3 Type of messages

- Press the "Instruction" button.
- Select the "Messages" menu.
- Select the type of message.
 - Information message (acknowledgeable).
 - Acknowledgeable message (acknowledgeable).
 - State message (not acknowledgeable).
 - Dialog message (with answer).
 - Deletes a state message.

Information message (acknowledgeable)

- 1 Information message
- 2 Parameters

- Use the fields to input the choices that appear when the message is displayed.
 - N°: number of message.
 - Language: message language.
 - Text: message text.
 - Parameters: parameters associated with the message.
- Press the "INSTR OK" button to confirm the form, and then click "OUI" ["YES"] to save.

Acknowledgeable message

- 1 Acknowledgeable message
- 2 Parameters

- Use the fields to input the choices that appear when the message is displayed.
 - N°: number of message.
 - Language: message language.
 - Text: message text.
 - Parameters: parameters associated with the message.
- Press the "INSTR OK" button to confirm the form, and then click "OUI" ["YES"] to save.

State message (not acknowledgeable)

- 1 Acknowledgeable message
- 2 Parameters

- Use the fields to input the choices that appear when the message is displayed.
 - N°: number of message.
 - Language: message language.
 - Text: message text.
 - Parameters: parameters associated with the message.
- Press the "INSTR OK" button to confirm the form, and then click "OUI" ["YES"] to save.

Dialog message (with answer)

- 1 Acknowledgeable message
- 2 Parameters

- Use the fields to input the choices that appear when the message is displayed.
 - N°: number of message.
 - Language: message language.
 - Text: message text.

- Answers: answer text.
- Parameters: parameters associated with the message.
- Press the "INSTR OK" button to confirm the form, and then click "OUI" ["YES"] to save.

Deletes a state message



- 1 Acknowledgeable message
- 2 Parameters

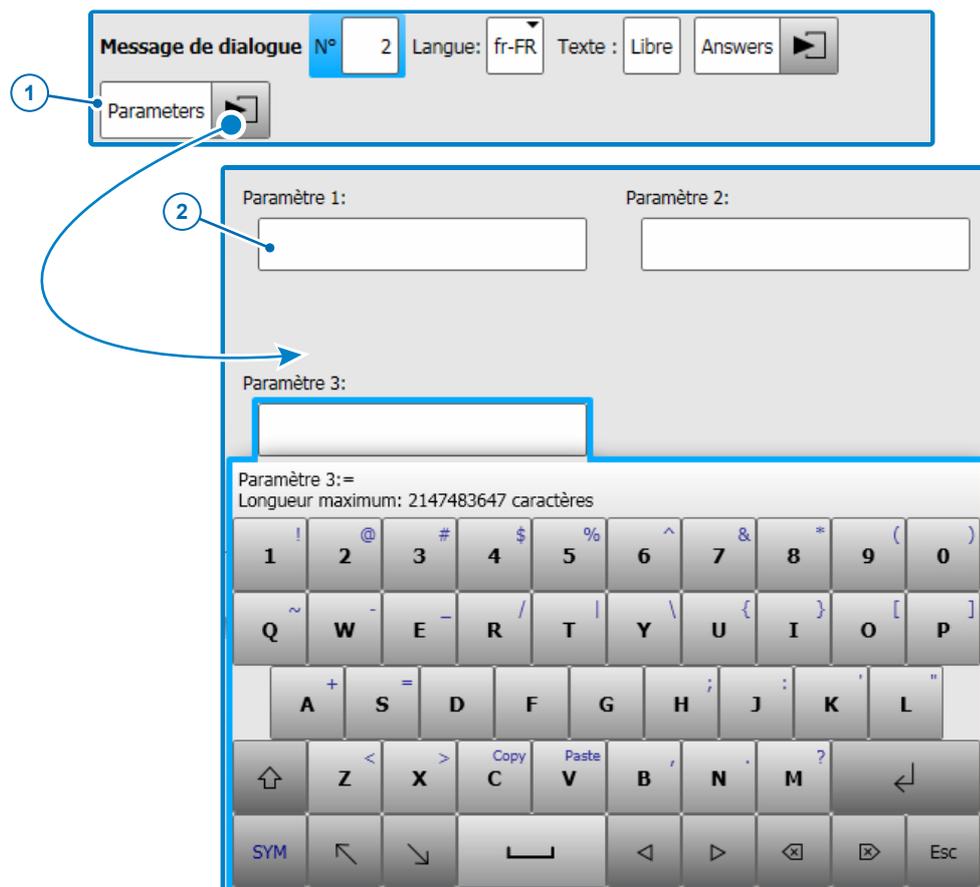
- From the list, select the number of the message to be displayed.
- Press the "INSTR OK" button to confirm the form, and then click "OUI" ["YES"] to save.

11.1 Displaying values in the messages

Description

WopCore can display variable values in movement task error messages.

Procedure



- 1 "Parameters" button
 - 2 "Parameters" window
- Click the "Parameters" button.
 - Specify up to 3 parameters to be displayed in a message. These parameters are variables whose result will be inserted into the message to be displayed.
 - Press "Instr OK" to confirm the form.

11.2 Program the processing of answers to the dialog messages

Description

WopCore lets you program the processing of answers to the dialog messages.

When the error message instruction generates a message with response keys, each response key is processed via the "M_Answer_DialogMessage" variable that contains the number of the key pressed.

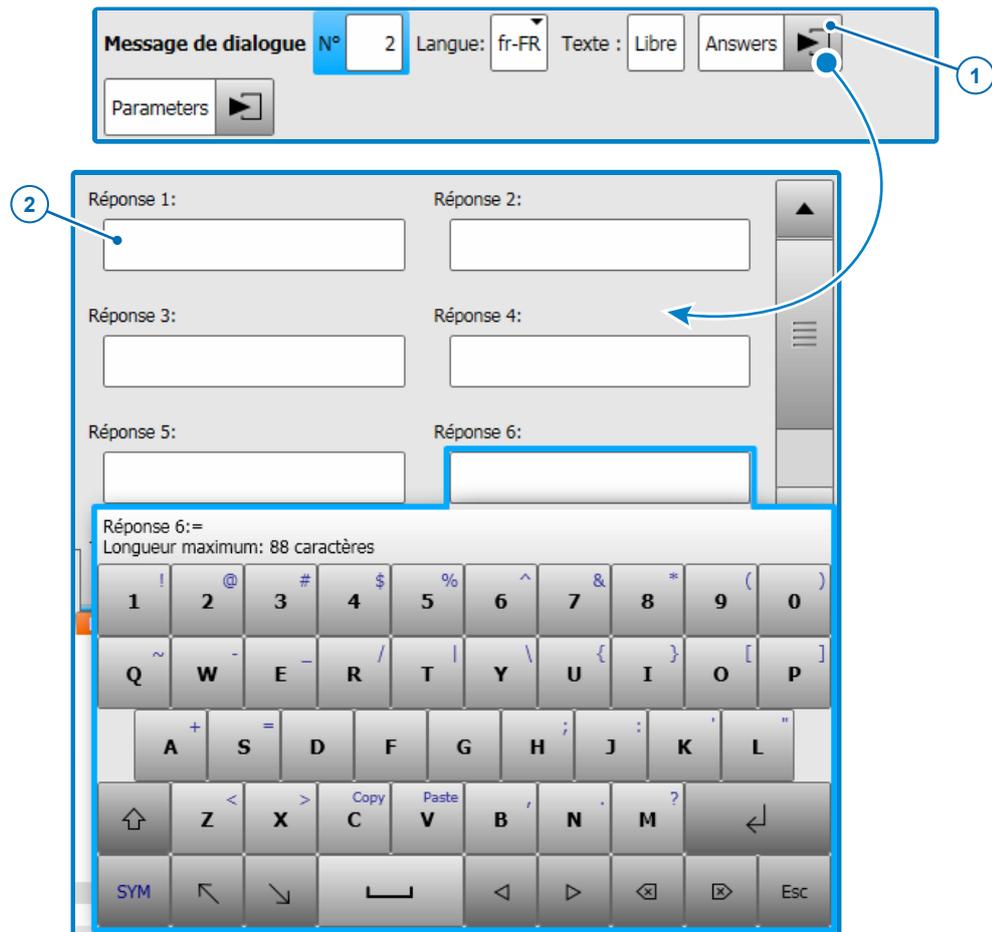
The integrator must program the action for each key according to a specific program structure.

A maximum of seven keys can be programmed.

Condition

- User group: Expert.

Procedure



- 1 "Answer" button
- 2 "Answer" window

- Click the "Parameters" button.
- Specify up to seven answers.
- Press "Instr OK" to confirm the form.

In the program

- For each CASE corresponding to a programmed key, insert the key processing program.

12 Service trajectories

12.1 Define the points LoopPos, Parking



Definition of service position

Warning: The parking position must be in a location where there is no risk of collision. The parking position must not allow any person to be trapped, crushed, or otherwise endangered during a manual movement or the execution of the program.

Description

The points LoopPos and Parking are declared in the application, but their coordinates do not match the site requirements.

That is why the integrator must define these points before launching the program.

To avoid any collision, these points are stored in the system so that the position cannot be modified unexpectedly.

- PT_LoopPos1: loop point of all trajectories.
- PT_ParkingPos1: parking point.

Condition

- Mode T1.

Procedure

- Click the "Program" folder.
- Click the "Service paths" folder.
- Select the file "W_LoopPos.src".
 - Place the cursor on the line of the point:
`PTP PT_LoopPos1 Vel= 100% DEFAULT`
 - Set the robot to "manual control" on the physical position of the point to be reached.
 - Press "TOUCH-UP".
 - Confirm the position by pressing "YES".
- Click the file "W_Parking.src".
 - Place the cursor on the line of the point:
`PTP PT_ParkingPos1 Vel= 100% DEFAULT`
 - Move the robot to "manual control" on the physical position of the point to be reached.
 - Press "TOUCH-UP".
 - Confirm the position by pressing "YES".

12.2 Programming a service trajectory

Description

WopCore can easily create up to five service trajectories.

Condition

- User group: Expert.

Procedure

- Create a trajectory in the Service Paths folder. Example W_Cleaning.
- Open the file "WopUserSps.sub".
- Press "OPEN/CLOSE FOLD" to open the "SubmitTask: Services Calls condition" fold.
- Modify the equation of the service trajectory concerned: X_UserService_Request [x]=FALSE
 - For example: X_UserService_Request [1]=I_CleaningPosition_Request
- Open the file "WopUserSrc.src".
- Press "OPEN/CLOSE FOLD" to open the "User Service pathselection" fold.
- Press "OUVRIR/FERMER FOLD" ["OPEN/CLOSE FOLDER"] to open the service trajectory folder.
 - Example :
 - ▶ FOLD --- Trajectory called with Service equation 1 ---
 - ▶ Write here the name of the Trajectory called with Service equation 1
 - ▶ E.g.: W_Service1 ()

12.3 Programming an escape

Description

WopCore has a program W_Escape.SRC, which manages an escape in the event of a problem.

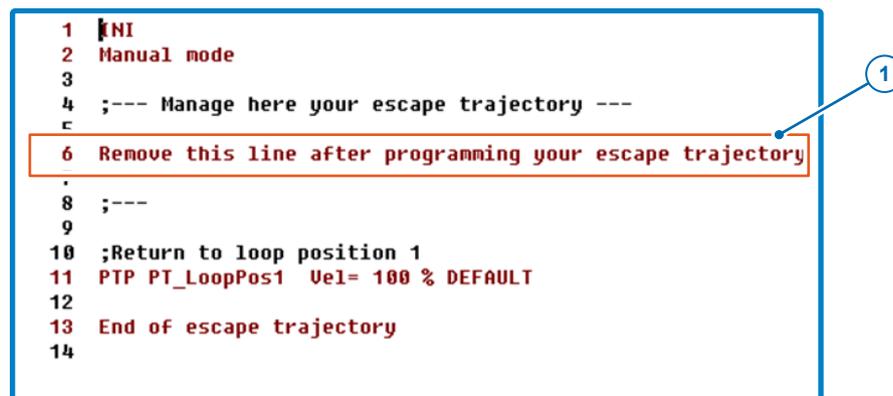
Management of this escape is directly linked to the work structure in the locking of the equations: X_CurrentWork_Robot.

The various states of this structure are used to identify the robot's state.

```

1  #NI
2  Manual mode
3
4  ;--- Manage here your escape trajectory ---
5
6  Remove this line after programming your escape trajectory
7
8  ;---
9
10 ;Return to loop position 1
11 PTP PT_LoopPos1 Ue1= 100 % DEFAULT
12
13 End of escape trajectory
14

```



1 Safety loop

- The safety loop (1) is used to not execute the escape program. Delete this loop when the configuration is completely finished.

The escape trajectory is W_Escape.SRC executed by the Master.src file.

The Escape request has the effect of executing an exit from the current path program at the position where the robot is located and returning to the Master.src

The Master.src will then execute the file W_Escape.SRC in which the escape strategy is configured. The execution of the W_Escape program has priority over the work and service cycles

**WARNING**

To avoid an escape when the configuration is not complete, a safety loop is present at the start of the W_Escape.SRC program.

Take care to delete this loop only when you are sure that the execution of an escape takes place without incident.

Condition

- User group: Expert

Procedure

- Open the file KRC>R1>TP>WopCore>WopUserSps.SUB

Insert a condition for the variable: X_UserEscape_Request

Example:

X_UserEscape_Request=Condition1 And Condition2

**WARNING**

The escape is called on the rising edge of the request.

13 Productions

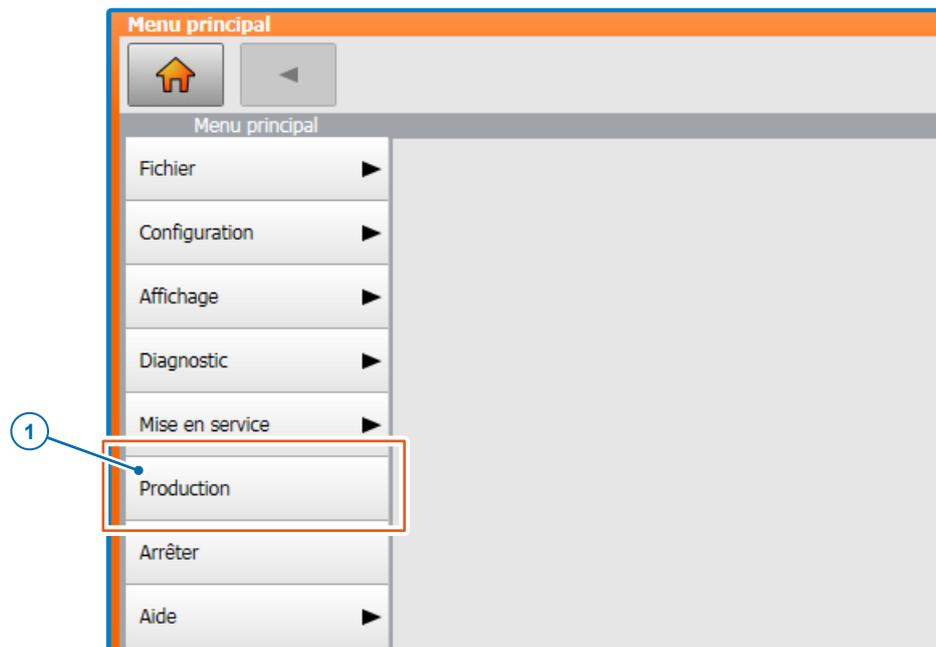
13.1 Creating a production

Condition

- Mode T1 and Expert Mode.
- Multiple production mode selected.

 Before generating or duplicating productions, check that there are enough bits configured in the "Production Management" menu.

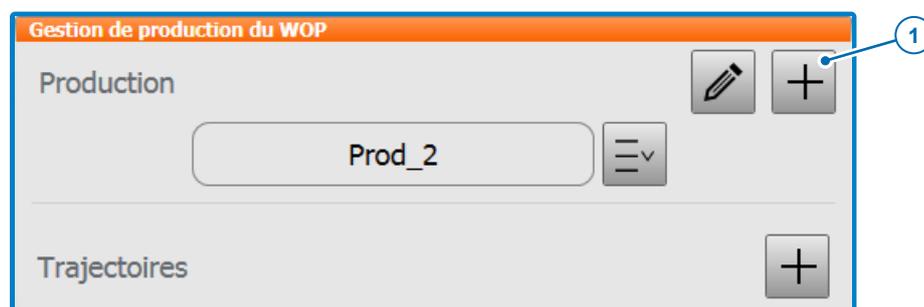
Procedure



1 "Production" menu

Select the following menu:

- Main menu > Production
- A screen opens.



1 Icon (+)

- Click the icon (1) to create a production.

A window opens.

Nouvelle production

Nom :

Code automate :

Production vierge
 Dupliquer la production

- 1 Name of the new production
- 2 "New production" button

- Enter the name of the production to be created.
- Click Confirm.
- The production is created and appears in the list.

Duplicate an existing production

Nouvelle production

Nom :

Code automate :

Production vierge
 Dupliquer la production

Copie des conditions
 Copie des trajectoires

- 1 Name of the new production
- 2 "Duplicate production" button

To create a new production by duplicating an existing production, click the "Duplicate production" button.

- Enter the name of the production to be created.

- From the list, choose the name of the production to be duplicated.
- Select the options “Copy conditions” and “Copy trajectories” if necessary.
- Click Confirm.
- The production is created and appears in the list.

13.2 Creation of trajectory calls

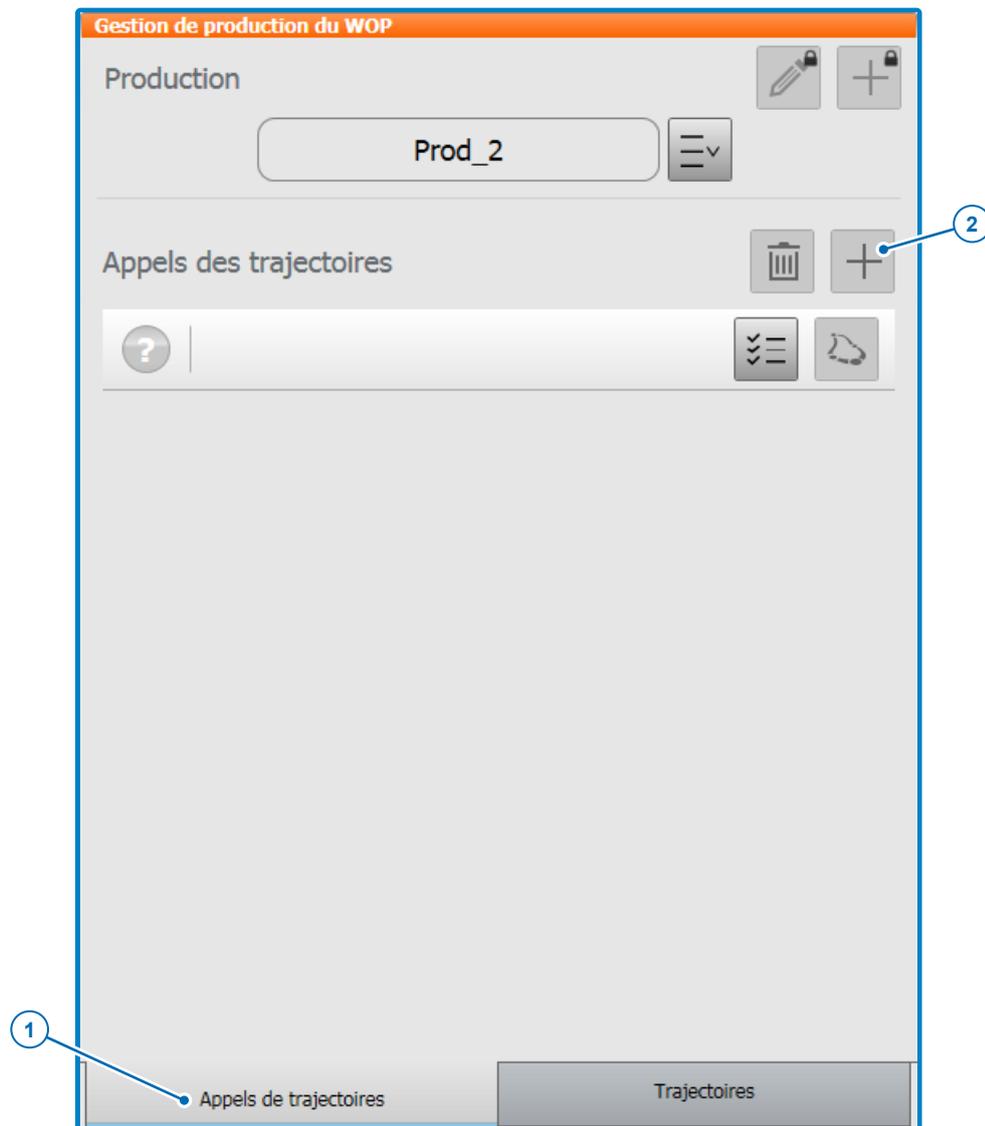
Description

A trajectory call defines the call conditions for each trajectory.

Condition

- Mode T1 and Expert Mode.

Procedure



- 1 "Trajectory calls" tab
 - 2 Icon (+)
- Select the "Trajectory calls" tab
 - Click the icon (1). A trajectory call is added.

13.3 Creating a trajectory

Description

The work trajectory is a trajectory that allows the robot to perform a specific job.



Expert information

- ▶ Adding trajectories without the production screen

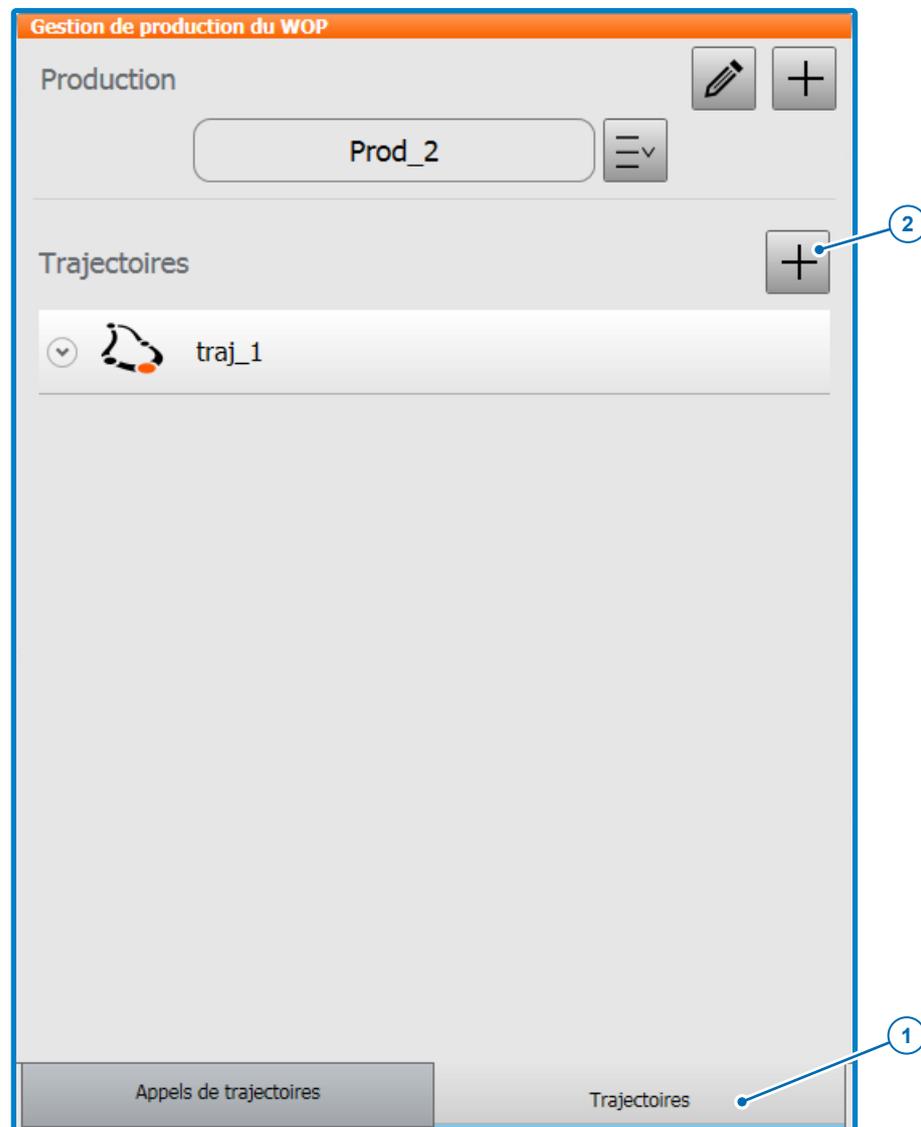
It is possible to add trajectories to the WorkingPath folder without using the production screen. These will be integrated automatically when the production screen is opened.

Warning: The trajectories must not be in a sub-directory of the WorkingPath folder.

Condition

- Mode T1 and Expert Mode.
- Trajectory call created.

Procedure



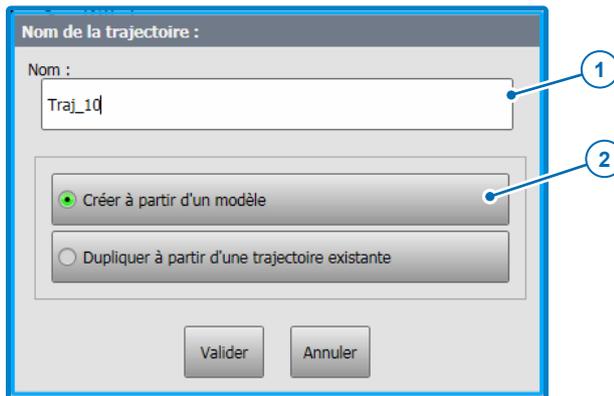
- 1 "Trajectory" tab
- 2 Icon (+)

After creating one or more trajectory calls:

- After creating one or more trajectory calls, select the "Trajectories" tab.

- Click the icon (1).

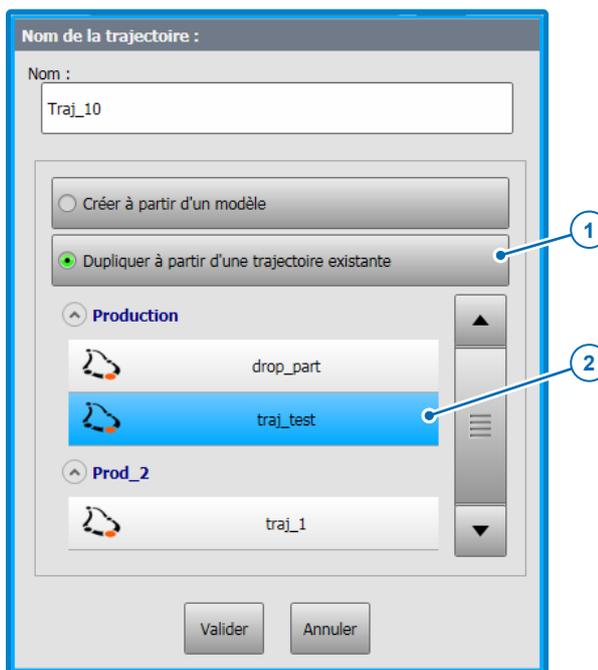
A window opens, allowing you to create a trajectory:



- 1 Name of the new trajectory
- 2 “Create from a template” button

- Enter the name of the trajectory to be created.
- Click Confirm.
- The trajectory is created and appears in the list.

Duplicate an existing trajectory



- 1 Name of the new trajectory
- 2 “Duplicate based on an existing trajectory” button

To create a new trajectory by duplicating an existing trajectory, click the “Duplicate based on an existing trajectory” button.

- Enter the name of the trajectory to be created.
- From the list, choose the name of the trajectory to be duplicated.
- Click Confirm.

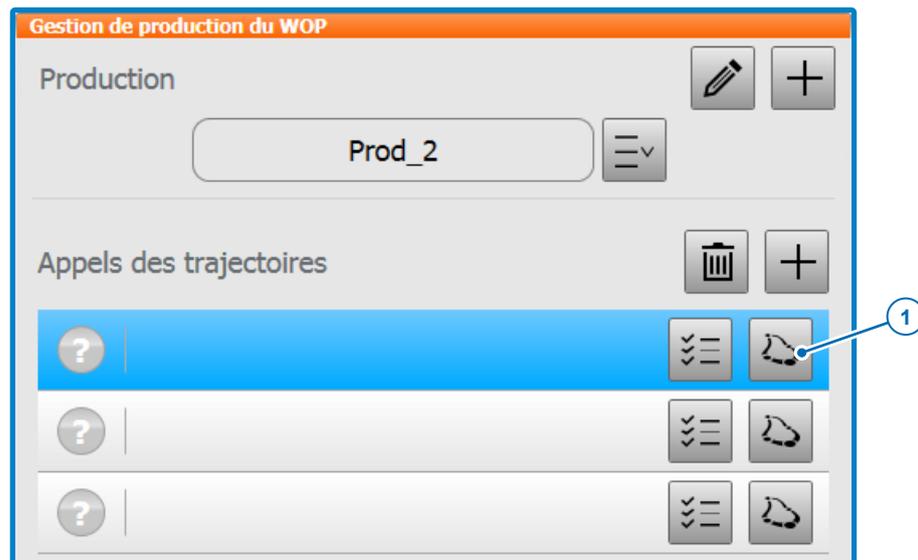
- The trajectory is created and appears in the list.

13.4 Associate a trajectory call with a trajectory

Condition

- Mode T1 and Expert Mode.
- Trajectory call created.
- Trajectory created.

Procedure



1 Trajectory selection icon

- When you are in the production screen and one or more trajectory calls have been created, click the icon (1).
- A window opens, allowing you to choose a trajectory.



1 Trajectory selection

- Click the desired trajectory to highlight it in blue.
- Click Select. The trajectory has been associated with the call created earlier.



- The trajectory has been created in the robot's directory "R1\Program\WorkingPaths". You can now select this trajectory to insert the necessary points and instructions.

13.5 Defining the trajectory call conditions

Description

The program MASTER.SRC is used to direct the trajectories according to the call conditions.

Each call condition must be unique and associated with only one trajectory.



Expert information

► Editing equations in WorkVisual

The cycle code equations can be edited directly in the file WopUserCycleEquation.src (R1/TP/WopCore/User/Production).

► Adding a trajectory call to WorkVisual

To add a new call directly in the file WopUserCycleEquation.src, simply copy the six lines of an existing call, increment the index of the table, and initialize the cycle code to 0.

For example:

```
X_UserCycle_WOP[2].Equation1 = Equation 1
X_UserCycle_WOP[2].Equation2 = Equation 2
X_UserCycle_WOP[2].Equation3 = Equation 3
X_UserCycle_WOP[2].GlobalEquation = Equation globale
X_UserCycle_WOP[2].CycleCode =0
X_UserCycle_WOP[2].PathName[]=" "
```

Condition

- Mode T1 and Expert Mode.

Procedure



- 1 Call conditions definition icon

- When you are in the production screen and one or more trajectory calls have been created and their trajectories associated, click the icon (1).
- A window opens, allowing you to enter the trajectory call conditions:

The screenshot shows a software interface titled "Edition des équations". It contains three text input fields labeled "Condition 1", "Condition 2", and "Condition 3". The first field contains the text "(\$ov_pro==20)", the second contains "FALSE", and the third contains "FALSE". Below these is a larger text input field labeled "Equation principale" containing "Condition1". At the bottom of the interface are two buttons: "Valider" and "Annuler". Two callout boxes with numbers 1 and 2 point to the condition fields and the main equation field respectively.

- 1 Conditions text field
- 2 Main equation text field

In the text fields "Condition 1", "Condition 2", and "Condition 3", the trajectory call partial conditions must be entered. It is not necessary to enter the three conditions.

In the "Main equation" text field, you can combine conditions 1, 2 and 3 with AND and OR operators.

The "ConditionX" buttons provide an easy way to insert the references to the conditions defined above.

- Configure the various equations.
 - If the equation entered is correct and the result is True, the box around the text field will be green.
 - If the equation entered is correct but its result is False, the box around the text field will be red.
 - If the equation is not correct (syntax error), the text will be red and the box around the text field will be grey.
- Click Confirm.

A trigger condition is now linked to the trajectory.

13.6 Locking the cycle code equations

Description

The WOP lets you manage a complex equation and lock the cycle code equations by an efficient method described in detail below.

The user must be very meticulous about locking these equations.

The variable X_CurrentWork_Robot lets you track the robot's operation and the locking of the cycle code equations.

- WorkInteger contains the current work value (optional).
- WorkEnum contains the current work type (EnumWorkRobot variable enumeration).
- StateInteger contains the current work state value (optional).
- StateEnum contains the current work state type (EnumStateRobot variable enumeration).

Condition

- User group: Expert.

Procedure

(See section 10.1, page 46).

- See section 8.1.1, page 17.
- Create types for each job configured. This enumeration is used to represent a job or a state with an understandable variable name.

Example of a work declaration:

- GLOBAL ENUM EnumWorkRobot Nothing, PickUpPart1, DropOffPart1

Example of a work state declaration:

- GLOBAL ENUM EnumStateRobot Nothing, PickUpPartOK, DropOffPartOK

Example of a trajectory insertion declaration:

- The robot is being asked to pick up a part from one machine and drop it in another. We therefore have a part pick-up trajectory and a part drop-off trajectory.

13.7 Inserting the Job robot form in trajectories

Description

The Job Robot inline form allows you to insert the robot work and status structure into the paths.

Condition

- User group : Expert

Procedure

The Job Robot inline form Menu Instruction>Job Robot allows you to insert the robot work and status structure into the paths.

The screenshot shows the 'Job Robot' form with the following fields: 'Sur Trigger' (unchecked checkbox), 'Label de travail:' (dropdown menu showing 'Pick_Part'), 'N° de travail:' (input field with '0'), 'Label d'état:' (dropdown menu showing 'InProgress'), and 'N° d'état:' (input field with '0').

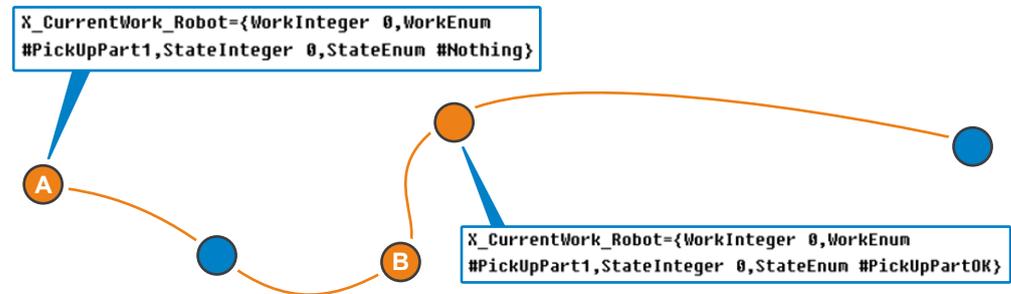
- On Trigger allows to execute the instruction at the beginning or at the end of a point, or over a time (-5000 to 5000 ms).
- When the Trigger option is checked, the settings:Distance and time appear.

The screenshot shows the 'Job Robot' form with 'Sur Trigger' checked. The 'Label de travail:' dropdown is 'Pick_Part', 'N° de travail:' is '0', 'Label d'état:' is 'InProgress', and 'N° d'état:' is '0'. The 'Distance:' dropdown is 'Début' and 'Temps:' is '0'. Below the form, a blue button labeled 'Début' and a grey button labeled 'Fin' are visible.

- The "Work Label" parameter of the form contains the enumerations that have been declared in the configuration screen Main Menu>Configuration>CellParameter "Work Enumeration".
- The parameter Job No. contains a freely definable number of type INT. This parameter can be used to identify which process position the robot was at during a clearance request, for example.

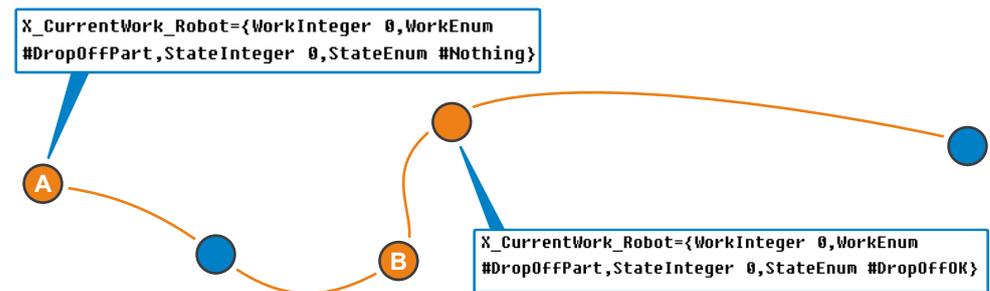
- The "Status Label" parameter of the form contains the enumerations that have been declared in the configuration screen Main Menu>Configuration>CellParameter "Work Enumeration".
- The parameter Status no. contains a freely definable number of type INT. This parameter can be used to identify which process state the robot was in when a clearance request was made, e.g.

The following instructions are written in the pick-up trajectory:



- A 1st point of the trajectory
- B Part pick-up point

The following instructions are written in the drop-off trajectory:



- A 1st point of the trajectory
- B Part drop-off point

- Equation locking method in the cycle code management window screen.
 - Assignment of cycle code for the part pick-up trajectory.
 $((X_CurrentWork_Robot.WorkValue==\#Nothing) \text{ AND } (X_CurrentWork_Robot.StateValue==\#Nothing)) \text{ OR } ((X_CurrentWork_Robot.WorkValue==\#DropOffPart1) \text{ AND } (X_CurrentWork_Robot.StateValue==\#DropOffOk))) \text{ AND } I_Machine1_Prete$
 - Assignment of cycle code for the part drop-off trajectory.
 $(X_CurrentWork_Robot.WorkValue==\#PickUpPart1) \text{ AND } (X_CurrentWork_Robot.StateValue==\#PickUpPartOK) \text{ AND } I_Machine2_Prete$

This means that the pick-up and drop-off trajectories can only be executed under very specific conditions.

The advantage of this process is that you can find out which state of the process the robot is in at any time, and above all you can manage escapes in the event of a problem.

13.8 Creating a template for work trajectories

Description

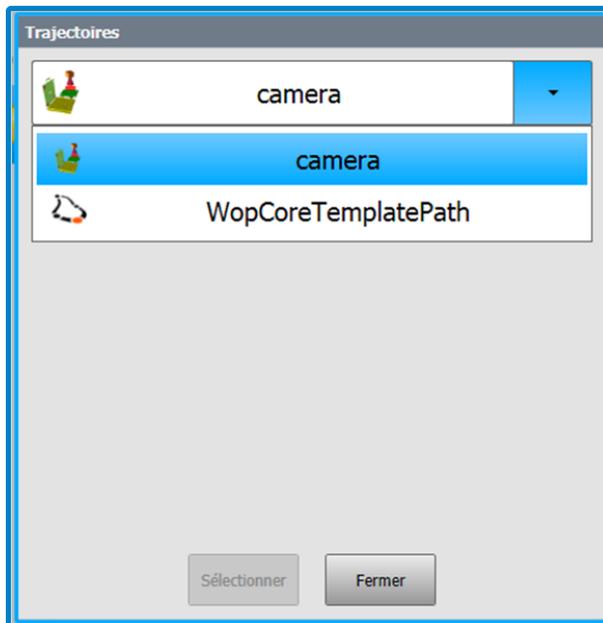
You can create your own trajectory template to make it easier and faster to create a trajectory with a well-defined format.

Condition

- User group: Expert.

Procedure

- Create a trajectory template with three files: TemplateName.src, TemplateName.dat, and TemplateName.png (icon representing the template type).
- In the file TemplateName.dat, there is a variable to be entered PathType\[]="TemplateName".
- Copy these 3 files to the directory "C:\KRC\USER\WopCore\DataProd\TrajectoriesTemplates".
- A new template is available in the trajectory creation window.



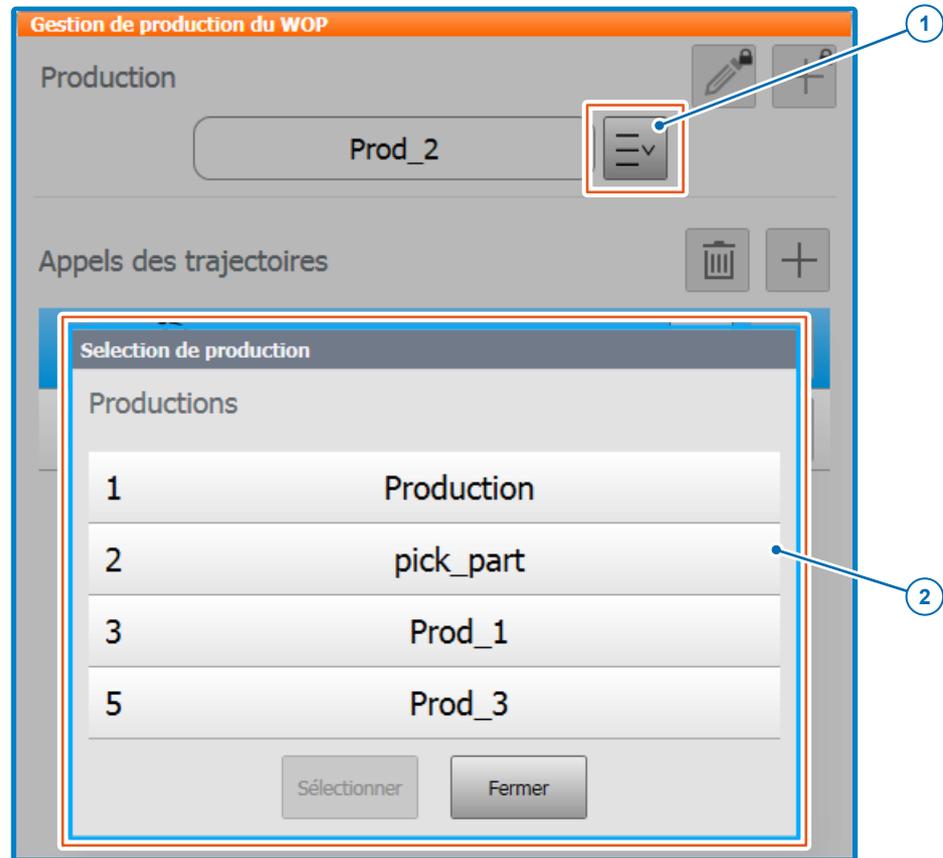
13.9 Selecting a production via the SmartPad screen

Condition

- Mode T1 and Expert Mode.

Procedure

- Go into the menu "WOP > Production". A screen opens:



- 1 Production list icon
- 2 Productions list

- Click the icon (1) to choose a production from the list.

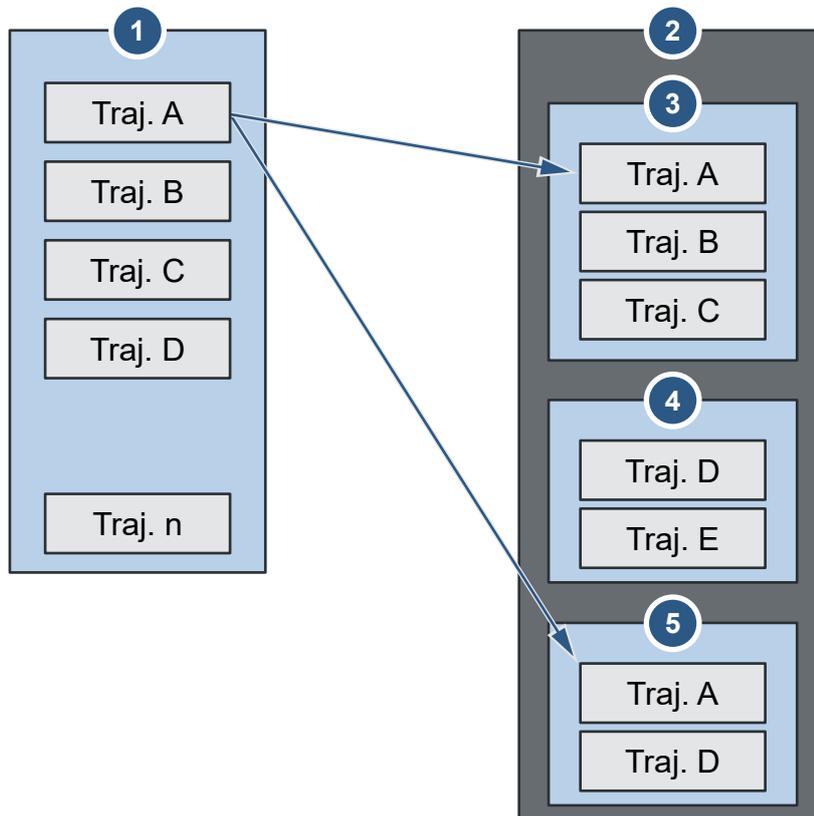
14 Management of several productions

Description

On the robot there is a library of trajectories that can be linked to created productions. We also then have a library of productions.

This allows the management of "trajectory batches" that can be loaded onto the robot according to the production.

This allows you to load only the trajectories related to the production in progress, to avoid saturating the robot's RAM.

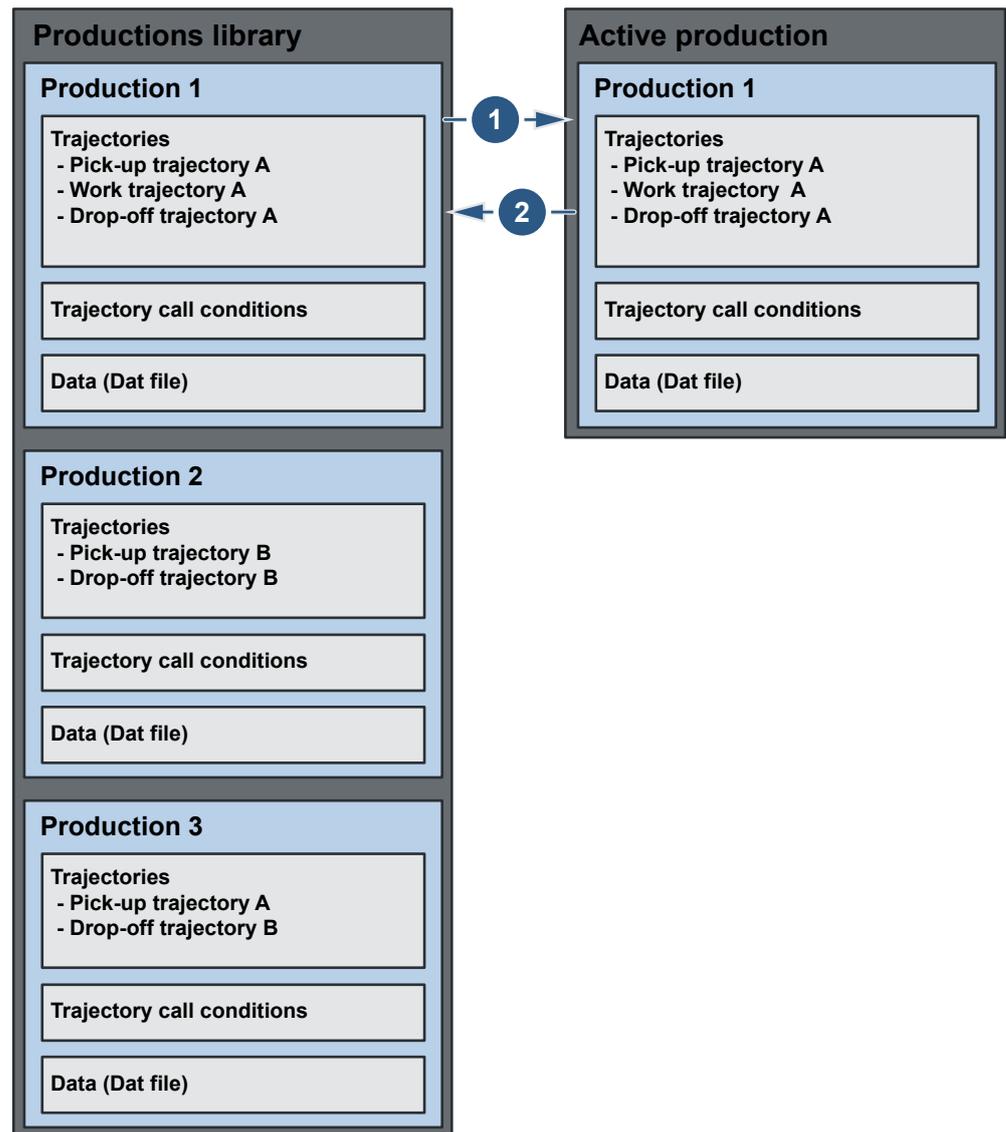


Traj Trajectory

- 1 Trajectory library
- 2 Production library
- 3 Introduction 1
- 4 Introduction 2
- 5 Introduction 3

Each trajectory can be associated with any production.

A production is a group of several trajectories that can be loaded into the robot memory.



- 1 Loading
- 2 Back-up

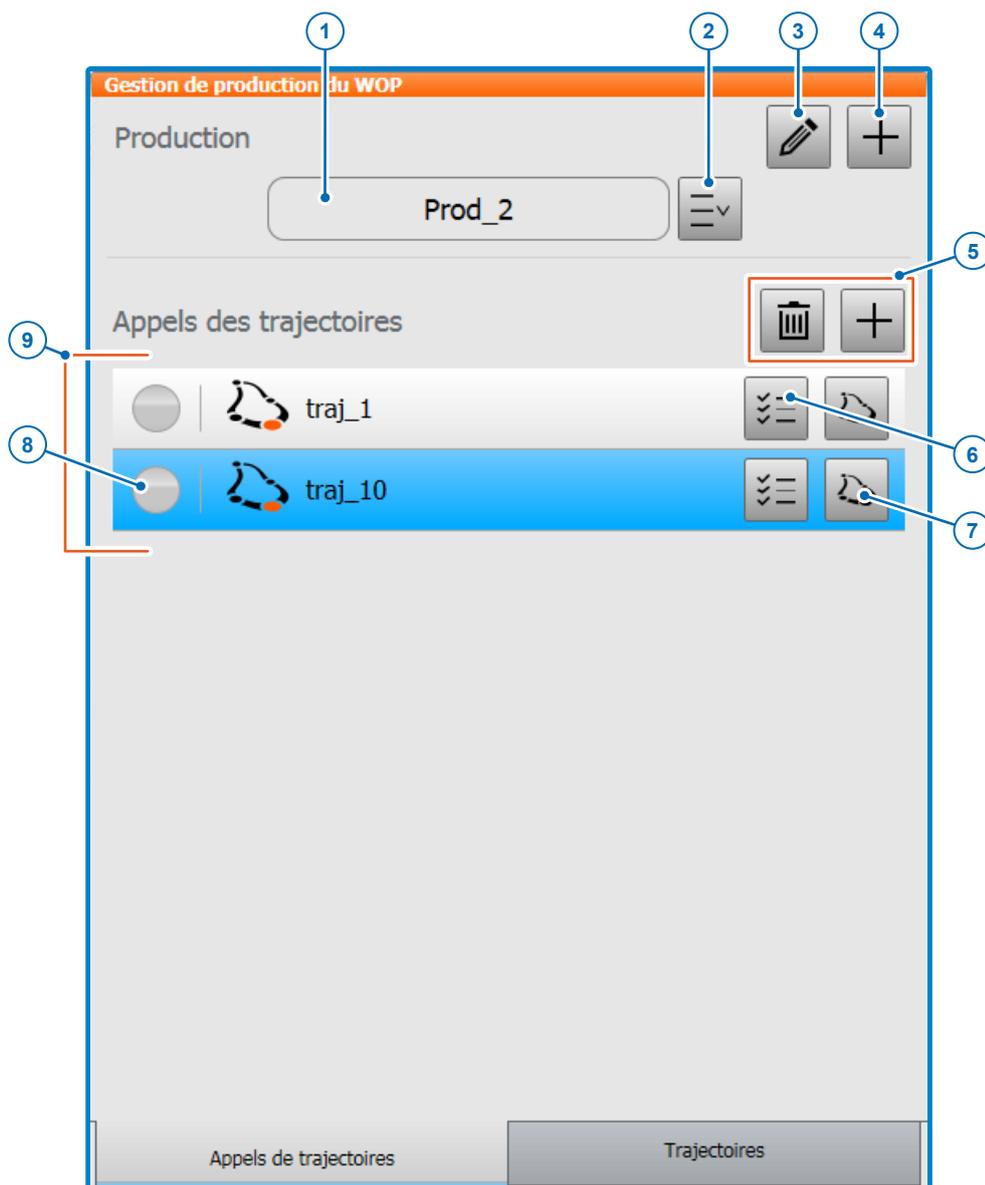
The active production is the one visible in the robot's standard tree structure.

Condition

- You must be in T1, in expert mode and program aborted.

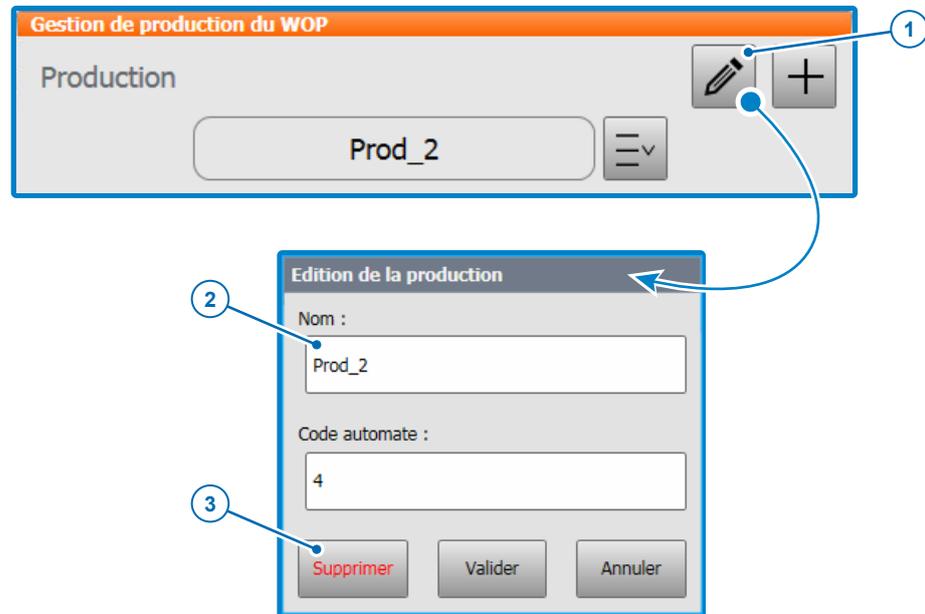
Procedure

- Go into the "WOP-Production" menu. The production management screen opens.



- 1 Active production
- 2 Choice of production
- 3 Edit production
- 4 Add a production
- 5 Add deletion of production trajectory

- 6 Configuration of the cycle code equation for this trajectory
 - 7 Choice of trajectory
 - 8 State of the equation for this trajectory
 - 9 List of production trajectories
- To load a new production, select it from the list of production choices (2).
 - The icon (4) is used to create a new production. A window opens.
 - Enter the name of the production.
 - Click Confirm.



- 1 Edit production
 - 2 Rename the active production
 - 3 Active production deletion
- The icon (3) is used to rename the active production. A window opens.
 - Enter the new name of the production.
 - Click Confirm.
 - Click on the "Active production deletion" button to delete the active production.
 - State of the equations.

	Code not configured
	Code configured but not valid
	Code valid

Each production has a WopUserParams.dat file in which the user can enter the variables that are specific to each production. This file is empty by default. When a production is loaded, the dat file corresponding to this production is located in the directory "R\TP\WOPCore\User\Production".

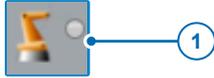
15 Shunt the MoveRobot in progress

Condition

- Select the following menu: Configuration > Function keys > WOP.
- Select mode T1 or mode T2.

Procedure

- Hold the deadman switch in the middle position.



1 "Shunt MoveRobot" key

- Press the key (1) to shunt the current MoveRobot.

16 List of error codes

No	Message	Effect	Cause	Solution
1	The robot is in an unknown position. Note: the robot will go directly to its starting position. Switch to T1.	Robot movement stopped.	A robot start-up action is in progress, although the robot was not at a recognized point.	Run the Master in T1, watching the robot to avoid any risk of collision.
2	Switch to T1 manual mode to return to the ParkingPos point.	Robot movement stopped.	A robot start-up action is in progress, although the robot was not at a recognized point.	Run the Master in T1, watching the robot to avoid any risk of collision.
3	The robot is in startup position. Switch to Automatic mode to start the process.	Robot movement stopped.	The robot is in LOOPPOS position after manual guidance.	Switch to Auto mode.
5	Robot stopped by user input Stop.	Robot movement stopped.	Robot stop request with input R I_StopCycle_Request.	Apply a Start to resume.
6	A WOP application asked to return to the Master program.	The trajectory in progress is aborted.	A request for return to master was triggered by the conditions of the variable X_WopAppReq_ReturnMaster.	-
8	Move the robot to service point %1, and then change the coordinates of point %1.	-	The service point is being learned.	-
10	One or more service points have not been learned or have been modified. Please select the service trajectories and learn the points.	Robot movement stopped in Auto mode with the master.	The robot is launched in Auto while one or more service positions have not been learned.	Switch to mode T1. Learn the service points.
11	A block coincidence was performed. The next points will be executed at actual speed (T2).	The robot can now continue at actual speed if T2.	The user executed a block coincidence. The robot is physically at a recognized point.	-
12	No escape strategy was programmed in the trajectory W_Escape.	Robot movement stopped.	An escape request is submitted by the user or the PLC, but no escape strategy is programmed.	Learning an escape strategy in the program W_Escape.SRC.
15	Note: the robot will go directly to its parking position.	Robot movement stopped.	A robot start-up action is in progress, although the robot was not at a recognized point.	Run the Master in T1, watching the robot to avoid any risk of collision.
16	Note: the robot will go directly to its LoopPos position.	Robot movement stopped.	A robot start-up action is in progress, although the robot was not at a recognized point.	Run the Master in T1, watching the robot to avoid any risk of collision.
17	Unknown cycle code (%1).	No effect; the robot is waiting for an executable code.	The requested cycle code is unknown and does not match any work saved in the robot.	Check the code sent.
18	Robot stopped by a WOP application.	Robot movement stopped.	The robot is stopped at the request of an application. Dynamic control was triggered by a signal forcing the robot to stop.	Restore or check the dynamic control signals.
19	Stop robot by the user flag.	Robot movement stopped.	The robot is stopped by the user-defined dynamic control.	Restore or check the dynamic control signals.

No	Message	Effect	Cause	Solution
101	Robot in parking position (ParkingPos).	-	-	-
102	Incorrect program selected for automatic mode.	Robot movement stopped.	A program other than the Master program is selected for automatic startup.	Abort all programs in progress and restart the automatic startup procedure.
103	Robot out of trajectory. Switch to mode T1. Run a COI, and then restart in Automatic mode.	Robot movement stopped.	The robot was moved manually when it was on a trajectory in progress.	Run a block coincidence.
104	Run a direct to home to the ParkingPos point in T1 manual mode.	Robot movement stopped.	The robot has left its trajectory.	-
105	Waiting for cycle code.	-	The robot is waiting for the cycle execution conditions.	-
106	The LoopPos service point was modified or has not yet been learned.	Robot movement stopped.	The LoopPos point has not been learned or was modified.	Learn the LoopPos point.
107	The ParkingPos service point was modified or has not yet been learned.	Robot movement stopped.	The ParkingPos point has not been learned or was modified.	Learn the ParkingPos point.
108	No robot movement authorization.	Robot movement stopped.	<p>The input "I_MoveRobot_Ok" is not set to "TRUE".</p> <p>The doors are not closed.</p> <p>The robot has not finished its direct to home to the ParkingPos point.</p> <p>The key is in mode T2.</p> <p>One or more submit tasks are not running.</p> <p>The variable "M_MoveRobot_User" is not set to "TRUE".</p>	<p>Check the value of input "I_MoveRobot_Ok".</p> <p>Check that doors are closed.</p> <p>Complete the direct to home to the ParkingPos point in mode T1, or reset the program.</p> <p>Mode T2 is prohibited. Turn the key to position T1.</p> <p>Check the submit tasks.</p> <p>Check the "M_MoveRobot_User" equation.</p>
109	The current automatic operating mode is different from the configured mode.	Robot stopped.	Inconsistent start mode configuration.	Check the "X_Start_Mode" variable. Possible values: (#AUT, #EXT_PLC, #EXT_SFC).
116	The PLC requested a new production. Waiting for robot to stop.	Waiting for robot to stop.	The PLC requested a new production.	-
117	Error loading new production. Check the files in C:\KRC\User\WopCore\DataProd.	Cannot load the production.	Error loading new production.	Check the files in C:\KRC\User\WopCore\DataProd.

No	Message	Effect	Cause	Solution
118	Several cycle equations are active at the same time. Cannot execute a cycle.	Cannot start production.	Several cycle equations are active at the same time.	Check the interlocks in the cycle code equations.
119	Production management plugin error. Must restart the software.	Robot movement stopped.	Production management plugin error.	Must restart the software.
120	Waiting for PLC production change request reinitialization.	Robot movement stopped.	Waiting for PLC production change request reinitialization.	Reinitialize the production change request "I_LoadProd_Request".
121	No production loaded.	Robot movement stopped.	Waiting for production to load.	Check the loading sequence of a production.
122	No user cycle start authorization.	Robot movement stopped.	No user start authorization.	Check the equation of the M_ReadyForStart_User variable.
123	Mode T2 prohibited. Robot movement blocked.	Robot movement stopped.	Mode T2 selected.	Change the operator mode.
124	Dry run. All checks are disabled.	All checks are disabled.	Dry run enabled.	Deselect the Dry Run box in the production/operating mode menu.
201	WARNING: The stop robot function is disabled.	The stop robot function is disabled.	Activate the "Move enable" status key.	-
202	The stop robot function is restored.	The stop robot function is restored.	Release the "Move enable" status key.	-
203	Not all background tasks are enabled; execution of the trajectory is prohibited.	Robot movement stopped.	Not all background tasks are enabled.	Enable all background tasks.
1000	Configuration error. Certain necessary files or folders do not exist.	The WopProduction module is disabled.	The folder or system file of the WopProduction plugin was deleted.	It must be reinstalled.
1001	Invalid file path argument '{0}'.	-	-	Contact KUKA support.
1002	Invalid krl module name format '{0}'.	-	-	Contact KUKA support.
1003	Invalid module identifier format '{0}'.	-	-	Contact KUKA support.
1004	Inconsistent identifier for the module '{0}'.	-	-	Contact KUKA support.
1005	Invalid attributes for the module '{0}'.	-	-	Contact KUKA support.
1006	Inconsistent production loaded in the robot!	-	-	Contact KUKA support.
1007	Trajectory template not found '{0}'.	-	-	Contact KUKA support.
1008	Associated image not found for the trajectory template '{0}'.	-	-	Contact KUKA support.
1009	No trajectory found with the ID '{0}' for cycle code N ^o {1}.	-	-	Contact KUKA support.
1010	Production xml file not recognized '{0}'.	-	-	Contact KUKA support.
1011	The following trajectories: {0} are inconsistent.	-	-	Contact KUKA support.
1012	The following productions had to be modified following the deletion of trajectories: {0}.	-	-	Contact KUKA support.
1013	The following trajectories were duplicated manually: {0}.	-	-	Contact KUKA support.
1014	Failed to delete production {0}.	-	-	Contact KUKA support.

No	Message	Effect	Cause	Solution
1015	Unknown folder '{0}'.	-	-	Contact KUKA support.
1016	Failed to read cycle code equations.	-	-	Contact KUKA support.
1017	Failed to read trajectory calls.	-	-	Contact KUKA support.
1018	Failed to create a new production.	-	-	Contact KUKA support.
1019	Failed to rename a production '{0}' as '{1}'.	-	-	Contact KUKA support.
1020	Failed to duplicate trajectory.	-	-	Contact KUKA support.
1021	Failed to create trajectory.	-	-	Contact KUKA support.
1022	Failed to synchronize on deletion of trajectory.	-	-	Contact KUKA support.
1023	Failed to create new trajectory call.	-	-	Contact KUKA support.
1024	Failed to delete trajectory call.	-	-	Contact KUKA support.
1025	Failed to synchronize to the krc.	-	-	Contact KUKA support.
1026	Failed to obtain information on the trajectories to add or delete.	-	-	Contact KUKA support.
1027	Unknown production loaded '{0}'! Empty production loaded.	-	-	Contact KUKA support.
1028	Productions disabled: {0}.	-	-	Contact KUKA support.
1029	Duplication of PLC codes for the productions: {0}.	-	-	Contact KUKA support.
1030	Production N°{0} '{1}' was loaded.	-	-	Contact KUKA support.
1031	Production N°{0} '{1}' was loaded with errors!	-	-	Contact KUKA support.
1032	Production N°{0} '{1}' was deleted.	-	-	Contact KUKA support.
1033	Failed to rename trajectory.	-	-	Contact KUKA support.
1034	Failed to import trajectory: No template in the database.	-	-	Contact KUKA support.
1035	Failed to duplicate production.	-	-	Contact KUKA support.

17 KUKA After-sales service

Introduction

The KUKA Roboter GmbH documentation includes a great deal of information concerning service and ordering, and will be helpful with troubleshooting. Your local branch is available to provide any additional information or answer any other questions you might have.

Information

To handle a question, we need the following information:

- Robot type and serial number.
- Order type and serial number.
- Type and serial number of the linear unit (optional).
- KUKA System Software version.
- Optional software or modifications.
- Software archives.

- Existing application.
- Existing additional axes (optional).
- Description of the problem, duration and frequency of the fault.

Availability

Our KUKA customer support service is available in several countries. We are here to answer all of your questions.

South Africa
 Jendamark Automation LTD
 (Agency)
 76a York Road
 North End
 6000 Port Elizabeth
 South Africa
 Tel. +27 41 391 4700
 Fax +27 41 373 3869
 www.jendamark.co.za

Germany
 KUKA Roboter GmbH
 Zugspitzstr. 140
 86165 Augsburg
 Germany
 Tel. +49 821 797-4000
 Fax +49 821 797-1616
 info@kuka-roboter.de
 www.kuka-roboter.de

Argentina
 Ruben Costantini S.A. (Agency)
 Luis Angel Huergo 13 20
 Parque Industrial
 2400 San Francisco (CBA)
 Argentina
 Tel. +54 3564 421033
 Fax +54 3564 428877
 ventas@costantini-sa.com

Brazil
 KUKA Roboter do Brasil Ltda.
 Avenida Franz Liszt, 80
 Parque Novo Mundo
 Jd. Guançã
 CEP 02151 900 São Paulo
 SP Brazil
 Tel. +55 11 69844900
 Fax +55 11 62017883
 info@kuka-roboter.com.br

Australia
 Headland Machinery Pty. Ltd.
 Victoria (Head Office & Showroom)
 95 Highbury Road
 Burwood
 Victoria 31 25
 Australia
 Tel. +61 3 9244-3500
 Fax +61 3 9244-3501
 vic@headland.com.au
 www.headland.com.au

Austria
 KUKA Roboter Austria GmbH
 Regensburger Strasse 9/1
 4020 Linz
 Austria
 Tel. +43 732 784752
 Fax +43 732 793880
 office@kuka-roboter.at
 www.kuka-roboter.at

Belgium
 KUKA Automatisering + Robots N.V.
 Centrum Zuid 1031
 3530 Houthalen
 Belgium
 Tel. +32 11 516160
 Fax +32 11 526794
 info@kuka.be
 www.kuka.be

USA
 KUKA Robotics Corp.
 22500 Key Drive
 Clinton Township
 48036
 Michigan
 USA
 Tel. +1 866 8735852
 Fax +1 586 5692087
 info@kukarobotics.com
 www.kukarobotics.com

Chile
Robotec S.A. (Agency)
Santiago de Chile
Chile
Tel. +56 2 331-5951
Fax +56 2 331-5952
robotec@robotec.cl
www.robotec.cl

China
KUKA Automation Equipment
(Shanghai) Co., Ltd.
Songjiang Industrial Zone
No. 388 Minshen Road
201612 Shanghai
China
Tel. +86 21 6787-1808
Fax +86 21 6787-1805
info@kuka-sha.com.cn
www.kuka.cn

Korea
KUKA Robotics Korea Co. Ltd.
RIT Center 306, Gyeonggi
Technopark
1271-11 Sa 3-dong, Sangnok-gu
Ansan City, Gyeonggi Do
426-901
Korea
Tel. +82 31 501-1451
Fax +82 31 501-1461
info@kukakorea.com

Spain
KUKA Robots IBÉRICA, S.A.
Pol. Industrial
Torrent de la Pastera
Carrer delBages s/n
08800 Vilanova i la Geltrú
(Barcelona)
Spain
Tel. +34 93 8142-353
Fax +34 93 8142-950
Comercial@kuka-e.com
www.kuka-e.com

France
KUKA Automatisme + Robotique
SAS
Techvallée
6, avenue du Parc
91140 Villebon S/Yvette
France
Tel. +33 1 6931660-0
Fax +33 1 6931660-1
commercial@kuka.fr
www.kuka.fr

Hungary
KUKA Robotics Hungaria Kft.
Főút 140
2335 Taksony
Hungary
Tel. +36 24 501609
Fax +36 24 477031
info@kuka-robotics.hu

India
KUKA Robotics India Pvt. Ltd.
Office Number-7, German Centre,
Level 12, Building No. - 9B
DLF Cyber City Phase III
122 002 Gurgaon
Haryana
India
Tel. +91 124 4635774
Fax +91 124 4635773
info@kuka.in
www.kuka.in

Italy
KUKA Roboter Italia S.p.A.
Via Pavia 9/a - int.6
10098 Rivoli (TO)
Italy
Tel. +39 011 959-5013
Fax +39 011 959-5141
kuka@kuka.it
www.kuka.it

Japan
 KUKA Robotics Japan K.K.
 Daiba Garden City Building 1F
 2-3-5 Daiba, Minato-ku
 Tokyo
 135-0091
 Japan
 Tel. +81 3 6380-7311
 Fax +81 3 6380-7312
 info@kuka.co.jp

Malaysia
 KUKA Robot Automation Sdn Bhd
 South East Asia Regional Office
 No. 24, Jalan TPP 1/10
 Taman Industri Puchong
 47100 Puchong
 Selangor
 Malaysia
 Tel. +60 3 8061-0613 or -0614
 Fax +60 3 8061-7386
 info@kuka.com.my

Mexico
 KUKA de Mexico S. de R.L. de C.V.
 Rio San Joaquin #339, Local 5
 Colonia Pensil Sur
 C.P. 11490 Mexico D.F.
 Mexico
 Tel. +52 55 5203-8407
 Fax +52 55 5203-8148
 info@kuka.com.mx

Norway
 KUKA Sveiseanlegg + Roboter
 Bryggeveien 9
 2821 Gjøvik
 Norway
 Tel. +47 61 133422
 Fax +47 61 186200
 geir.ulsrud@kuka.no

Poland
 KUKA Roboter Austria GmbH
 Spółka z ograniczoną
 odpowiedzialnością
 Oddział w Polsce
 Ul. Porcelanowa 10
 40-246 Katowice
 Poland
 Tel. +48 327 30 32 13 or -14
 Fax +48 327 30 32 26
 ServicePL@kuka-roboter.de

Czech Republic
 KUKA Roboter Austria GmbH
 Organisation Tschechien und
 Slowakei
 Sezemická 2757/2
 193 00 Praha
 Horní Počernice
 Czech Republic
 Tel. +420 22 62 12 27 2
 Fax +420 22 62 12 27 0
 support@kuka.cz

UK
 KUKA Automation + Robotics
 Hereward Rise
 Halesowen
 B62 8AN
 UK
 Tel. +44 121 585-0800
 Fax +44 121 585-0900
 sales@kuka.co.uk

Russia
 KUKA Robotics Rus
 Webnaja ul. 8A
 107143 Moskau
 Russia
 Tel. +7 495 781-31-20
 Fax +7 495 781-31-19
 www.kuka-robotics.ru

Sweden
 KUKA Svetsanläggningar + Robotar
 AB
 A. Odhnersgata 15
 421 30 Västra Frölunda
 Sweden
 Tel. +46 31 7266-200
 Fax +46 31 7266-201
 info@kuka.se

Switzerland
 KUKA Roboter Schweiz AG
 Industriestr. 9
 5432 Neuenhof
 Switzerland
 Tel. +41 44 74490-90
 Fax +41 44 74490-91
 info@kuka-roboter.ch
 www.kuka-roboter.ch

A	
API	7
Auto	23
D	
dynamic control	61
E	
EnumStateRobot	16
EnumWorkRobot	16
Ext PLC	23
Ext SFC	23
F	
First_Bit_FaultCode	36
I	
I_Access_Request	32
I_Cycle_Start	23
I_EndCycle_Request	16
I_LoadProd_Request	42
I_MoveRobot_Ok	23
I_NumLoadProd_Request	42
I_Parking_Request	16
I_StopCycle_Request	23
M	
M_FirstBitMessage_ToPlc	34
M_LengthMessage_ToPlc	34
M_MoveRobot_User	58
mnemonics	9
M_Parking_IsRequested	58
M_ReadyForStart_User	58

O

O_Acces_Allowed	32, 57
O_Conf_Mess	34
O_DlgMessage_OnRobot	34
O_Drives_On.	23
O_External_Start.	23
O_GeneralError_Robot	34, 57
O_LoadProd_Error	42
O_LoadProd_Loaded	42
O_NumLoadProd_Loaded.	42
O_ReadyFor_LoadProd	42
O_ReadyFor_Start	23, 55
O_Rob_InCycle.	16, 57
O_Rob_LoopPos1.	16, 56
O_Rob_Manu	39, 56
O_Rob_Parking.	56
O_Rob_RunInAuto	39, 55

P

PLC	7
---------------	---

X

X_Allow_ParkingService	57
X_Cycle_DryRun.	39
X_MasterReset_User	58
X_PosRobot_OK.	58
X_ProductionMode_Enable.	42

KUKA



KUKA FRANCE

Techvallée

6, avenue du Parc

F-91140 Villebon-sur-Yvette

T +33.1.69.31.66.00

F +33.1.69.31.66.01

www.kuka.fr