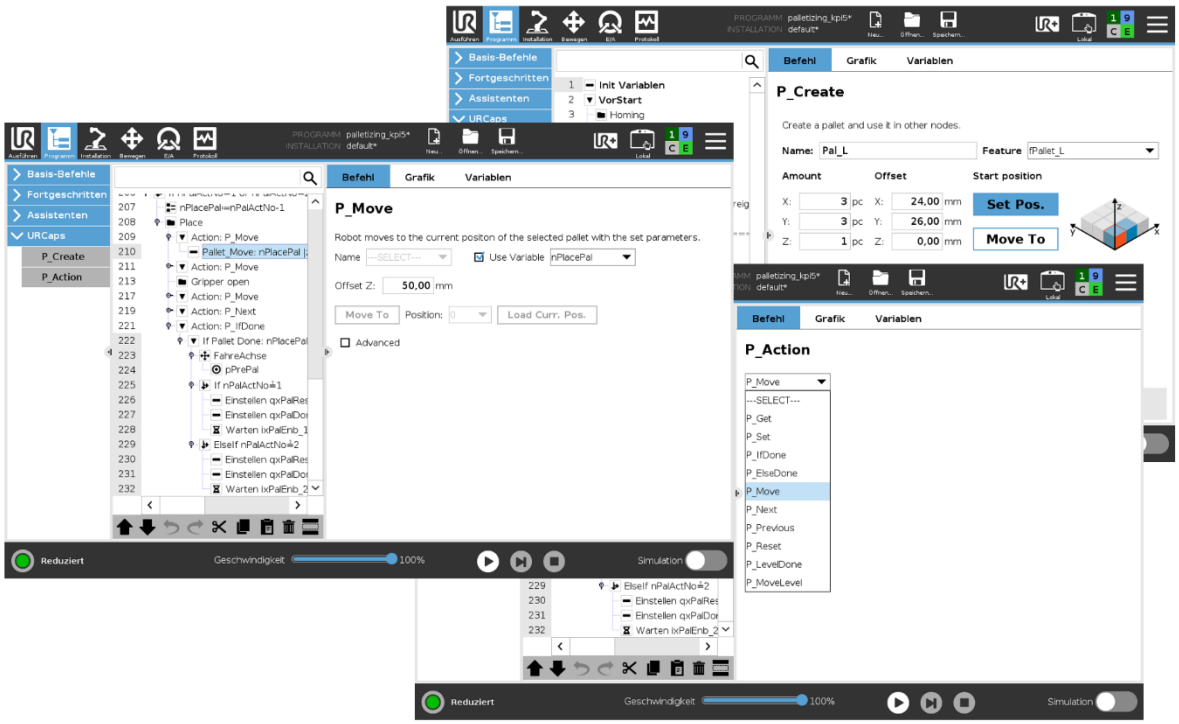
Reference Manual

URCap Advanced Palletizing – Version 1.0.7



KPI Software GmbH

Im Winkel 1

78588 Denkingen

Phone: +49 7424 7031610

E-Mail: info@kpi-software.de

Internet: www.kpi-software.de

Reference Manual

Version 1.0.7

© 2022

Table of Contents

1 Introduction 5
1.1 About this document 5
1.2 Requirements and supported versions 5
1.3 Update URCap 5
2 Installation 6
2.1 Installing the URCap 6
2.2 Uninstall the URCap 9
3 Installation Node 10
3.1 Licensing 10
3.2 Online help 11
4 Program Node P_Create 12
4.1 Insert P_Create 12
4.2 Create a new pallet 13
4.3 Copy a pallet 14
4.4 Delete a pallet 14
4.5 Enter pallet values 15
4.6 Types of pallets 17
4.6.1 Uniform grid 17
4.6.2 Offset grid 18
4.7 External axes 19
5 Program Node P_Action 20
5.1 Insert P_Action 20
5.2 P_Reset: Pallet reset 21
5.3 P_Move: Approach pallet position 22
5.4 P_Next: Increase pallet counter 24
5.5 P_Previous: Decrease pallet counter 25
5.6 P_IfDone: Query whether the pallet has been processed 26
5.7 P_NotDone: Query whether the pallet is still being processed 27
5.8 P_Get: Read out current pallet position 28
5.9 P_Set: Write pallet position 29
5.10 P_IfLevelDone: Query whether the current level has been processed 31
5.11 P_MoveLevel: Approach the intermediate layer 32
6 Troubleshooting 33
7 Directories 34
7.1 List of figures 34

8 Notes 35

1 Introduction

The URCap Advanced Palletizing is a software extension for the UR robot (Universal Robots). It was developed to implement complex palletizing tasks with minimal effort. Due to the flexible program structure, the most varied of palletizing tasks can be solved easily and clearly and the programming effort is significantly reduced. Even with processes with many pallets, no performance losses are to be feared. Subsequent adjustments such as the number of pieces on the pallet or the processing sequence can be changed with just a few clicks.

1.1 About this document

The reference manual contains an overview of all functions of the URCap. It was created for robot programmers, software developers and maintenance technicians.

1.2 Requirements and supported versions

E-Series robots (UR3, UR5, UR10 or UR16) from PolyScope 5.9.

1.3 Update URCap

Attention: Robot programs that were created with a previous version may no longer be used. The robot programs and the robot installation may have to be recreated or adapted. To install the URCap version 1.0.7 on a system where an earlier version is already installed.

- Uninstall the previous version
- Check the PolyScope version, if necessary update to a newer version (version 5.9)
- To avoid configuration conflicts, create a new robot installation
- Install URCap

2 Installation

2.1 Installing the URCap

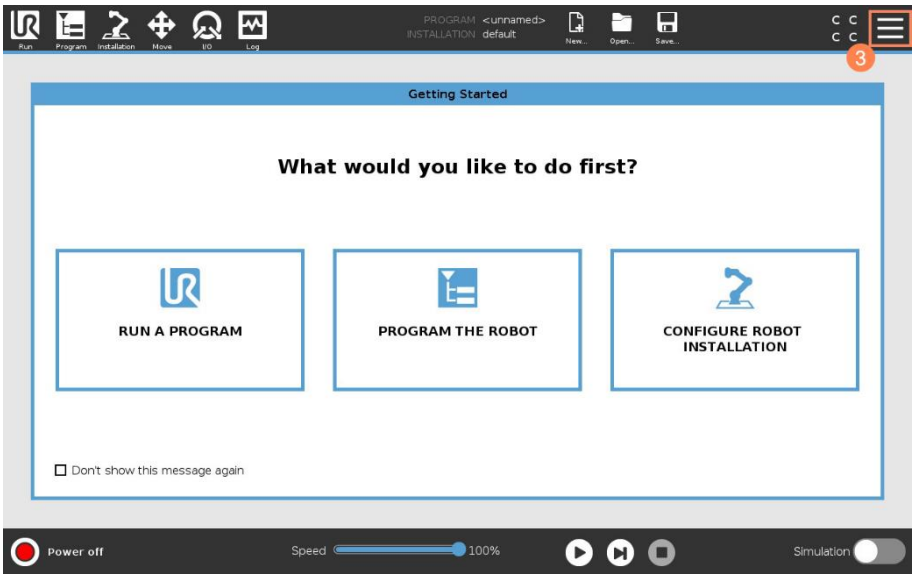


Figure 1: Home screen

1. Start the robot
2. Insert the USB stick with the URCap
3. Click the hamburger menu in the top right corner

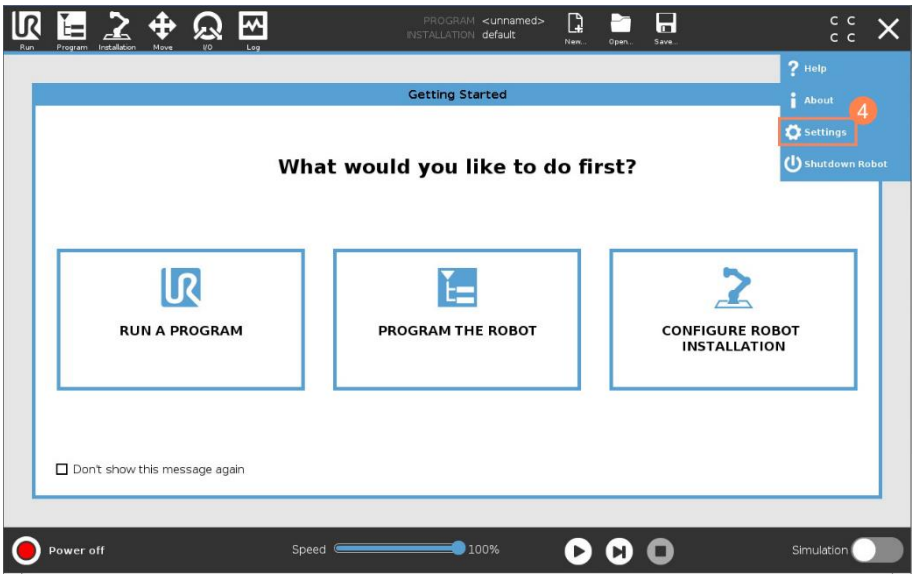


Figure 2: Select Settings

4. Click Settings

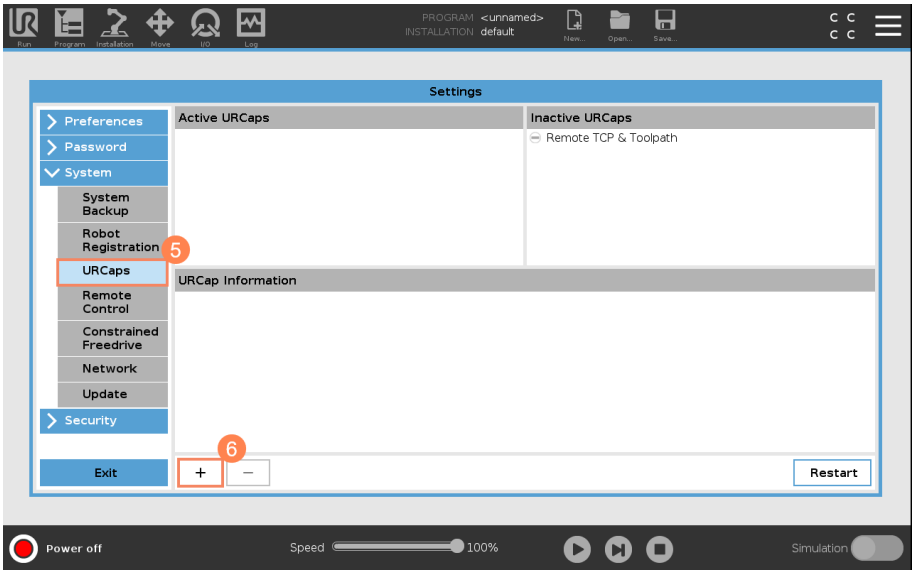


Figure 3: Add URCap

5. Click on URCaps
6. Click +

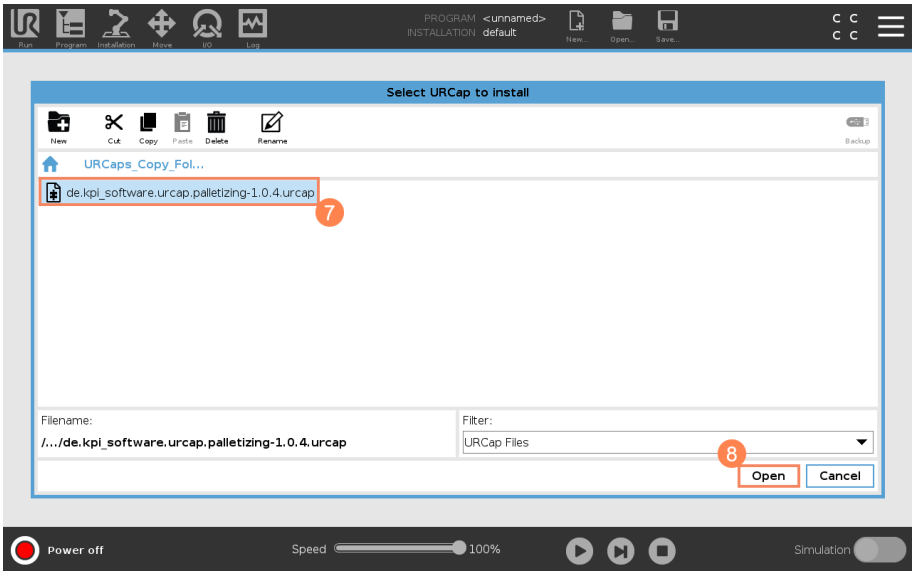


Figure 4: Select URCap on USB stick

7. Select the URCap on the USB stick
8. Click Open to install the URCap

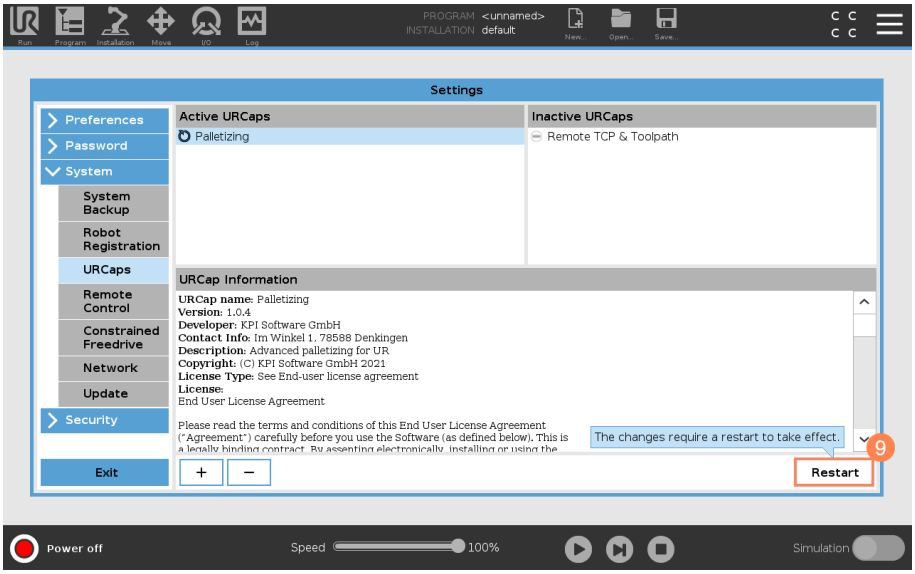


Figure 5: Restart the robot

9. Click Restart to restart the robot

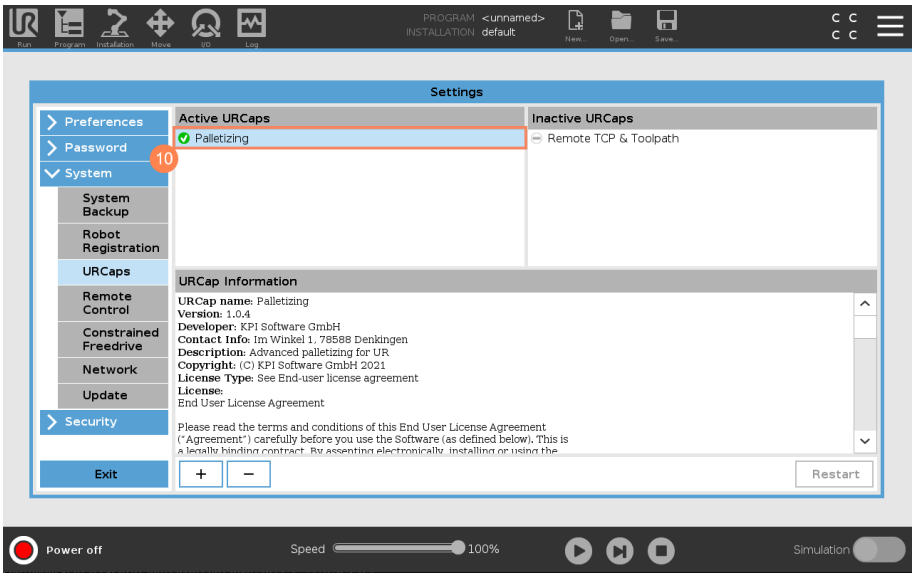


Figure 6: URCap is installed

10. A green tick will appear next to the URCap if it has been installed correctly

2.2 Uninstall the URCap

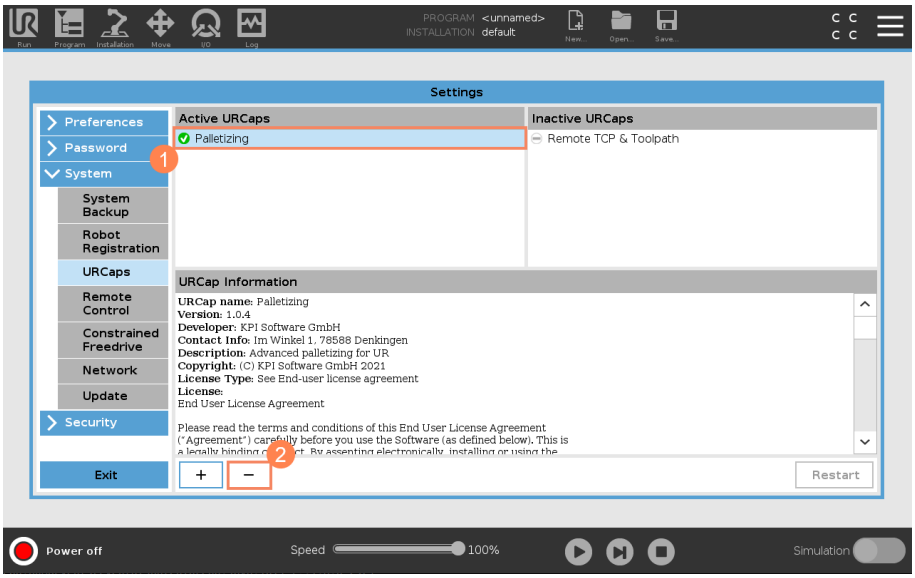


Figure 7: Select URCap

1. Select the URCap to be uninstalled
2. Click on -
3. Restart the robot

3 Installation Node

3.1 Licensing

In order for Advanced Palletizing to be used, a valid license key must be entered in advance. This is generated by the manufacturer with the help of the generator string.

Figure 8: Installation Node

1. Open the installation page
2. Go to URCaps
3. Click on Advanced Palletizing
4. Click Update License

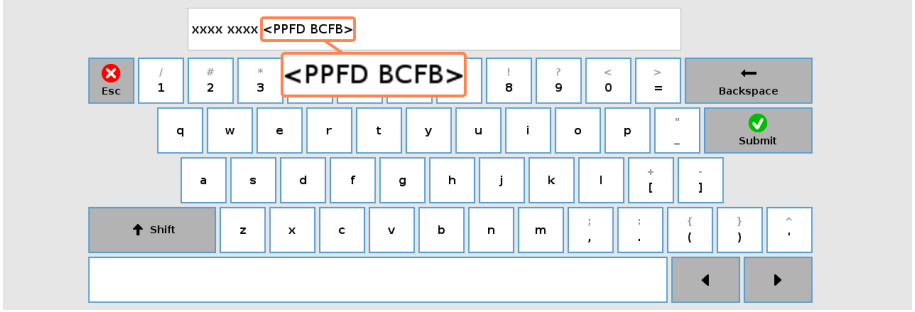


Figure 9: Generator string

5. Make a note of the 8-digit generator string and send it together with your contact details to redeemlicense@kpi-software.de

Figure 10: Enter license key

6. You will then receive your 8-digit license key, which must be entered instead of the generator string

Figure 11: Successful licensing

7. You can recognize successful activation by the green tick

3.2 Online help

Our online help is located in the installation node and contains brief information on the individual program nodes. To do this, simply click on the corresponding menu item on the left.

Figure 12: Online help

4 Program Node P_Create

The palettes are created and defined in the program node P_Create. It may only be inserted once in the pre-start, as it is not required for the normal program sequence. If you have accidentally inserted P_Create a second time, simply delete the second instance again.

4.1 Insert P_Create

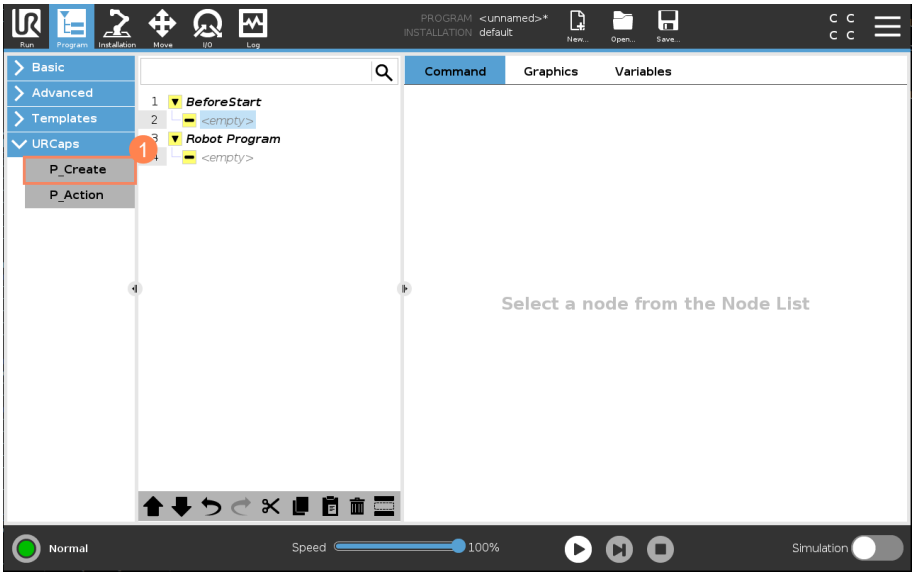


Figure 13: Program Node P_Create

1. Add P_Create to the pre-start

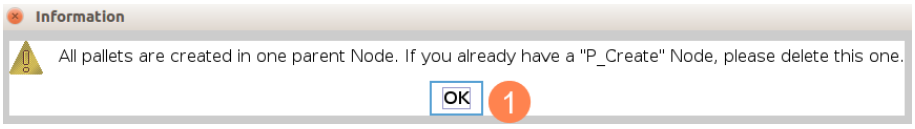


Figure 14: PopUp P_Create

1. When inserting P_Create, a message appears that you can only insert the program node once.

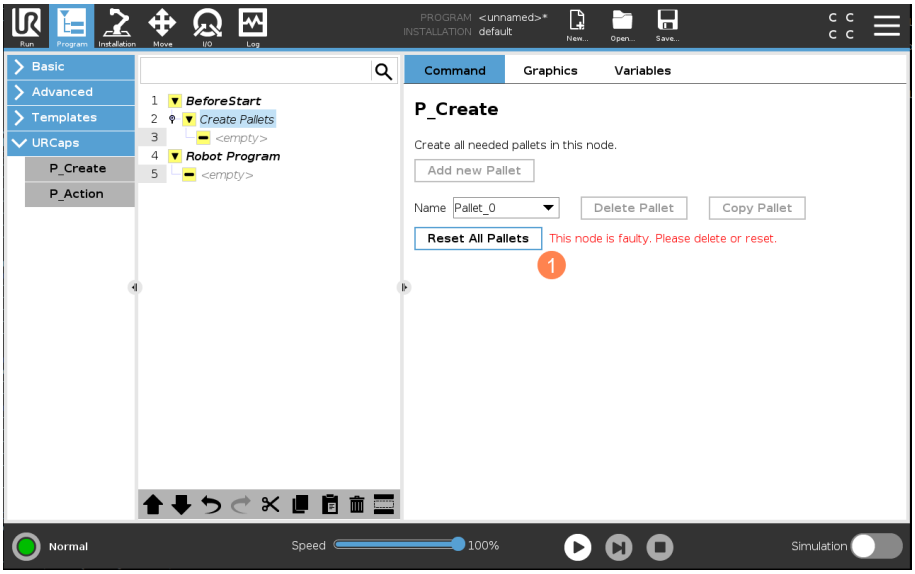


Figure 15: P_Create Reset

When inserting P_Create, you must first reset all pallets. To do this, click on Reset All Pallets. The red notice should then go away.
Do not do this step if you have accidentally inserted P_Create a second time.

4.2 Create a new pallet

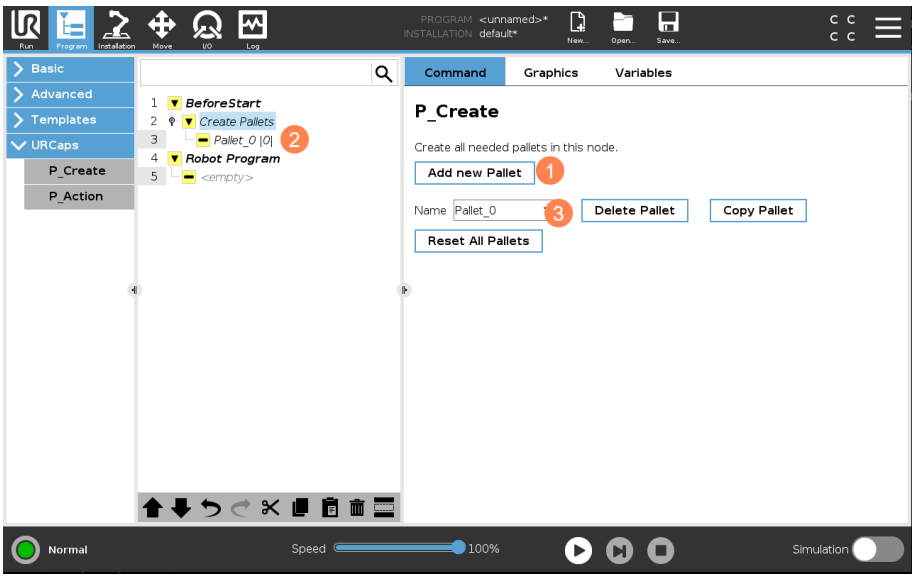


Figure 16: Create a new pallet

1. Click on Add new Pallet
2. The new pallet is added as a new node
3. The new pallet will appear in the drop-down menu

4.3 Copy a pallet

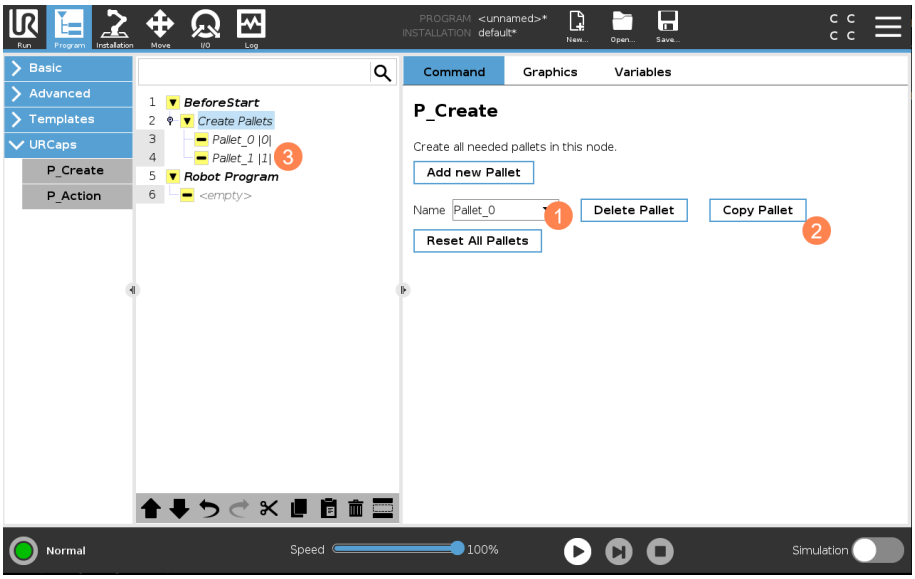


Figure 17: Copy a pallet

1. Select the pallet you want to copy
2. Click on Copy Pallet
3. The copied pallet is added as a new node

4.4 Delete a pallet

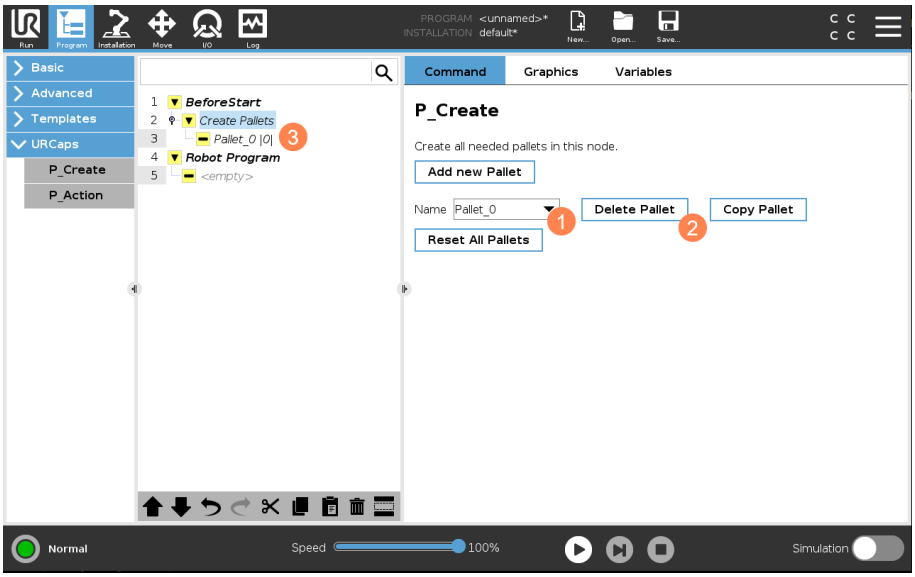


Figure 18: Delete a pallet

1. Select the pallet you want to delete
2. Click Delete Pallet
3. The pallet is deleted from the list

4.5 Enter pallet values

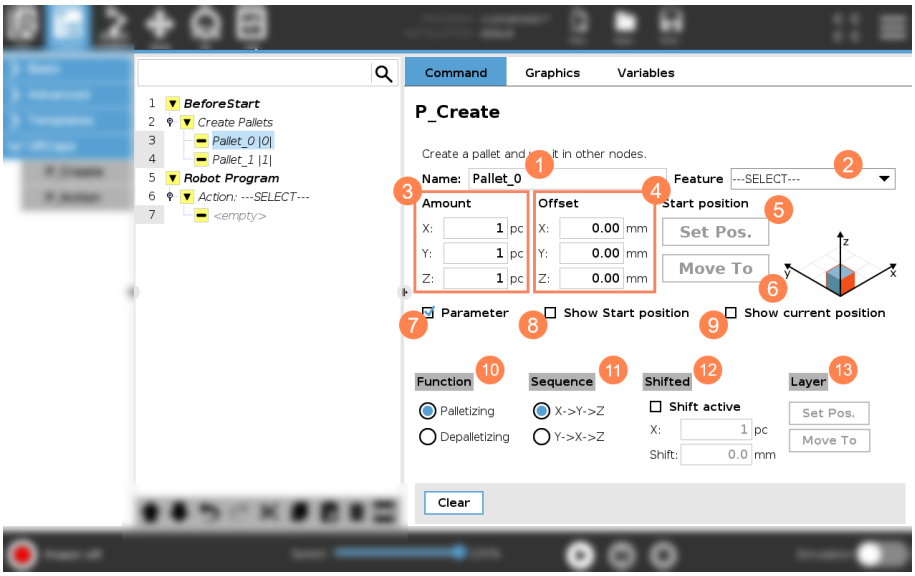


Figure 19: Pallet values

1. You can assign any name to the pallet under Name. This can be used to call up the palette later.
2. Under Feature you select an already created coordinate system for palletizing. It should be noted here that the X / Y / Z direction corresponds exactly to the position of the pallet. The zero position of the coordinate system does not have to coincide with the first pallet position. Often the locating bolts of the pallet are used to teach the coordinate system.
3. The number of workpieces on the pallet is entered under Amount.
4. The distance between the workpieces is entered under Offset. If the number of pieces is greater than one, the offset must not contain 0.
5. The starting position of the pallet is taught with Set Pos. Specifically, the exact gripping position of the first workpiece is to be taught in here.
6. The taught start position can be approached at any time with Move To.
7. In the case of simple pallets, nothing needs to be set under Parameter. However, if you want to make settings under Parameter, you can tick the checkbox.
8. With Show Start Position the start position is displayed as it was taught under Set Pos. Manual adjustments can also be made here.
9. With Show current position the current pallet counter is displayed.
10. Function is used to set whether workpieces are to be placed on the pallet (palletizing) or whether workpieces are to be taken from the pallet (depalletizing). With depalletizing, palletizing begins automatically with the last workpiece. As with palletizing, the start position is taught in the 1st position.
11. The palletizing sequence can be set under Sequence
12. Under Shifted you can set whether it is an offset pallet. A detailed description can be found in Chapter 4.6.2 Offset pallets.

13. Under Layer you can set whether an intermediate layer (e.g. a piece of cardboard) should be inserted between the layers. The position of the lowest layer is taught with Set Pos.

If all values have been entered correctly, the program node is no longer displayed in yellow.

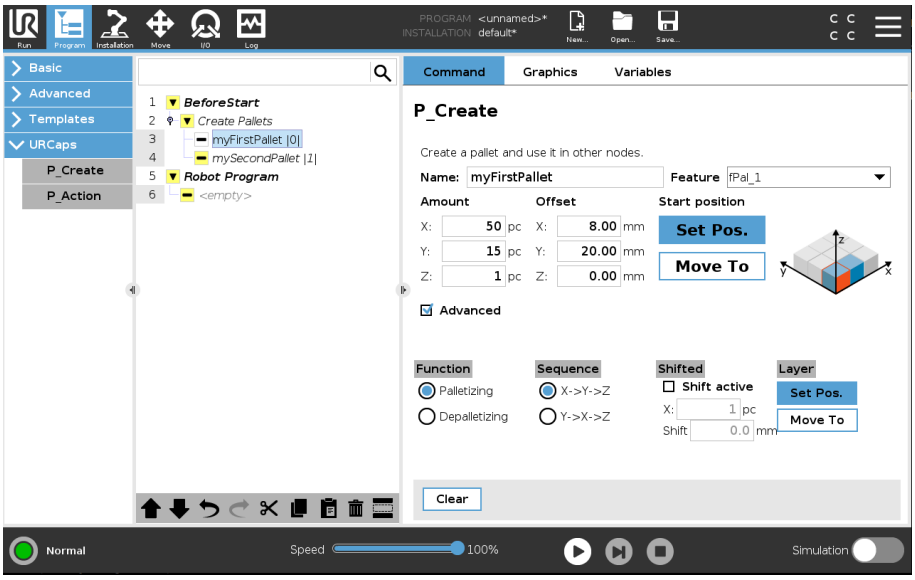


Figure 20: Example pallet

4.6 Types of pallets

4.6.1 Uniform grid

The simplest type of pallet are pallets with an even grid in X, Y and Z. For such pallets, it is sufficient to specify the number and the spacing of the rows, columns and levels. As soon as the start position has been taught in, you can work with such a pallet.

However, it must always be ensured that the selected coordinate system has exactly the same orientation as the pallet.

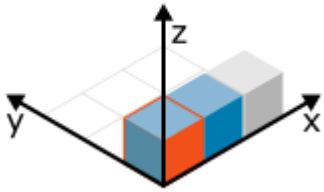


Figure 21: Example pallet 1 has 3 workpieces in X and 1 each in Y and Z.

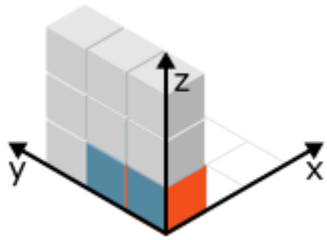


Figure 22: Example pallet 2 has 1 workpiece in X and 3 each in Y and Z.

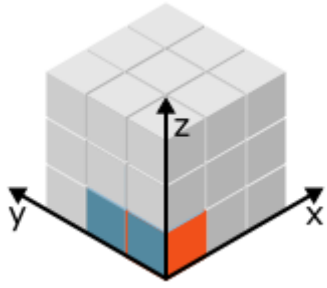


Figure 23: Example pallet 2 has 3 workpieces each in X, Y and Z.

4.6.2 Offset grid

The offset pallets are a special type of pallet, in which every 2nd row is either slightly offset or a different number of workpieces is available. It is also possible to combine both.

In the following you can see such a pallet, which has 12 positions in the 1st row and only 11 positions in the 2nd row. In addition, the workpieces in the 2nd row are offset to the right.

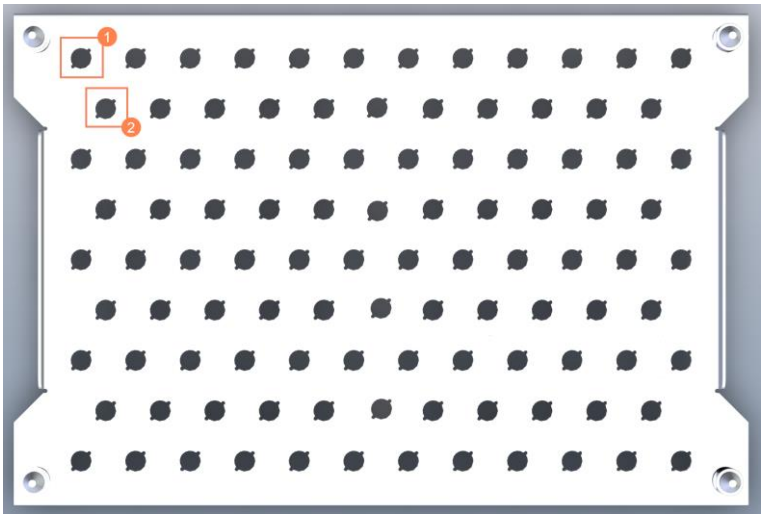


Figure 24: Offset grid

In the case of offset pallets, it must be ensured that the processing direction is always row by row. In the example, processing would have to be done first from left to right and then from top to bottom.

The configuration of the example pallet in P_Create would look like this:

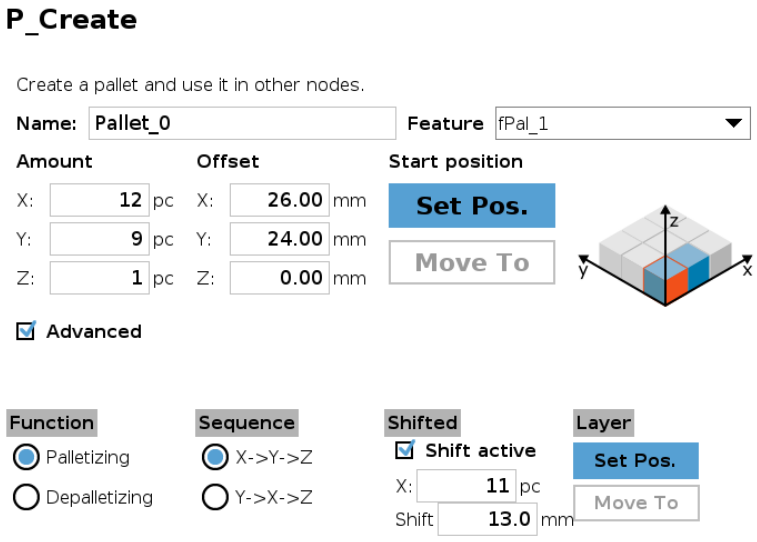


Figure 25: Offset pallet configuration

4.7 External axes

If the robot is mounted on an external axis, the current axis position can be specified as an offset. This offset is then taken into account when approaching the pallet position. The values are written to the variable kpi5_nExtAxisOff[X, Y, Z, RX, RY, RZ] using a script command.

For example, if the robot stands on an axis that makes a Z-stroke, the variable kpi5_nExtAxisOff[2] must be set to the current axis position before the P_Move command.

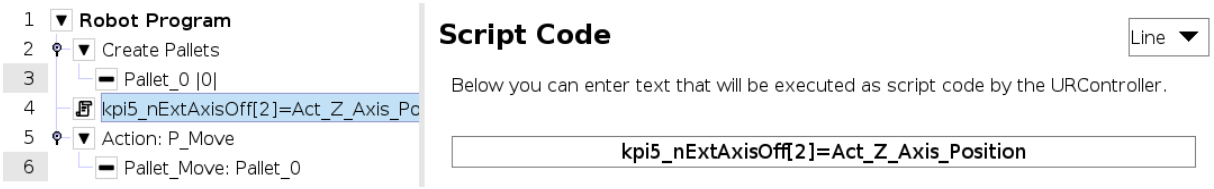


Figure 26: Script Code

5 Program Node P_Action

The various palette operations are called in the program node P_Action. It can be inserted at any point in the program.

5.1 Insert P_Action

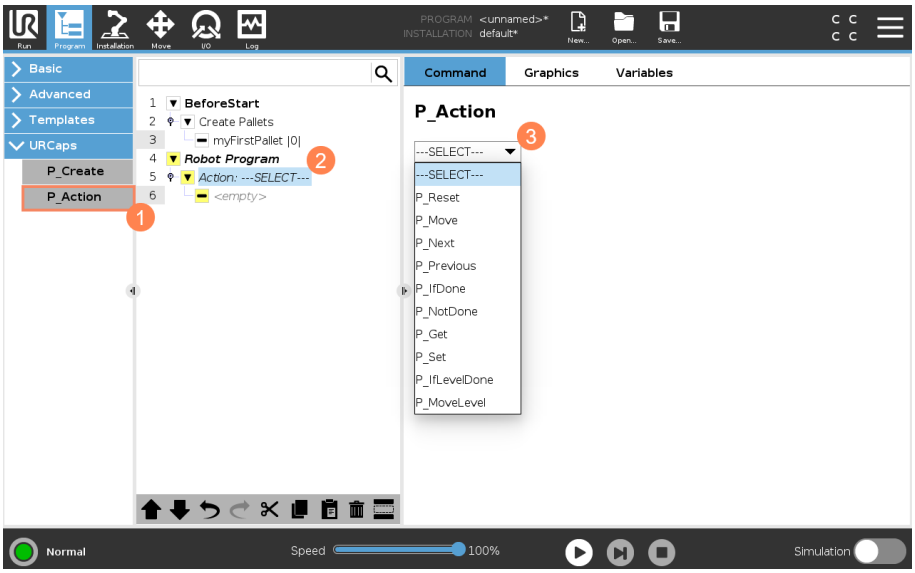


Figure 27: Program Node P_Action

1. Insert P_Action
2. Mark the node
3. Choose an action

5.2 P_Reset: Pallet reset

P_Reset resets a pallet to the starting position. You can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0 - 19.

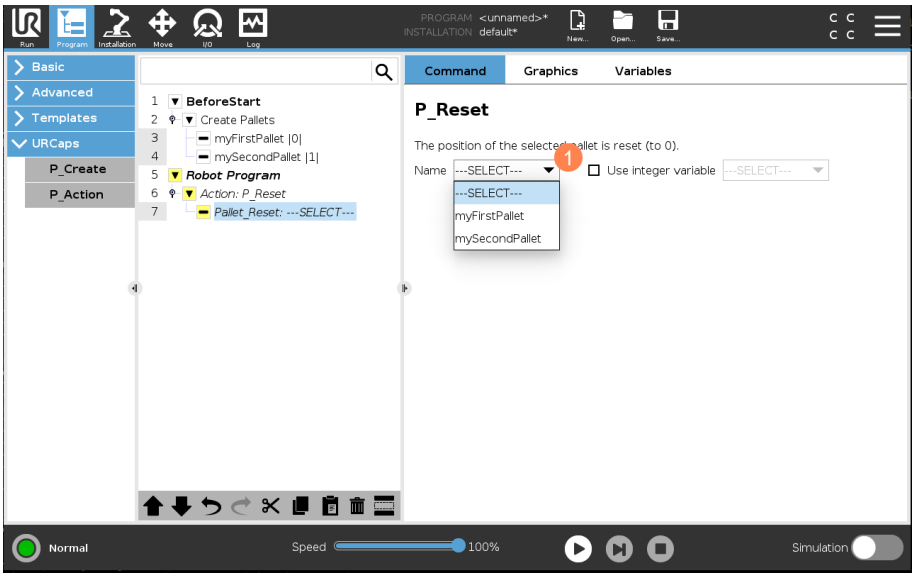


Figure 28: P_Reset called by name

1. Select the name of the pallet

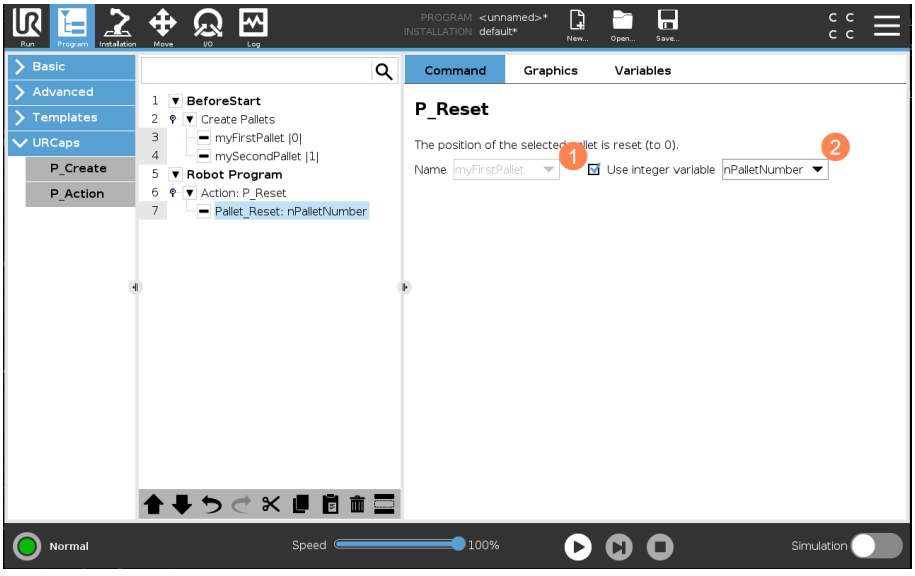


Figure 29: P_Reset call via integer variable

1. Check the box in front of Use Integer Variable
2. Select an integer variable

5.3 P_Move: Approach pallet position

P_Move performs a robot movement. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

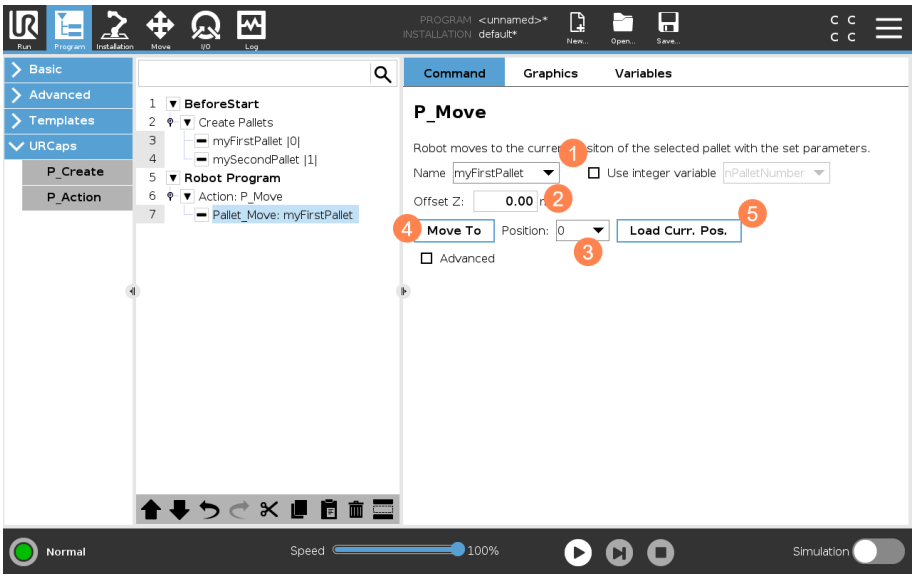


Figure 30: P_Move standard

1. Select the pallet either by name or by variable
2. If you enter an offset in the Z direction, the robot moves over the current position
3. Here you can select a position on the pallet for manual testing of the pallet and move to it with Move To
4. Here you can approach the selected position
5. Here you can read out the current position of the pallet and display it in the text field

Note: Points 3 - 5 are only intended for manual testing of the pallet and have no effect on the subsequent automatic process. Points 3 - 5 can only be carried out if the pallet has been selected by name.

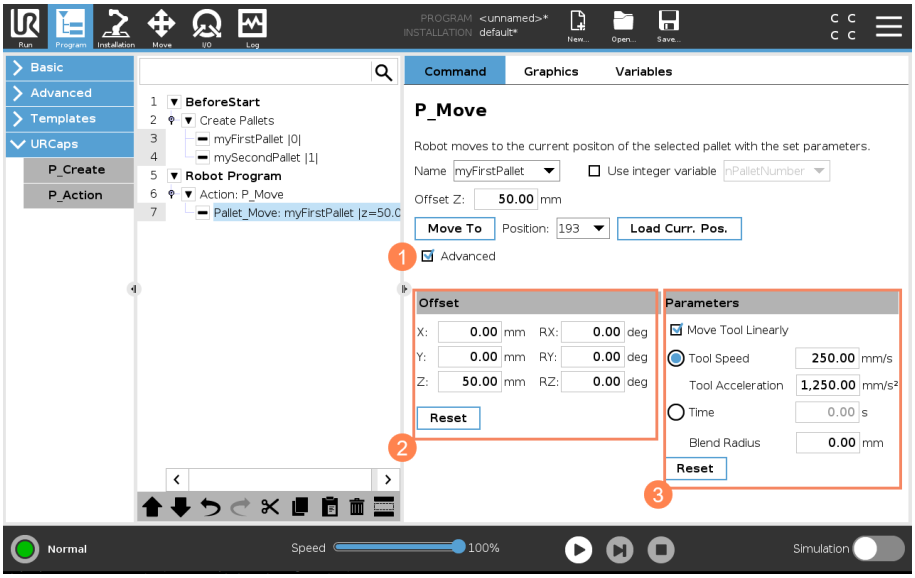


Figure 31: P_Move advanced

1. The advanced settings can be opened with Advanced
2. The offsets for the current position can be specified under Offset
3. The parameters for the robot movement are specified under Parameters

5.4 P_Next: Increase pallet counter

P_Next increases the pallet counter by one position. Another step size can also be entered under Advanced. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

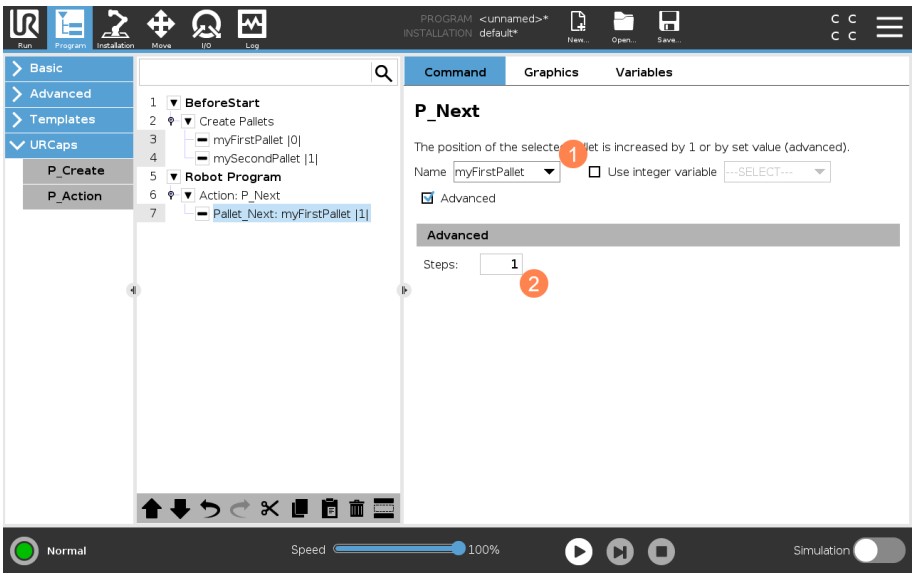


Figure 32: P_Next

1. Select the pallet either by name or by variable
2. The standard step size is 1 and can be adjusted here

5.5 P_Previous: Decrease pallet counter

P_Previous decreases the pallet counter by one position. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

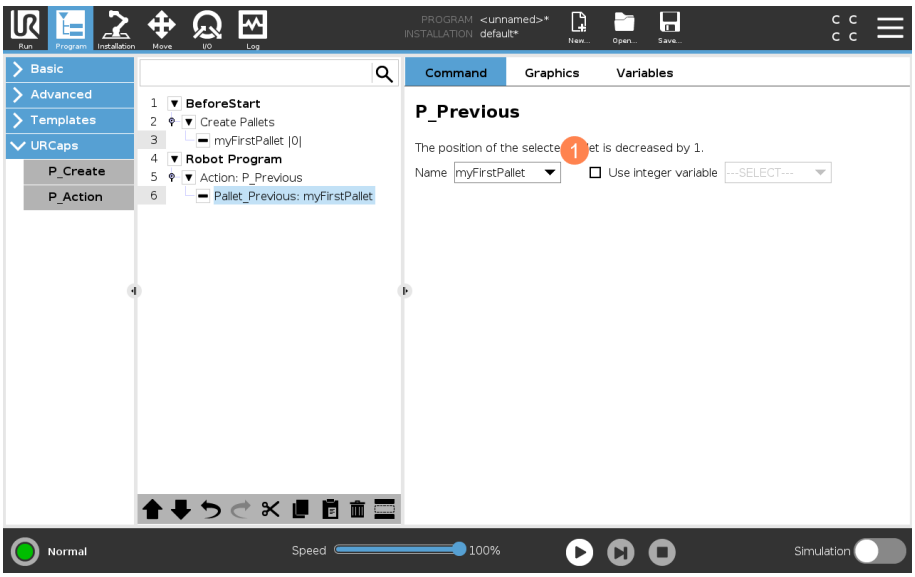


Figure 33: P_Previous

1. Select the pallet either by name or by variable

5.6 P_IfDone: Query whether the pallet has been processed

P_IfDone checks whether the current pallet has been processed. If this is the case, the sub-node is called and the instructions programmed in it are executed. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

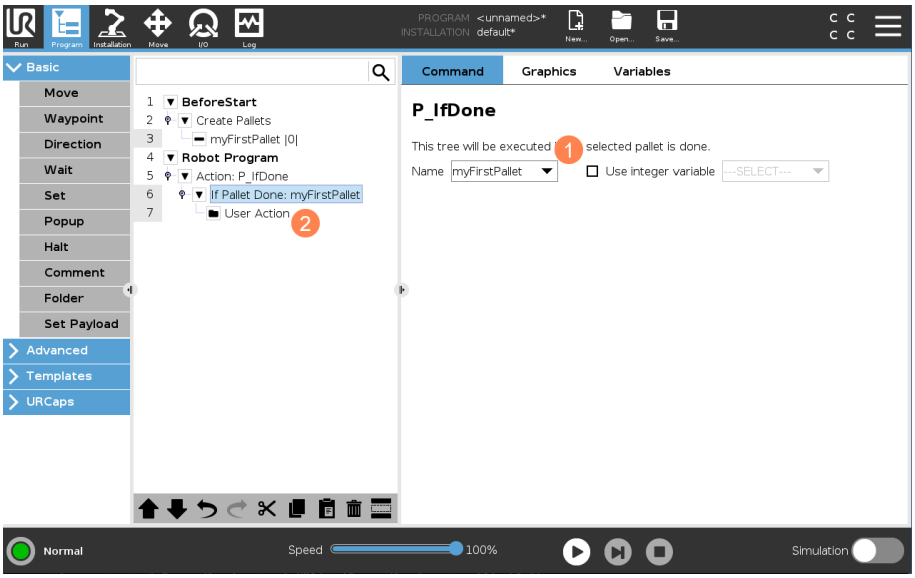


Figure 34: P_IfDone

1. Select the pallet either by name or by variable
2. Insert the desired program code in the child node

5.7 P_NotDone: Query whether the pallet is still being processed

P_NotDone is the counterpart to P_IfDone and checks whether the current pallet has not yet been processed. If this is the case, the child node is called and the instructions programmed in it are executed. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

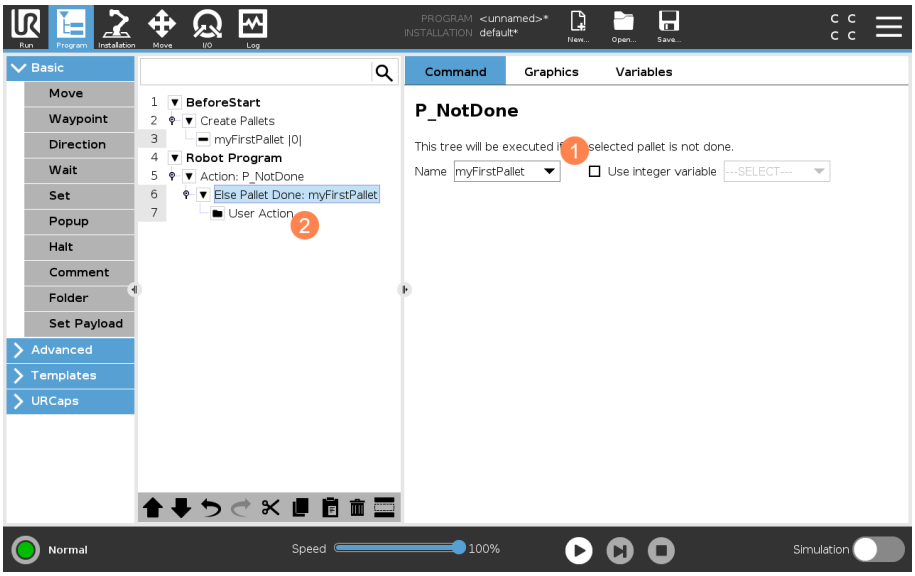


Figure 35: P_NotDone

1. Select the pallet either by name or by variable
2. Insert the desired program code in the child node

5.8 P_Get: Read out current pallet position

P_Get can output the current total counter or the current X, Y or Z position. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

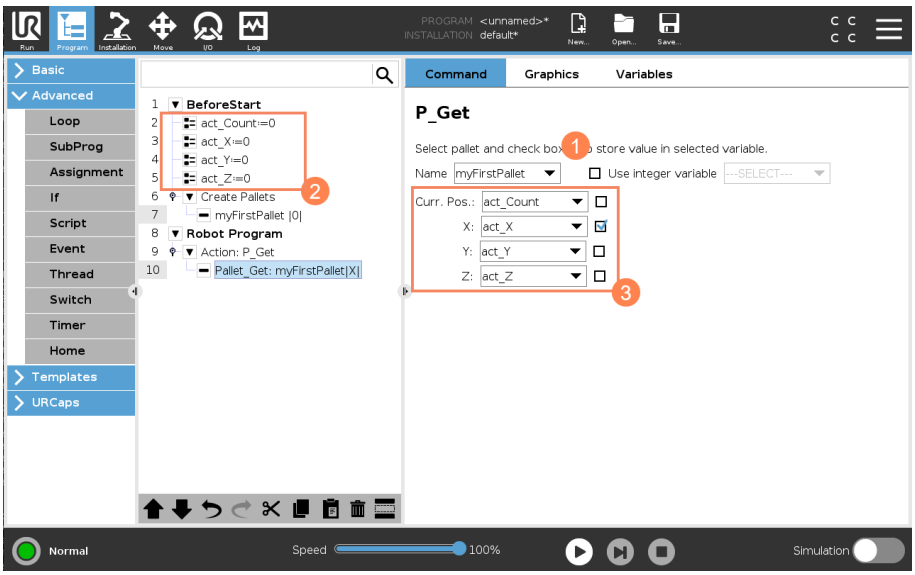


Figure 36: P_Get

1. Select the pallet either by name or by variable
2. Create a local or global integer variable
3. In the drop-down field you can select an integer variable to be written with the current pallet value. The action is only carried out in the automatic sequence if the checkbox is ticked

5.9 P_Set: Write pallet position

P_Set can reset either the total counter or the X, Y or Z position of a pallet. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

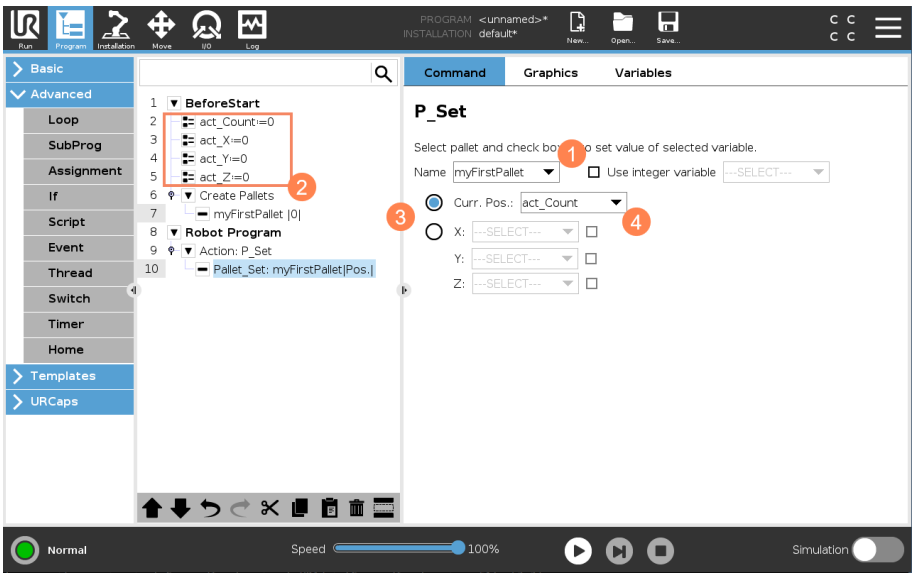


Figure 37: P_Set total counter

1. Select the pallet either by name or by variable
2. Create a local or global integer variable
3. Choose between Curr. Pos. (Total counter) or XYZ
4. You can select an integer variable in the drop-down field. The value of this variable is now set as the current pallet position. The variable must not be negative or greater than the maximum number of workpieces. If, however, a value that is too small or too large is entered, the pallet position is set to 0.

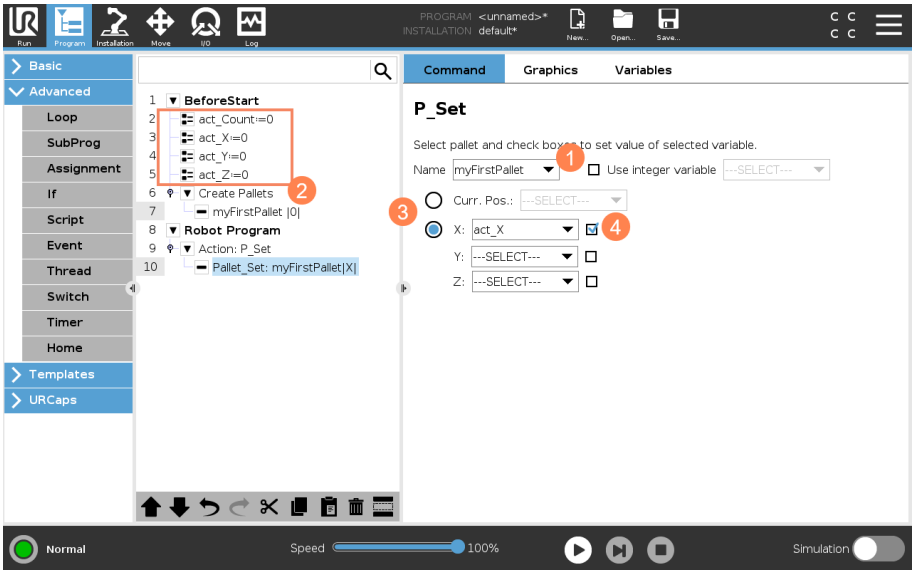


Figure 38: P_Set XYZ

1. Select the pallet either by name or by variable
2. Create a local or global integer variable
3. Choose between Curr. Pos. (Total counter) or XYZ
4. In the dropdown field you can select an integer variable and tick the checkbox. The value of this variable is now set as the current X, Y or Z position. The variable must not be negative or greater than the maximum number of workpieces in this direction. If, however, a value that is too small or too large is entered, the position is not changed and the old value is retained

5.10 P_IfLevelDone: Query whether the current level has been processed

P_IfLevelDone checks whether the current level of the pallet has been processed. If this is the case, the child node is called and the instructions programmed in it are executed. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

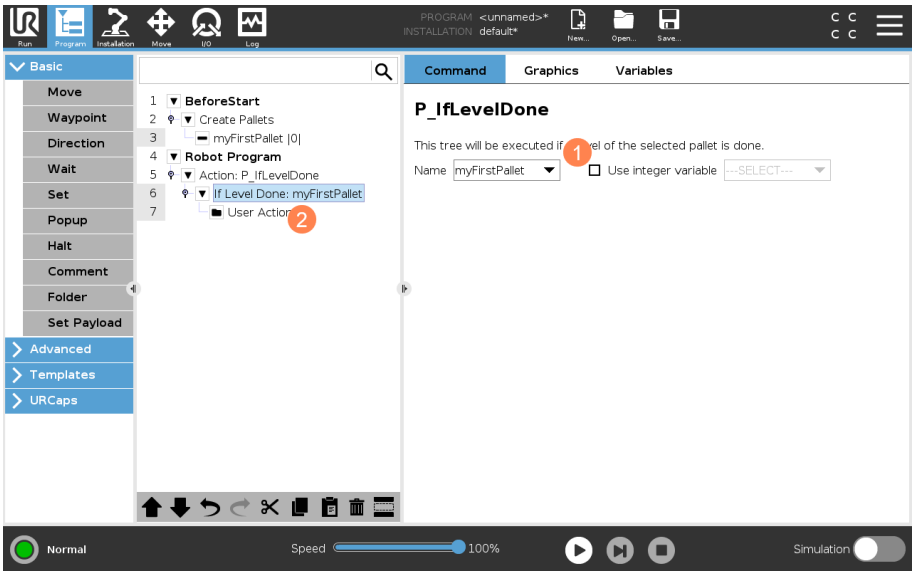


Figure 39: P_IfLevelDone

1. Select the pallet either by name or by variable
2. Insert the desired program code in the child node

5.11 P_MoveLevel: Approach the intermediate layer

P_MoveLevel performs a robot movement. Here, too, you can either select the pallet by name or address the pallet with an integer variable. It is important that the variable can only contain the values 0-19.

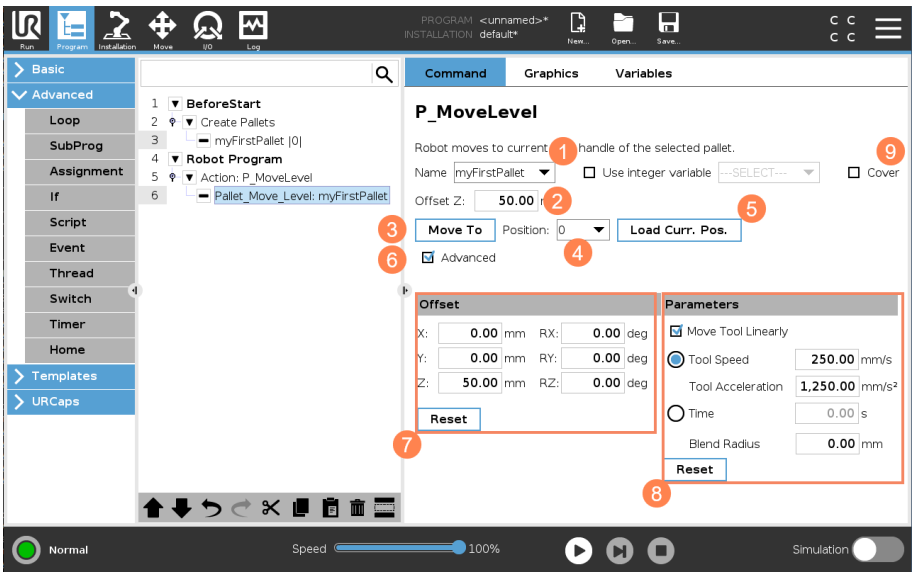


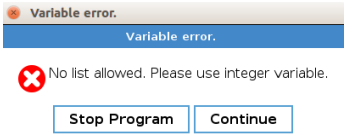
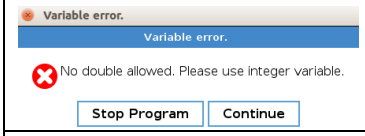
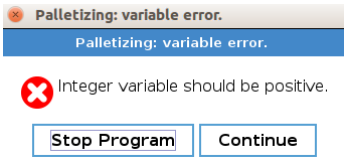
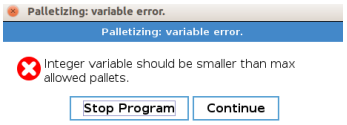
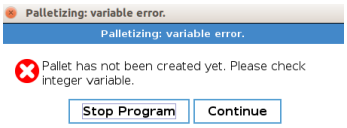
Figure 40: P_MoveLevel

1. Select the pallet either by name or by variable
2. If you enter an offset in the Z direction, the robot moves over the current position
3. Here you can select a position on the pallet for manual testing of the pallet and move to it with Move To. The robot then automatically calculates the current level and approaches it
4. Here you can approach the selected position
5. Here you can read out the current position of the pallet and display it in the text field

Note: Points 3 - 5 are only intended for manual testing of the pallet and have no effect on the subsequent automatic process. Points 3 - 5 can only be carried out if the pallet has been selected by name.

6. The advanced settings can be opened with Advanced
7. The offsets for the current position can be specified under Offset
8. The parameters for the robot movement are specified here
9. If Cover is selected, the robot always moves to the top level, regardless of where the pallet counter is currently. This selection is only required if, for example, the robot only has one Z-plane, but has to put down the pallet and a cover pallet.

6 Troubleshooting

Problem	Possible Cause	Solution
Variable error. 	In the program node P_Action, a list was transferred with "use integer variable"	Check the variable and replace it with an integer variable
Variable error. 	In the program Node P_Action, a double variable was transferred for "use integer variable".	Check the variable and replace it with an integer variable
Variable error 	In the program Node P_Action, an integer variable was transferred with "use integer variable", which is negative.	The valid range of values for the pallet numbers is from 0-19
Variable error 	In the program Node P_Action, an integer variable greater than 19 was transferred for "use integer variable".	The valid range of values for the pallet numbers is from 0-19
Variable error 	In the program Node P_Action, an integer variable was transferred for "use integer variable" where the pallet has not yet been created.	Adjust the value of the variable. Or create a new pallet under P_Create

7 Directories

7.1 List of figures

Figure 1: Home screen 6
Figure 2: Select Settings 6
Figure 3: Add URCap 7
Figure 4: Select URCap on USB stick 7
Figure 5: Restart the robot 8
Figure 6: URCap is installed 8
Figure 7: Select URCap 9
Figure 8: Installation Node 10
Figure 9: Generator string 10
Figure 10: Enter license key 11
Figure 11: Successful licensing 11
Figure 12: Online help 11
Figure 13: Program Node P_Create 12
Figure 14: PopUp P_Create 12
Figure 15: P_Create Reset 13
Figure 16: Create a new pallet 13
Figure 17: Copy a pallet 14
Figure 18: Delete a pallet 14
Figure 19: Pallet values 15
Figure 20: Example pallet 16
Figure 21: Example pallet 1 has 3 workpieces in X and 1 each in Y and Z. 17
Figure 22: Example pallet 2 has 1 workpiece in X and 3 each in Y and Z. 17
Figure 23: Example pallet 2 has 3 workpieces each in X, Y and Z. 17
Figure 24: Offset grid 18
Figure 25: Offset pallet configuration 18
Figure 26: Script Code 19
Figure 27: Program Node P_Action 20
Figure 28: P_Reset called by name 21
Figure 29: P_Reset call via integer variable 21
Figure 30: P_Move standard 22
Figure 31: P_Move advanced 23
Figure 32: P_Next 24
Figure 33: P_Previous 25
Figure 34: P_IfDone 26
Figure 35: P_NotDone 27
Figure 36: P_Get 28
Figure 37: P_Set total counter 29
Figure 38: P_Set XYZ 30
Figure 39: P_IfLevelDone 31
Figure 40: P_MoveLevel 32

8 Notes