

# eLearning Program Template



## Cabinet Door Template Explained

ROBOTIQ

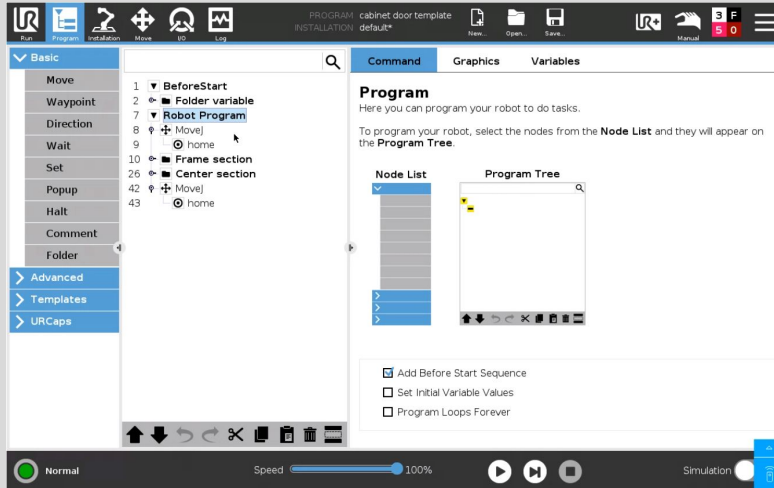
# PROGRAM TEMPLATE FOR CABINET DOOR FINISHING





# Description

Use this document to help you program your application using the template. You can learn more about the steps on how to program the Surface Finishing Kit for a first application using the available video for this course. Visit [support.robotiq.com](https://support.robotiq.com) for more details.



## What you will need

- Robotiq Surface Finishing Kit
- Orbital sander
- Latest URcap - Finishing Copilot
- Universal Robot UR5 or UR10
- Cabinet doors
- Program Template: cabinet door template.urp

This template can be used to finish **cabinet doors** of various sizes:

- This template is divided in two sections: the frame and the center.
- The reference waypoints are taught at first to create a model.
- When running the program, the operator is asked to manually enter the size of the part and the program adjusts to these new dimensions.
- This template can be modified to fit your own application. It is a basis from which to start.
- We do assume that the user is somewhat familiar with the **Force Control** node as well as the **Path Generator**. For more details, please consult the tutorials on [Robotiq eLearning](#).



# Program Layout

PROGRAM: cabinet door template  
INSTALLATION: default\*

Run Program Installation Move I/O Log

New... Open... Save...

Manual

3 F  
5 0

Basic

Move  
Waypoint  
Direction  
Wait  
Set  
Popup  
Halt  
Comment  
Folder

Advanced  
Templates  
URCaps

Command Graphics Variables

1 BeforeStart  
2 Folder variable  
7 Robot Program  
8 MoveJ  
9 home  
10 Frame section  
26 Center section  
42 MoveJ  
43 home

Program

Here you can program your robot to do tasks.

To program your robot, select the nodes from the **Node List** and they will appear on the **Program Tree**.

Node List

Program Tree

Add Before Start Sequence  
 Set Initial Variable Values  
 Program Loops Forever

Normal Speed 100% Simulation

1 The program starts with a **BeforeStart** sequence where the variables are set up.

2 The **Robot Program** section is composed of a pre-routine move commands and two folders: the **Frame section** and the **Center section**

# About Folders ...

The **Folder** node is a very useful tool when programming on Universal Robots:

- By suppressing/unsuppressing them, the user can run small sections of the program without having to run the whole program.
- It can be used to copy and paste parts of the program without re-writing everything.



# Folder Variable

PROGRAM: cabinet door template  
INSTALLATION: default\*

Basic

Move  
Waypoint  
Direction  
Wait  
Set  
Popup  
Halt  
Comment  
Folder

Advanced  
Templates  
URCaps

1 BeforeStart  
2 Folder variable  
3  
4 part\_width=Enter part width  
5 part\_length=Enter part length  
6 top\_frame\_width=Enter top/bottom  
7 side\_frm\_width=Enter side frame  
7 Robot Program  
8 MoveJ  
9 home  
10 Frame section  
26 Center section  
42 MoveJ  
43 home

Command Graphics Variables

Program  
Here you can program your robot to do tasks.  
To program your robot, select the nodes from the **Node List** and they will appear on the **Program Tree**.

Node List  
Program Tree

Add Before Start Sequence  
 Set Initial Variable Values  
 Program Loops Forever

Normal Speed 100% Simulation

3

In this **Folder**, the user is asked to enter the dimensions of the part when the program **Starts (standard mode)**.

These variables are used in the **Path Generator** nodes to determine the motions.

The value of these variables could also be set using different means with custom programming:

- With the touch of the robot
- External measuring device (ex. laser sensor)
- External PC or PLC
- etc.



# Robot Program

PROGRAM: cabinet door template  
INSTALLATION: default\*

Run Program Installation Move I/O Log

New... Open... Save...

Manual 3 F 5 0

Basic Command Graphics Variables

Move  
Waypoint  
Direction  
Wait  
Set  
Popup  
Halt  
Comment  
Folder

Advanced  
Templates  
URCaps

1 ▼ BeforeStart  
2 ◂ Folder variable  
7 ▼ Robot Program 4  
8 ⚙ MoveJ  
9 ⓪ home  
10 ◂ Frame section  
26 ◂ Center section  
42 ⚙ MoveJ  
43 ⓪ home

Folder

A folder is simply a collection of program lines.  
Please enter text to be displayed in the program tree:

Folder variable

Hide Folder Program Tree

Normal Speed 100% Simulation

- 4 The **Robot Program** is composed of 4 main elements:
- The robot moves to its home position
  - The robot sends the frame of the door
  - The robot sends the center of the door
  - The robot moves back to the home position





# Frame Section

PROGRAM: cabinet\_door\_template  
INSTALLATION: default\*

Basic | Command | Graphics | Variables

### Force Control

Feature: Tool [Test]

Enable control	Force/Torque	Stiffness	Deviation range
x <input type="checkbox"/>	Fx 0 N	100 %	-100 100 mm
y <input type="checkbox"/>	Fy 0 N	100 %	-100 100 mm
z <input checked="" type="checkbox"/>	Fz 10 N	100 %	-100 100 mm
Rx <input type="checkbox"/>	Mx 0 Nm	100 %	-20 20 °
Ry <input type="checkbox"/>	My 0 Nm	100 %	-20 20 °
Rz <input type="checkbox"/>	Mz 0 Nm	100 %	-20 20 °

Enable adaptive stiffness  
 Apply force based on:  Targeted position  Current position

LEAN ROBOTICS | ROBOTIQ

Normal | Speed 100% | Simulation

- 5 In the frame section **Folder**, the first step is to move to the approach point slightly over the first point of the **4-Point Path**. Then the zeroing of the force sensor is done.
- 6 A **Force Control** node is used to apply a force in the Tool-Z+ direction. The force applied will depend on your process. A **Wait for force** node is used to detect when the contact is made with the surface, then the tool is started.



# Frame Section

PROGRAM: cabinet\_door\_template  
INSTALLATION: default\*

Run Program Installation Move I/O Log

New... Open... Save... Manual

3 F 5 0

Basic

Move  
Waypoint  
Direction  
Wait  
Set  
Popup  
Halt  
Comment  
Folder

Advanced  
Templates  
URCaps

1 BeforeStart  
2 Folder variable  
7 Robot Program  
8 MoveJ  
9 home  
10 Frame section  
11 MoveL  
12 approach\_frame  
13 Zero FT Sensor  
14 Force Control  
15 Wait for force  
16 Wait until force and torque  
17 Start tool  
18 4-Point Path (7)  
19 Point 1  
20 Point 2  
21 Point 3  
22 Point 4  
23 Stop tool  
24 MoveL  
25 retract\_frame (8)  
26 Center section

Command Graphics Variables

Path Generator

Path Surface Perimeter Dimensions Margins

Width  
 Fixed  
 Dynamic (1)  
Taught part W 356 mm  
Current part A part\_width (1)

Length  
 Fixed  
 Dynamic (1)  
Taught part L 711 mm  
Current part B part\_length (1)

ROBOTIQ

Normal Speed 100% Simulation

7 A **4-Point Path** node is used to sand the frame of the door. The **Perimeter Only** mode is used for the frame and the dimensions are dynamic. It uses the variables that were previously set in **Folder Variable**.

8 Once all the points are taught, a **relative move** in the Z direction is added. This is to move away from the part once the operation is complete.



# Center Section

PROGRAM: cabinet door template  
INSTALLATION: default\*

Run Program Installation Move I/O Log

Command Graphics Variables

### Force Control

Feature: Tool [Test]

Enable control	Force/Torque	Stiffness	Deviation range
x <input type="checkbox"/>	Fx 0 N	100 %	-100 100 mm
y <input type="checkbox"/>	Fy 0 N	100 %	-100 100 mm
z <input checked="" type="checkbox"/>	Fz 10 N	100 %	-100 100 mm
Rx <input type="checkbox"/>	Mx 0 Nm	100 %	-20 20 °
Ry <input type="checkbox"/>	My 0 Nm	100 %	-20 20 °
Rz <input type="checkbox"/>	Mz 0 Nm	100 %	-20 20 °

Enable adaptive stiffness  
Apply force based on:  Targeted position  Current position

LEAN ROBOTICS ROBOTIQ

Normal Speed 100% Simulation

- 9 Similar to the frame, the **Center section** uses an approach point. This time however the approach point should be over the center section above the first waypoint of the **4-Point Path**.
- 10 A **Force Control** node is used to apply a force in the Tool-Z+ direction. The force applied will depend on your process. A **Wait for force** node is used to detect when the contact is made with the surface, then the tool is started.



# Center Section

The screenshot displays the ROBOTIQ software interface. On the left, a tree view shows the robot program structure, with a '4-Point Path' node highlighted and circled with the number 11. The main window is titled 'Path Generator' and has tabs for 'Path', 'Surface', 'Perimeter', 'Dimensions', and 'Margins'. The 'Margins' tab is active, showing four margin settings (M1, M2, M3, M4) with input fields set to '0' mm and dropdown menus for 'set with' options. To the right, a 3D model of a door frame is shown with four points (1, 2, 3, 4) and four margin lines (M1, M2, M3, M4) extending from the points to the frame edges. The bottom status bar shows 'Normal' mode, 'Speed' at 100%, and 'Simulation' mode.

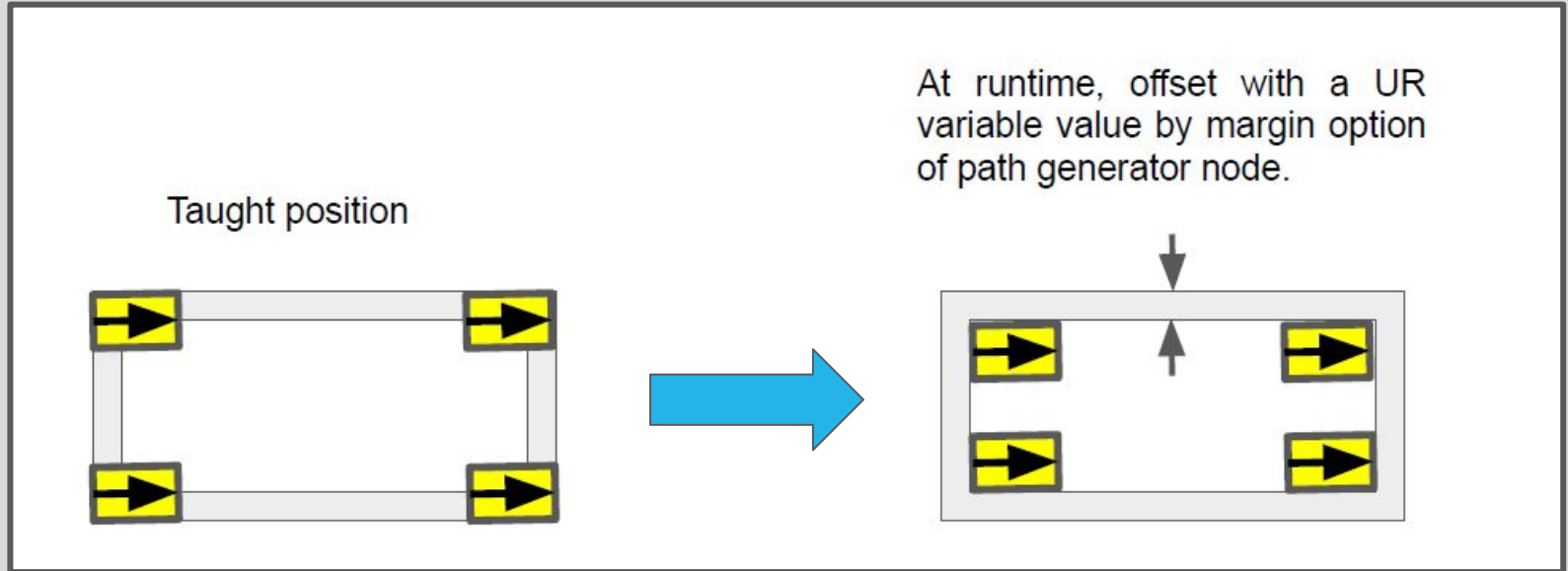
**11** A **4-Point Path** node is added to sand the center section of the door. Both the **Perimeter and Surface** options are enabled. The dimensions are dynamic and use the same variables as for the frame section.

This time, margins are added. We use variables to set the size of these margins which corresponds to the **size of the frame of the door**.



## Center Section

**Margins** are used to shift the waypoints over the center part of the door. When the 4 points of the **Path Generator** are taught, they need to be taught at the extremity of the part. The margins will move these positions over the center.





# Center Section

The screenshot shows the ROBOTIQ software interface. The main window displays a robot program with the following structure:

- 7 Robot Program
- 8 MoveJ
- 9 home
- 10 Frame section
- 26 Center section
- 27 MoveL
- 28 approach\_center
- 29 Zero FT Sensor
- 30 Force Control
- 31 Wait for force
- 32 Wait until force and torque
- 33 Start tool
- 34 4-Point Path (12)
- 35 Point 1
- 36 Point 2
- 37 Point 3
- 38 Point 4
- 39 Stop tool
- 40 MoveL
- 41 retract\_center (14)
- 42 MoveJ
- 43 home

The 'Path Generator' window is open, showing the 'Margins' tab. The margins are set to 0 mm, and the 'set with' dropdown is set to 'top...' for M1, M2, M3, and M4. The 3D model shows the part with four points (1, 2, 3, 4) and four margins (M1, M2, M3, M4).

**12** Teach the 4 points at the **4 corners of the part**. Leave a few millimeters of space near the edge of the part so that the weaving motion does not hit the border of the frame when sanding the lower center part.

**13** Margins are selected in the **Margins Tab**. Top and Side margins are set to the variables:  
**Top\_Frame\_width**  
**Side\_Frame\_width**

**14** A relative move is used to move away from the part once the process is complete.



# Center Section

The screenshot shows the Robotiq software interface. The top menu bar includes 'Run', 'Program', 'Installation', 'Move', 'I/O', and 'Log'. The main window is divided into a left sidebar with a 'Basic' tab, a central program tree, and a right-hand 'Command' panel. The program tree shows a 'Robot Program' with a 'Center section' folder. A 'Move' command is being added to this folder, indicated by a circled '15'. The 'Command' panel shows a 'Folder' section with a text input field containing 'Center section'. The bottom status bar shows 'Normal' speed, a 'Speed' slider at 100%, and a 'Simulation' button.

**15** A Move is finally added to move the robot back to its starting position.



# At Runtime

The screenshot shows the Robotiq software interface during runtime. The main window displays a program tree on the left, with '4-Point Path' selected. The 'Waypoint' assignment window is open, showing a numeric keypad with the number '16' entered. The background shows a program tree with '4-Point Path' selected. The status bar at the bottom indicates 'Running' and 'Speed 100%'.

**16** **Assignment windows** will show up when the program start. The operator has to enter the dimensions of the current parts.

The program will adjust to these dimensions.





**More Templates Available!**



**ROBOTIQ**  
**eLearning**

[elearning.robotiq.com](http://elearning.robotiq.com)

