



myP Cognex Module Manual

Version 1.4.3
(corresponds to the myP version)

2019-09-30

The information contained in this document is property of F&P Robotics AG and shall not be reproduced in whole or in part without prior written approval of F&P Robotics AG. The information provided herein is subject to change without notice and should not be constructed as a commitment by F&P Robotics AG. This manual is periodically reviewed and revised. F&P Robotics AG assumes no responsibility for any errors or omissions in this document.

This manual consists of original instructions from F&P Robotics AG and is intended for all users of myP. Before using the functions mentioned in this document together with myP and a P-Rob robotic arm or P-Grip robotic gripper, you must read the corresponding user manuals carefully, particularly paying attention to the safety instructions on the P- Rob user manual. Compliance with the instructions in all manuals is mandatory.

Copyright © 2011-2019 by F&P Robotics AG. All rights reserved.

The F&P logo as well as P-Rob®, P-Grip® and myP® are registered trademarks of F&P Robotics AG.

Contents

| | |
|---------------------------------------|----------|
| General Module Information | 4 |
| Initializing the Cognex Camera | 4 |
| Additional Script Functions | 5 |
| Cognex Functions | 5 |
| Connect to Camera | 5 |
| Trigger Camera | 5 |
| Change Insight Job | 6 |
| Get Camera Message | 6 |
| Disconnect Camera | 6 |
| Deprecated Functions | 7 |
| Get Format String | 7 |
| Send Native Command | 7 |

1 General Module Information

The **Cognex Camera Module** is an additional module for myP and is installed on a P-Rob using the default update procedure of myP. To access the full functionality of this module, a license file is provided together with the module when it is purchased. This license is easily installed onto your P-Rob by placing it together with a valid myP update onto a USB stick and plugging this stick into one of the free USB slots of P-Rob. For detailed information about the installation and update procedure, please refer to the [myP User Manual](#).

1.1 Initializing the Cognex Camera

Together with the physical camera module itself a Quick-Start-Guide has been delivered, which explains the details of how to mount and set up the camera with the [Insight-Explorer](#). For the remainder of this document it will be assumed that the following steps have been done and were successful:

1. Mounting the Camera Module to the Gripper
2. Connecting the Ethernet-Cables
3. Having a connection to the camera over the Insight-Explorer of Cognex
4. Having set up the IP's of camera and P-Rob such that they are in the same network

If all of this was successful, the camera will be powered whenever the robot is connected and has a fixed IP (default: **10.0.0.210**) which can be accessed whenever myP is connected to the P-Rob. When connecting to a camera, myP will always use the "Telnet-Port" of the camera and connect to it over a standard TCP-Socket. The socket will always try to open a connection to the port 3000, so please make sure that the "Telnet-Port" of the camera is not changed. It is also essential that the camera is set to online whenever myP tries to interact with the camera, as in case the camera being offline, only Insight-Explorer internal commands and events are valid.

2 Additional Script Functions

Once the Cognex Camera Module has properly been installed on your P-Rob, myP provides access to these additional script functions:

2.1 Cognex Functions

All Cognex related functions are called using the prefix `cognex.`

Connect to Camera

```
cognex.connect_to_camera(ip, user="admin", password="")
```

This function opens the TCP-Client on myP side and tries to establish a connection to the camera.

| <i>Parameter</i> | <i>Types</i> | <i>Description</i> |
|------------------|--------------|---|
| ip | string | IP address of the camera as a string, e.g. "10.0.0.210" |
| user | string | Choose the camera login name. This name can be set in the Insight-Explorer. "admin" is the Cognex default user. |
| password | string | Choose the camera password. This password can be set in the Insight-Explorer. The default password for "admin" is set as an empty string. |

Example

```
# connect to a camera with F&P Delivery IP and credentials  
cognex.connect_to_camera("10.0.0.210")
```

Connect to a freshly delivered camera with the default IP and credentials.

```
# connect to a camera with changed IP address and the following credentials  
# user: Developer  
# password: my_password_1234  
cognex.connect_to_camera("192.168.80.235", user="Developer", password="my_password_1234")
```

Connect to a camera with modified IP and access rights.

Trigger Camera

```
cognex.trigger_camera(ip)
```

Sets the camera Trigger event ("SW8") which takes a new picture on the camera.

| <i>Parameter</i> | <i>Types</i> | <i>Description</i> |
|------------------|--------------|---|
| ip | string | IP address of the camera as a string, e.g. "10.0.0.210" |

Change Insight Job

```
cognex.change_job(ip, job_name)
```

This function allows to change the Insight-Job of the camera. It works over the Native Cognex Command “LF”, which loads a job file stored on the camera. Note that this command can take several seconds to execute.

| <i>Parameter</i> | <i>Types</i> | <i>Description</i> |
|------------------|--------------|---|
| ip | string | IP address of the camera as a string, e.g. “10.0.0.210” |
| job_name | string | Name of the Job to execute |

Example

```
# change to a job with the name "FP_calibration.job"  
cognex.change_job("10.0.0.210", "FP_calibration")
```

Get Camera Message

```
cognex.get_camera_message(ip, event_id, keyword, timeout)
```

This function can be used to request a message sent over the Cognex “WriteMessage” function over native TCP connection. The event_id specified has to be linked to the “WriteMessage” command on Insight-side

| <i>Parameter</i> | <i>Types</i> | <i>Description</i> |
|------------------|--------------|---|
| ip | string | IP address of the camera as a string, e.g. “10.0.0.210” |
| event_id | int | Soft Event ID on Cognex Insight Side, which will be triggered by this command. Has to be linked to the desired “WriteMessage” on Cognex side to get the message |
| keyword | string | Keyword of the requested message to uniquely identify the message |
| timeout | int or None | Timeout in seconds after which the command raises if the requested keyword could not be found. can be “None” |

Example

```
# Requests the message, containing the string "my_message", linked to the  
# Event 1 (SW1), from the Insight Job.  
message = cognex.get_camera_message("10.0.0.210", 1, "my_message", None)
```

This function requests the camera message linked to the Cognex Event “SW1” and waits until the camera returns a string containing “my_message”.

Disconnect Camera

```
cognex.disconnect_camera(ip)
```

This function closes the TCP-Client to the camera on myP side.

| <i>Parameter</i> | <i>Types</i> | <i>Description</i> |
|------------------|--------------|---|
| ip | string | IP address of the camera as a string, e.g. “10.0.0.210” |

2.2 Deprecated Functions

Get Format String

This function is deprecated → use “get_camera_message” instead!

```
cognex.get_format_string_from_cell(ip, cell)
```

This function reads an Insight “FormatString” from a cell. If it is executed on a cell not containing such a FormatString, the function will raise an error.

| <i>Parameter</i> | <i>Types</i> | <i>Description</i> |
|------------------|--------------|---|
| ip | string | IP address of the camera as a string, e.g. “10.0.0.210” |
| cell | string | Cell identifier you would like to read from |

Send Native Command

This function is deprecated!

```
cognex.send_native_command(ip, command_string)
```

This function allows to execute any of the Cognex Insight Native Commands through myP.

| <i>Parameter</i> | <i>Types</i> | <i>Description</i> |
|------------------|--------------|---|
| ip | string | IP address of the camera as a string, e.g. “10.0.0.210” |
| command | string | Native Cognex command to execute |

Example

```
# Sends a native command which imitates the trigger_camera() command  
answer = cognex.send_native_command("10.0.0.210", "SW8")
```

This function sends the “SW8” command to the camera which does the same as trigger_camera() in this case, but can also be used for any other of the Cognex native commands.