

The Pick-it URCap interface

Overview

Pick-it integrates seamlessly with Universal Robots by means of a URCap plugin. This plugin exposes a set of Pick-it specific command blocks that make the creation of vision-guided programs simple and easy. This article documents the interface of the Pick-it URCap plugin. For installation instructions please refer to the [Getting started with the Pick-it URCap \(//support.pickit3d.com/article/75-getting-started-with-the-pick-it-urcap\)](https://support.pickit3d.com/article/75-getting-started-with-the-pick-it-urcap) article.

- 1 Global variables (#global_variables)
- 2 Commands (#commands)
- 3 Helper functions (#helper_functions)

Global variables

To use one of the global variables defined by the Pick-it plugin, you need to declare it in the **BeforeStart** section of the program. You only need to declare the variables you actually use in your program.

```
pickit_pose
```

Pick pose for the latest detection results sent by Pick-it, represented as a 6D array.

The sequence of commands to get a valid and reachable pick pose is:

1. Declare the variable in the **BeforeStart** section of the program.

```
▼ BeforeStart
├─ pickit_pose:=p[0,0,0,0,0,0]
├─ pickit_pre_pose:=p[0,0,0,0,0,0]
```

2. In the **Robot Program** section, send a request to Pick-it using **Find object(s)** or **Get next object** command.
3. Wait for detection result using **Get Result** command.
4. Check that a valid detection is available by calling `pickit_object_found()`.
5. Check that the robot can reach the pose by calling `pickit_object_reachable()`.

Refer to the [Universal Robots URcap example program \(//support.pickit3d.com/article/76-universal-robots-urcap-example-program\)](https://support.pickit3d.com/article/76-universal-robots-urcap-example-program) article for a detailed usecase description.

pickit_pre_pose

Pre-pick pose for the latest detection results sent by Pick-it, represented as a 6D array.

The sequence of commands to get a valid and reachable pre-pick pose is the same as for `pickit_pose`. Refer to the documentation of `pickit_pose` for more details. For details on how `pickit_pre_pose` is computed from `pickit_pose`, refer to the documentation of the **Get result** command.

pickit_dim

Object dimensions, in meters, for the latest detection results sent by Pick-it, represented as a 3D array.

This is how to interpret the array according to the object type:

- **Square** [length, length, 0]
- **Rectangle** [length, width, 0]
- **Circle** [diameter, diameter, 0]
- **Ellipse** [length, width, 0]
- **Cylinder** [length, diameter, diameter]
- **Sphere** [diameter, diameter, diameter]
- **Point cloud (Pick-it Teach model)** [bbox x, bbox y, bbox z]
- **Blob** [bbox x, bbox y, bbox z]

where `bbox x` means the size of the bounding box along the x-axis.

The sequence of commands to get valid object dimensions is the same as for `pickit_pose`. Refer to the documentation of `pickit_pose` for more details.

pickit_type

Object type, as an integer identifier, for the latest detection results sent by Pick-it.

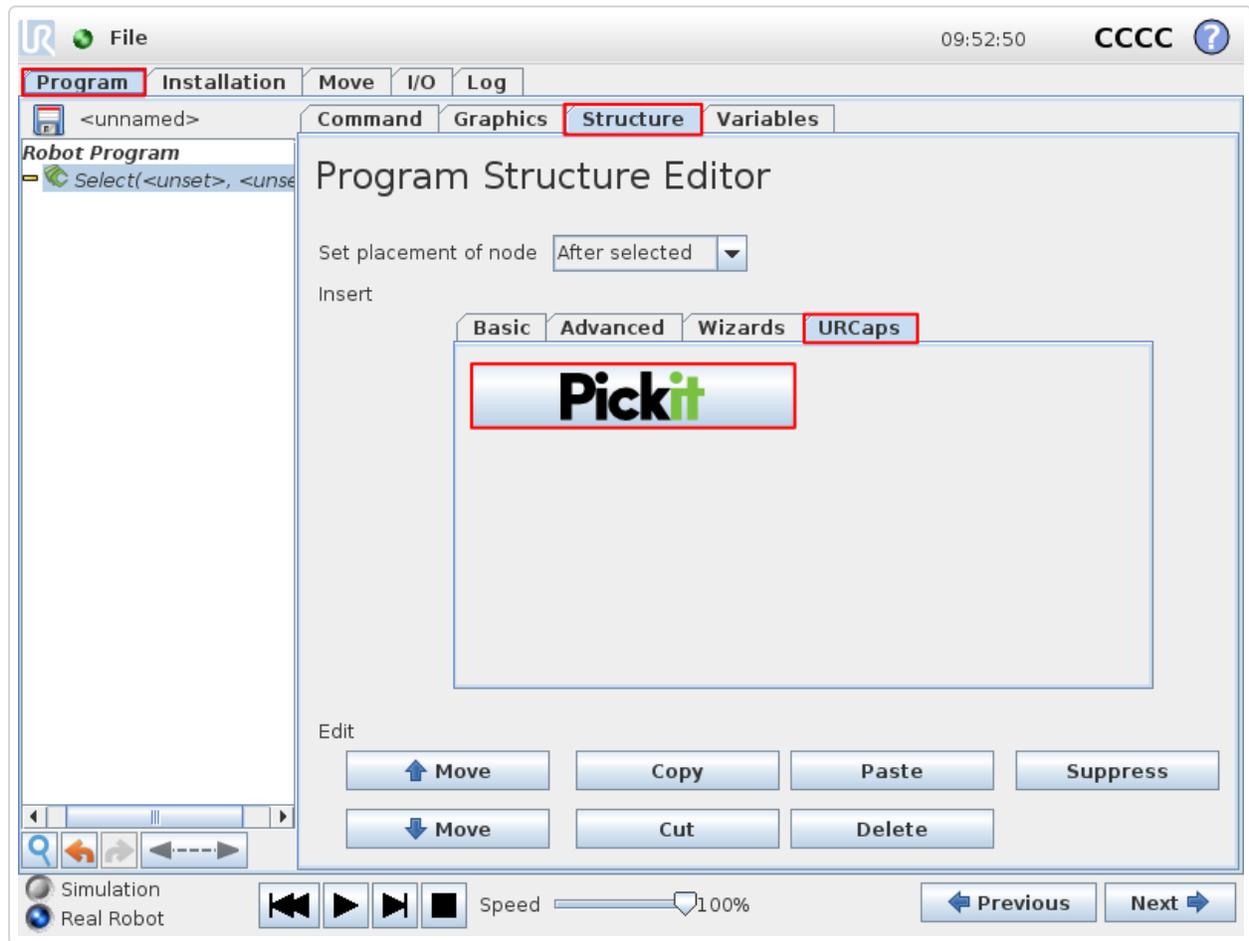
The mapping between object shape and identifier follows:

- **Square** 21
- **Rectangle** 22
- **Circle** 23
- **Ellipse** 24
- **Cylinder** 32
- **Sphere** 33
- **Point cloud (Pick-it Teach model)** 34
- **Blob** 50

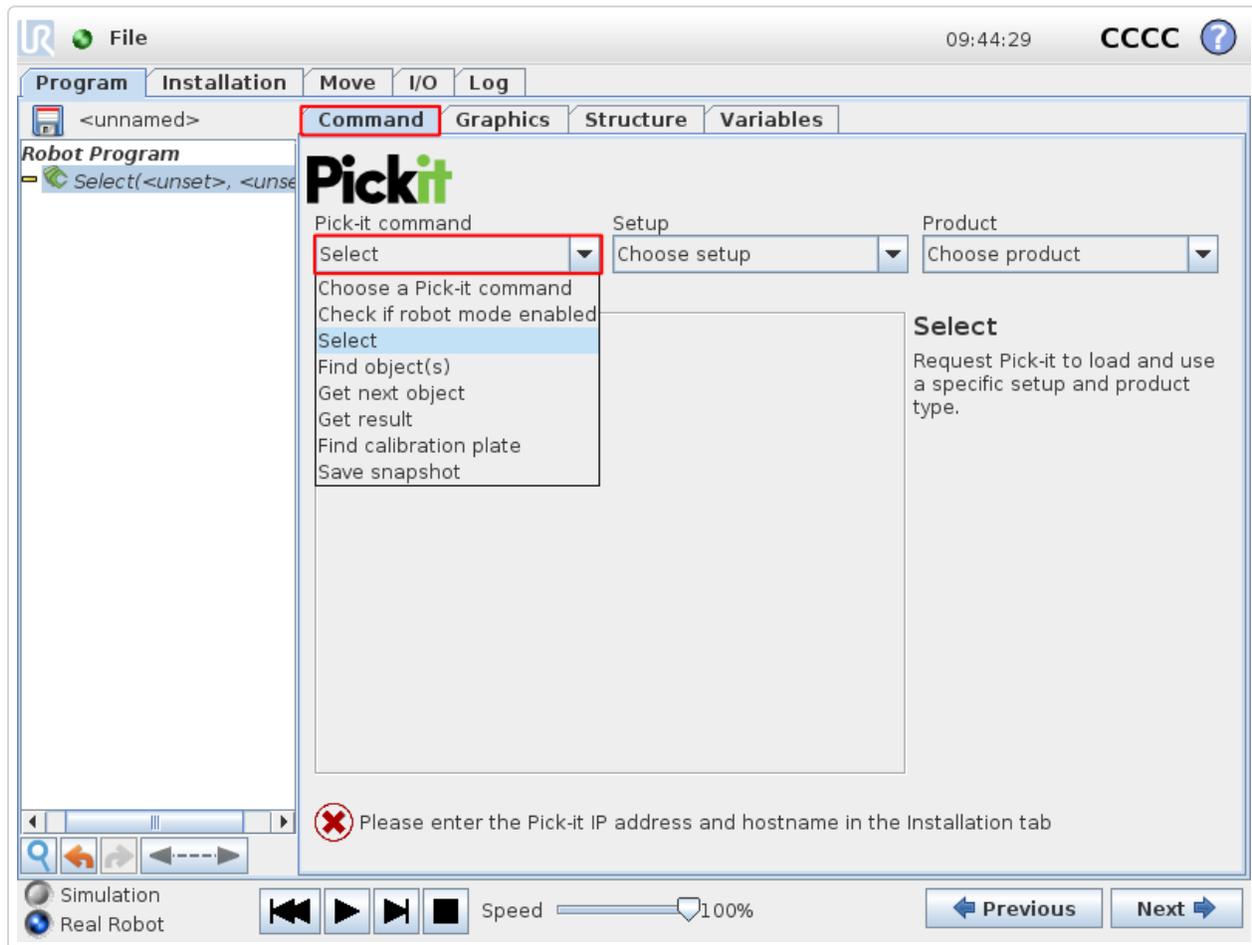
The sequence of commands to get a valid object type is the same as for `pickit_pose`. Refer to the documentation of `pickit_pose` for more details.

Commands

The Pick-it plugin provides a set of commands that add to the set of Polyscope's existing commands. They can be accessed from within the **Program** tab, under **Structure** → **URCaps**, and clicking the **Pick-it** button.



To insert a new command, navigate to the **Command** tab select an entry from the **Pick-it command** drop-down. When a command is selected, a brief description is shown on the rightmost part of the page. Some commands require the specification of input parameters.



Check if robot mode enabled

Checks whether robot mode is enabled in Pick-it.

- If robot mode is enabled, program execution continues
- If robot mode is not enabled, a pop-up is shown requesting the user to set Pick-it to robot mode.

Except for **Find calibration plate**, all other Pick-it plugin commands require robot mode to be Robot mode is enabled. Robot mode is enabled from the Pick-it web interface.

Select

Loads the specified setup and product configuration.

This configuration specifies the behavior of Pick-it detections, e.g. what to look for, in which part of the field of view.

Parameters

Setup Any of the setup configurations currently available in the connected Pick-it system.

Product Any of the product configurations currently available in the connected Pick-it system.

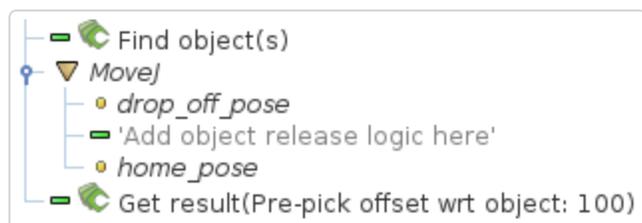
Available configurations are listed in drop-down menus.

Find object(s)

Trigger a Pick-it object detection using the currently active setup and product configuration.

The next Pick-it command after **Find object(s)** should always be **Get result**, which waits until a response for the detection request is ready.

Note that it's valid (and sometimes encouraged) to perform robot motions or other non Pick-it actions between calls to **Find object(s)** and **Get result**, for instance



Refer to the cycle time optimization section of the Universal Robots URCap example program ([//support.pickit3d.com/article/76-universal-robots-urcap-example-program](https://support.pickit3d.com/article/76-universal-robots-urcap-example-program)) article for the motivation behind performing robot motions while a Pick-it detection is.

Get next object

Request the next detected object.

A single call to **Find object(s)** might yield the detection of multiple objects. **Get next object** allows to request the next available object, if any, without the need of triggering a new detection and the time overhead it entails.

The next Pick-it command after **Find object(s)** should always be **Get next object**, which waits until a response for the request is ready.



```
Get next object
Get result(Pre-pick offset wrt object: 100)
```

It's recommended to use this command only when objects in the detection region have not moved (significantly) since calling **Find object(s)**. A good example of when to use **Get next object** is when a detection is unreachable by the robot. An example of when using **Get next object** is not ideal would be the following bin picking scenario:

- Trigger Pick-it detection that finds multiple objects.
- First object is picked. Since objects are randomly placed in bin, neighboring objects move and fall into place.
- Call **Get next object** and attempt to pick next object. If the next object is one of the neighboring parts that moved, the pick might fail.

When the objects in the detection region have moved, it's better to re-trigger **Find object(s)** instead.

Get result

Wait for Pick-it reply with detection results.

Get result should always be the next Pick-it command after a **Find object(s)** or **Get next object** request. It blocks until a reply from Pick-it is received, and the success of the request can then be queried by calling `pickit_object_found()`. When an object has been found, the following global variables are populated:

- Object pick pose: `pickit_pose`
- Object pre-pick pose: `pickit_pre_pose`.

This pose is computed by applying an offset to `pickit_pose` along a specified direction, as specified by the command parameters.

- Object dimensions: `pickit_dim`
- Object type: `pickit_type`

Parameters

Pre-pick offset: base frame `pickit_pre_pose` is computed by applying an offset along the z-axis of the specified frame. Valid options are object frame or robot base frame.

Pre-pick offset Offset in mm applied to compute `pickit_pre_pose`.

Find calibration plate

Trigger detection of the robot-camera calibration plate.

This command requires the Pick-it web interface to be in the Calibration page, hence robot mode should be disabled. When Pick-it is not in the calibration page, a pop-up is shown.

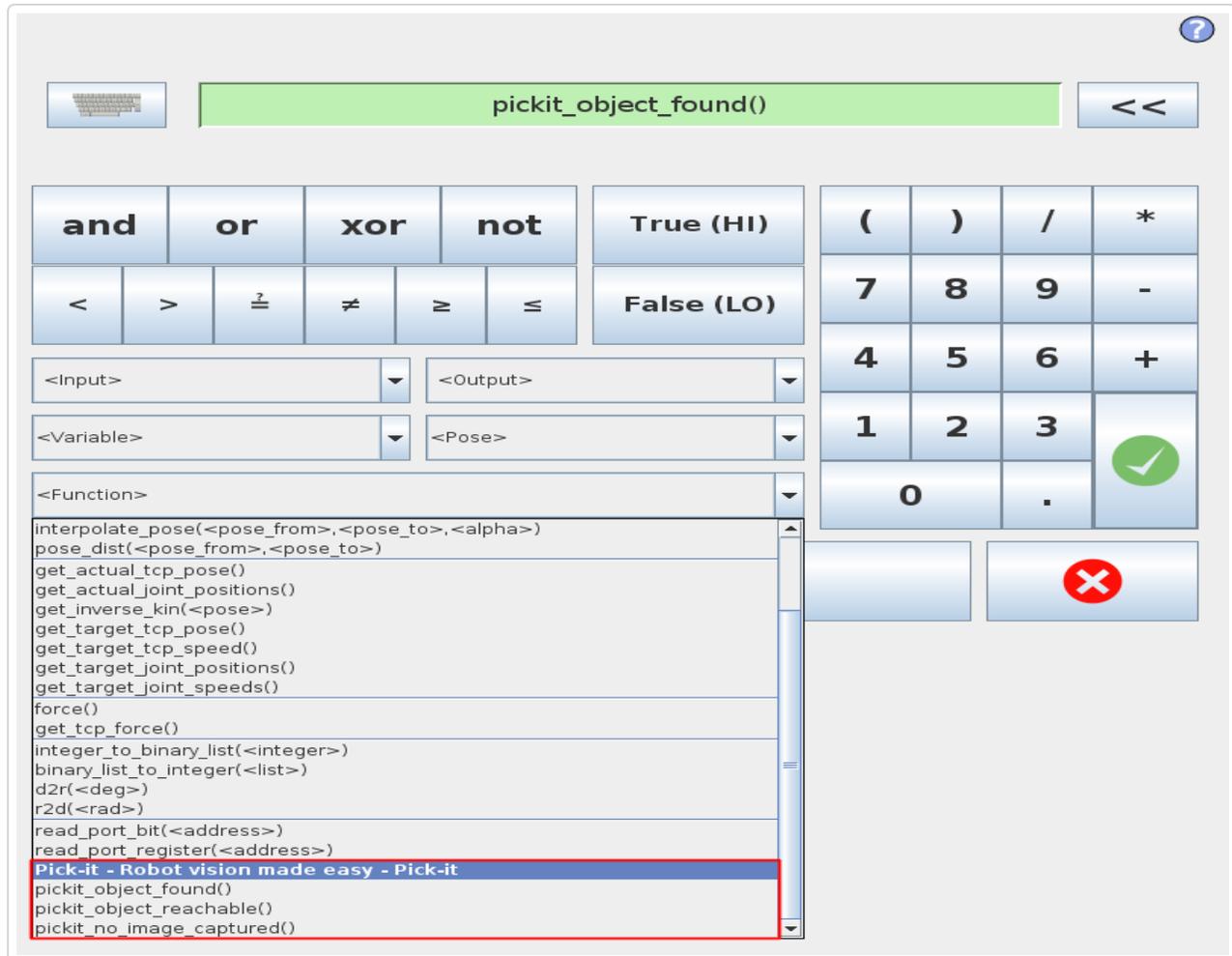
Save snapshot

Save a snapshot with the latest detection results.

The saved snapshot can then be loaded or downloaded by going to the Files page on the Pick-it web interface and searching for a file whose name contains the capture timestamp.

Helper functions

As opposed to commands, that don't have a return value; the Pick-it plugin also exposes a number of helper functions that return useful information. They typically are used as the expression of a conditional, such as an **if** statement, and can be selected from the **available functions drop-down**.



`pickit_object_found()`

Check if the last call to **Get result** produced valid detection results.

Return

True if detection results are available.

When results are available, the global variables `pickit_pre_pose` and `pickit_pose` have valid contents.

This function returns false when Pick-it replied with no detection results (nominal usecase); or if called without making a request to Pick-it and collecting the results with **Get result** (should be avoided, as it makes no sense).

`pickit_object_reachable()`

Check if the last call to **Get result** produced reachable pick and pre-pick poses.

Return

True if the global variables `pickit_pre_pose` and `pickit_pose` contain poses that are reachable by the robot.

Note that `pickit_object_reachable() == true` implies `pickit_object_found() == true`.

`pickit_no_image_captured()`

Check if object detection was unsuccessful due to a failure to capture a camera image.

When this is the case, it typically indicates a hardware disconnection issue, such as a loose connector or broken cable. This function can be used as trigger to send an alarm to a higher level monitoring system.

Return

True if object detection was unsuccessful due to a failure to capture a camera image.

 Still need help? [Contact Us \(/contact\)](/contact)

Last updated on February 1, 2018

© Pick-it (<http://support.pickit3d.com/>) 2018. Powered by Help Scout (https://www.helpscout.net/features/docs/?utm_source=docs&utm_medium=footerlink&utm_campaign=Docs+Branding)