Author Christian Müller
Date 06.11.2023
Revision 1.0

FLEXFEEDER OMRON PLUG-IN

Quickstart & Reference Guide

flexfactory AG
Giessenstrasse 15
CH-8953 Dietikon

Phone +41 44 774 55 66

application@flexfactory.com
flexfactory.com

INTRODUCTION

This document serves as a quick introduction to the setup and configuration of the flexfeeder STÄUBLI plug-in from flexfactory AG for the operation of a flexfeeder X with an OMRON robot.

Content:

1. System installation
2. Important robot positions
3. Checklist for preparing the flexfeeder
4. Installation and setup of the flexfeeder OMRON plug-in
5. Calibration
6. Productive operation
7. GUI [optional]
8. Troubleshooting

Note: Ideally, performing the calibration is only required once to determine the mapping between the coordinate systems of the camera (in pixels) and the robot (in mm).

For troubleshooting, please refer to section 8.

We recommend attending our 1-day flexfeeder training course. This training will provide you with valuable insights and extensive know-how to get the most out of our products.

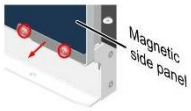
For more information and to register, please visit our training page at:
https://www.flexfactory.com/en/contact-and-support/trainings

How to connect the robot to the flexfeeder

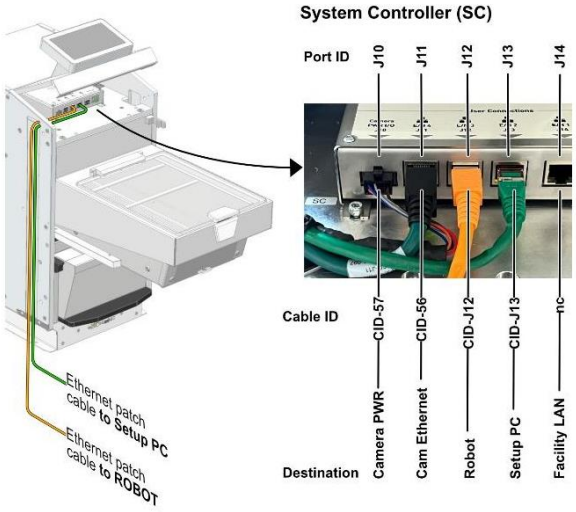
The flexfeeder controller provides Ethernet connections for the robot system. A simple patch cable allows the electrical and logical connection of the two systems.

How to install the Ethernet patch cables ?

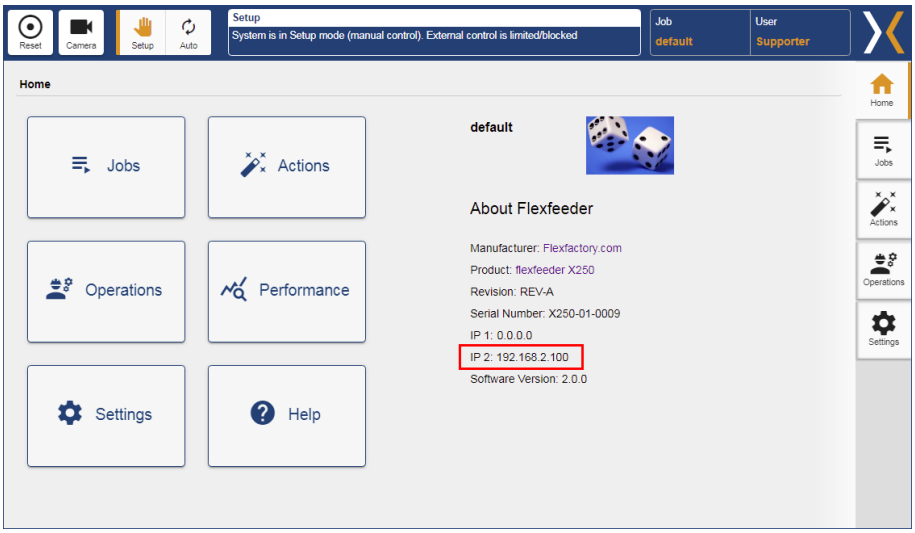
1. Remove the 2 screws and the plate as shown below:



2. Pull off the magnetic side panel
3. Pull the cable way cover up and out
4. Place the patch cables inside the cable way and plug them into the designated port
5. Insert the cable way cover from the top and push it all the way down to secure the cables
6. Install the magnetic side panel and secure it again.



The IP address of the flexfeeder can be found on the start page of the flexfeeder display:



2. IMPORTANT ROBOT POSITIONS

Pose out of sight:
Pose of the robot arm outside the camera's field of view during the calibration process.

Name of pose: ***cfg.out_of_view***

Attention! The gripper and any cables should also be out of the camera's field of view.

Pose in the center of the feeder surface:
Pose of the robot arm in the middle of the pick-up area of the flexfeeder.

In conjunction with the correct feeder model, this pose is used to calculate and set a monitored robot area. This monitored area is used in the multi-threaded example to realize an optimized triggering of the camera. This means that the camera is only triggered when the robot has left this monitored field of view of the camera. This area can be adjusted in the "ff.set_ff_model" program depending on the flexfeeder model.

Name of pose: ***cfg.pcenter***

Note: Only the X and Y coordinates of this robot pose are used to calculate the monitored robot area. However, the Z-coordinate is used as the pick-up height in the pick & place sequence.
This pose is also used to drop parts in the center of the feeder if there is already a part in the gripper at program start.

Poses for calibration:
The calibration object is placed in four different positions on the feeder surface. Set up these robot poses accordingly.

Pose names: **cfg.calpos[1]** – **cfg.calpos[4]**
-> index 0 is not used

Attention! The order of these 4 poses must be arranged either clockwise or counterclockwise. Not crossed!

3. FLEXFEEDER PREPARATION

Please follow the steps on this checklist to prepare the flexfeeder for productive operation or calibration.

1. Set up the flexfeeder electrically and mechanically. For more information, see the "Assembly instructions" documentation.

2. Power on the flexfeeder, execute a Reset command from the touchpad and set the operating mode to "Auto".

3. Teach a component for calibration or productive operation. See the "How to set up a new part" documentation for more information.

4. Configure a corresponding feeder job on the flexfeeder for productive operation. This includes setting the component-specific feeder parameters such as kicks and intensity for the different feeder actions as well as the type of illumination.

5. Activate the corresponding feeder job on the flexfeeder touchpad.

4. SOFTWARE INSTALLATION AND BASIC SETUP

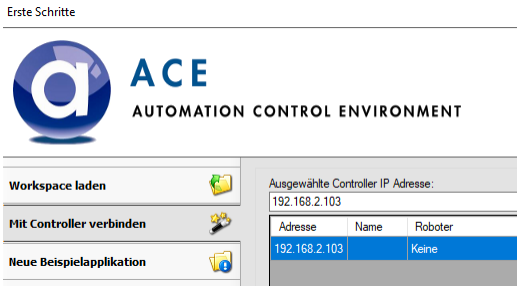
The "flexfeeder OMRON Plug-in" [Version: 1.0] was developed and tested with the development environment ACE [Version: 3.8.3.250].

To install and set up the flexfeeder OMRON plug-in, it is assumed that the ACE or Adept DeskTop software is installed and ready for use.

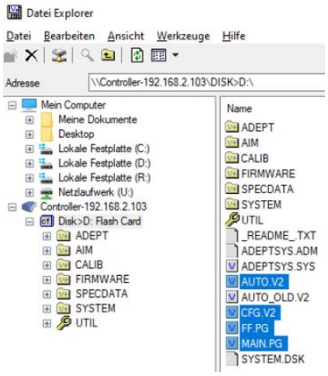
Chapters 4-6 illustrate the setup using the ACE development environment. Setup and operation is also possible with the older development environment "Adept Desktop" and is almost identical.

To install and then open the plug-in, follow these steps:

1. Launch the ACE program and connect to the controller.

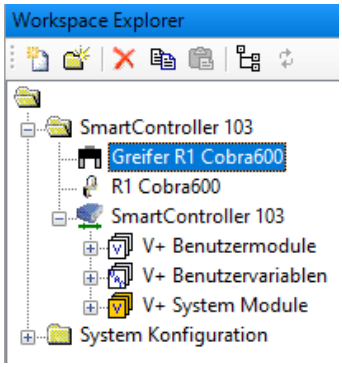


2. Unzip the ZIP file "flexfeeder-OMRON-Plug-in-1.0.zip".



3. Open the File Explorer in ACE and copy the four files of the plug-in (AUTO.V2, CFG.V2, FF.PG and MAIN.PG) to Disk>D: Flash Card.

4. Save the program as a workspace via "File > Save as...". The next time you start it, you can either start the workspace you just saved or open it via the "Load file >" menu.



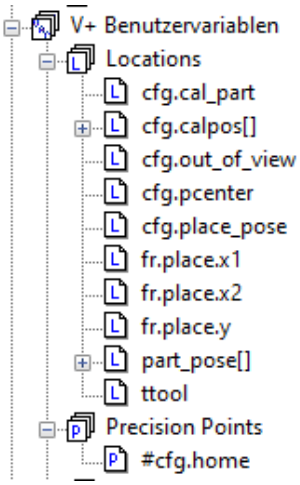
In order to run the plug-in's sample programs correctly, the following adjustments must be made to these project files:

1. Define Tool Height:
   Open the setting of the gripper (e.g. gripper R1 Cobra600) and then define the Z-height of the gripper.

2. Customization of the robot pose "Home Position":
   The robot poses can be taught using the "Robot Jog Control" tool. Make sure that the correct tool has been selected during the teach-in.

   i. ***cfg.home***
   Home position of the robot (start position).

3. Adaptation of robot poses for calibration:



   i. ***cfg.cal_part***
   Pick up pose of the calibration object to be used for calibration. When teaching this point, the calibration object should be present in the gripper.

   ii. ***cfg.calpos[1]*** to ***cfg.calpos[4]***
   The placement poses at the four points on the feeder surface. When teaching this point, the calibration object should be present in the gripper.

   iii. ***cfg.out_of_view***
   Robot pose outside the camera's field of view, which is approached to trigger a new image capture by the camera.

4. Adaptation of robot poses for productive operation:

   i. ***cfg.pcenter:***
   This robot pose brings the robot to the center of the feeder surface and is used, in conjunction with the correct feeder model, to calculate and set a monitored robot area. This monitored area is used in the multi-threaded example to realize an optimized triggering of the camera. This means that the camera is only triggered when the robot has left this monitored field of view of the camera. This pose is also used to drop parts in the middle if there is already a part in the gripper at program start

   ii. ***cfg.place_pose***
   This pose is used to place the gripped part in a defined manner. When teaching this point, the part should be present in the gripper.

   (optional)

   iii. *Frame „place“*
   As an alternative to a fixed place position, a frame can be defined in which the parts are placed.
   The number of positions in the frame results from the area of the taught frame "fr.place.x1", "fr.place.x2", "fr.place.y1" as well as their spacing in axes "dx" and "dx".
   ➔ See the place(opcode) function.

5. Adjustment of connection parameters:

   Please replace the appropriate values for your flexfeeder in the program "cfg.init".

   Explanation of the variables:

```
$cfg.ip_address = "192.168.2.100"
cfg.port_nr = 8082
cfg.socket_tout = 2
$ff.endofstring = $CHR(10)
```

   i. *cfg.ip_address*
   The IP address of the flexfeeder you want to connect to.

   Default value: "192.168.2.100"

   ii. *cfg.port_nr*
   TCP port of the flexfeeder communication interface.

   Note: This value should normally already be correct and should only be adjusted in special cases.

   Default value: 8082

   iii. *cfg.socket_tout*
   Socket Timeout = 2

   Note: This value is normally correct and should not be adjusted.

6. Customization of specific configuration variables

Please replace the configuration variables that are suitable for your flexfeeder in the program "cfg.init"

```
$cfg.ff_model = "X250"
$cfg.jobname = "Product 1"
$cfg.cal_job = "ffCalib.job"
```

:Explanation of the function parameters:

i. *cfg.ff_model*
Select your flexfeeder model.
Accepted values: "X185", "X250" and "X350"

ii. *cfg.jobname*
Name of the feeder job to be used for production.

iii. *cfg.cal_job*
name of the camera job to be used for calibration.

```
; operation mode : new picture aft
        cfg.search_allp = TRUE

; use of the shaker cleared messag
        cfg.shaker_clr = 0
```

iv. *cfg.search_allp*
This variable defines whether all found parts (max. 10) should be transferred to the robot (= TRUE), or if only the coordinate of the next part to pick should be transferred (= FALSE).

v. *cfg.shaker_clr*
(only with "cfg.run_multith" and without "cfg.search_allp" applicable)
This variable defines whether the "Shaker Vertical Cleared" command should be used. This command triggers a next feeder action as soon as the robot has picked up the last found part from the feeder and has moved up in the Z-axis, but is still above the feeder (premature feeder action).

- Value = 0:
  "Shaker Vertical Cleared" command is disabled.

- Value = Height of the Z-axis above the feeder surface at which the "Shaker Vertical Cleared" command will be executed.
  Attention: This value must be set high enough so that the feeder can continue to make movements.

```
SET ttool = TRANS(0,0,85,0,0,0)
```

```
; height of the feeder in world coordinates
        cfg.height_feed = 248

; approach/depart for feeder
        cfg.ff_app = 80

; approach/depart for place position
        cfg.place_app = 80
```

7. Adjustment of the tapping height in the program "cfg.init":

   i. *tTool*
   Defines the height of the gripper

   cfg.height_feed
   This variable serves as the initial value for the Z-height at which the components are picked up from the feeder surface during productive operation. However, when the program starts, this value is overwritten by the Z-height of the position "cfg.pcenter"

   ii. *cfg.ff_app*
   Approach height when moving to the feeder (approach)

   iii. *cfg.place_app*
   Approach height when moving to the tray (approach)

```
; signals
        o.gri_enable = 8
        o.gri_open = 3003 ; 3001
        o.gri_close = 3001 ;  3002
```

8. Adjustment of gripper settings

   i. In the program "cfg.init" you can adjust the signals for your gripper system.

   ii. Please apply these changes in the programs "grab_part" and "release_part" (in "a.main") to close or open your gripper system.

```
; soft signals
        s.start = 2017
        SIGNAL -s.start
```

9. Adjustment of the soft signals in the program "cfg.init":
   You can set the trigger signal to start the application at this point.

```
; run calibration ?
        cfg.run_calibr = FALSE
```

10. Calibration adjustment
    To perform the calibration via the soft signal, the variable "cfg.run_calibr" must be set to "TRUE".

11. Pick&Place setting
    To start productive operation via the soft signal, the variable "cfg.run_calibr" must be set to "FALSE".

    In addition, it can be defined whether the simple sequential pick & place process or the multi-threaded pick & place sequence (optimized for cycle time) should be executed:

```
; run multi-threaded ?
        cfg.run_multith = TRUE
; run calibration ?
        cfg.run_calibr = FALSE
```

    ➔ cfg.run_multith = FALSE
    or
    ➔ cfg.run_multith = TRUE

The modifications made are now automatically initialized from the "cfg.init" program every time the program is started.

To store modifications made to a module permanently on the flash memory of the robot controller, the "z" command of the respective module must be executed.

Example:
All settings and teach values are stored in the CFG module. To permanently store these values on the controller's flash, open the terminal in the ACE environment and execute the following command: "com z.cfg"

5. CALIBRATION

Before performing the calibration for the first time, please make sure that the following preparations and basic adjustments have already been made:

1. The camera recognition of the calibration object was set up in the calib-job of the vision system.
   Note: Please refer to the documentation "Instructions for setting up a new part".

2. The adjustments to the connection parameters have been made in the variables of the program "cfg.init", as described in chapter 4.

3. The corresponding camera job has been defined in the program "cfg.init".

4. The variable "cfg.run_calibr" has been set to "TRUE".

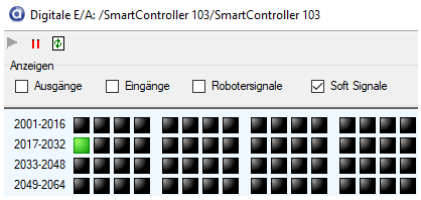
5. The robot poses have been adjusted as described in chapter 4.

   Note: Please make sure the Z-coordinates of the calibration poses are set up to match your gripping system and the calibration object.

6. The gripper settings have been adapted to your gripping system, as described in chapter 4 and the functions *„grab_part“* and *„release_part“* perform closing and opening of the gripper.

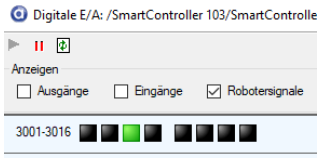
7. Attention! Before the first attempt, reduce the speed of the robot in the "init_robot" program.

```
; default monitor speed
        SPEED 10 MONITOR
```

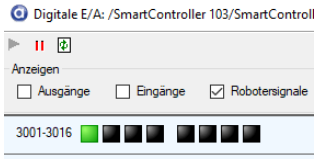


8. Open the "Digital Input/Output" tool and activate the soft signal "2017". This signal will start the application. The flexfeeder carries out an initialization and the robot moves over the calibration part.

9. To continue the process, activate the robot signal "3003":



Otherwise, activate the robot signal "3001" to abort the calibration process and return to the original position:



6. PRODUCTION OPERATION

Before executing the Pick&Place process for the first time, please make sure that the following preparations and basic adjustments have already been made:

1. The component to be separated was set up with the vision system of the flexfeeder. Note: Please refer to the documentation "Instructions for setting up a new part".

2. The adjustments to the connection parameters have been made in the variables of the program "cfg.init", as described in chapter 4..

3. The corresponding feeder job has been specified in the "cfg.init" program.

4. The variable "cfg.run_calibr" has been set to "FALSE".

5. The robot poses have been adjusted as described in chapter 4.

6. Feeder height adjustment:
   i. ***cfg.height_feed:***
      Initial value (set to max. height of the robot)
   ii. ***cfg.pcenter:***
      The Z-value of this point represents the pick-up height from the feeder.

7. The specific configuration variables in the program "cfg.init" have been adjusted:
   i. ***cfg.ff_model*** *(feeder type)*
   ii. ***cfg.jobname*** *(production job)*
   iii. ***cfg.search_allp*** *(search only one part / all part)*
   iv. ***cfg.run_multith*** *(sequential flow / Multi-Threaded)*
   v. ***cfg.shaker_clr*** *(early movement)*
   (only with "cfg.run_multith" and without "cfg.search_allp" applicable)

8. The gripper settings have been adapted to your gripping system, as described in chapter 4 and the functions *„grab_part“* and *„release_part“* perform closing and opening of the gripper.

9. Placement is defined by either a single point "cfg.place_pose" or the frame "fr.place", which can be defined with the following points:

   - fr.place.x1
   - fr.place.x2
   - fr.place.y

   The placement variant to be used is defined in the program "run_prd()" by the "IF" loop:

```
; place the part
                IF TRUE THEN
                    CALL place(1)
                ELSE
                    CALL status($ENCODE("placing part ",i), 0)
                    SINGLE
                    APPRO cfg.place_pose, cfg.place_app
                    COARSE
                    MOVES cfg.place_pose
                    BREAK
                    CALL release_part()
                    DEPARTS cfg.place_app
                END
```

   - If the IF loop is defined as TRUE, the parts are stored at the drop point "cfg.place_pose".

   - If the IF loop is defined as FALSE, the parts are placed in a frame. The distances to each other can be changed in the program "place()" using the variables "dx" and "dy".

10. Attention! Before the first attempt, reduce the speed of the robot in the "init_robot" program.

```
; default monitor speed
        SPEED 10 MONITOR
```

11. Open the "Digital IO" tool and activate the "2017" soft signal. This signal will start the application and run until the soft signal "2017" is reset.

6. GUI [OPTIONAL]

In the "GUI" folder of the flexfeeder OMRON plug-in, there are two different GUI variants, which are identical in terms of operation:

- **FF-UI EX:**
  This GUI version is intended for the newer Omron EX controller.

  Installation:
  start MiniUI.exe

  Program Path:
  GUI\FF-UI EX\FF-UI. C#\bin\Debug\MiniUI.exe

- **FF-MiniUI CX:**
  This GUI version is intended for the older Omron CX controller.

  Installation:
  start MiniUI.exe

  Optional:
  If the GUI does not open as shown below, you must first run the DLL file "RegisterDLL.bat" with administrator privileges. Then you can start the program MiniUI.exe.

  Program Path:
  GUI\FF-MiniUI CX\FF-MiniUI-CX.C#\bin\MiniUI.exe

The GUI makes it easy to start the calibration process and the pick & place.

Connection establishment:
In the GUI, enter the IP address of the robot controller and click the "Connect" button to connect. Once the connection is successfully established, the Connect button will turn green.

Start calibration:

1. Attention! Before the first attempt, reduce the speed of the robot in the "ini_robot" program:

```
; default monitor speed
        SPEED 10 MONITOR
```

2. Activate the "Calibration" checkbox and then press the "Start" button (soft signal 2017). The flexfeeder carries out an initialization and the robot moves over the calibration part.

3. To confirm the calibration procedure, select the "Gripper Open" checkbox (robot signal 3003).
   To cancel the calibration process, select the "Gripper Close" checkbox (robot signal 3001).

Start Pick&Place:

1. Attention! Before the first attempt, reduce the speed of the robot in the "ini_robot" program:

```
; default monitor speed
        SPEED 10 MONITOR
```

2. Uncheck the "Calibration" box and select the desired flexfeeder job on the right. Then click on the "Start" button (soft signal 2017).

3. To end the Pick & Place process, click on the "STOP" button (soft signal 2017 will be reset).

8. TROUBLESHOOTING

A TCP/IP connection to the flexfeeder cannot be established.	Check the Ethernet infrastructure, including the patch cable and switches. Check if the IP address of the flexfeeder is set correctly. Note: The IP address of the flexfeeder is displayed on the HMI on the home screen (accessible via web browser or touch screen).
The camera triggers at the wrong time or not at all.	1. Please verify that the robot pose "cfg.pcenter" is set correctly. 2. Please check if the correct flexfeeder model has been set in the variable "cfg.ff_model". 3. The monitored 'box' of the cameras field of view is then set by the dimension values in the function "ff.set_ff_model".