

Author Roland Loy
Date 05/09/2023
Revision 1.0

FLEXFEEDER EPSON PLUG-IN

Quickstart & Reference Guide

flexfeeder GmbH
Gerhard-Kindler-Straße 8
72770 Reutlingen

Phone +49 7121 8933-663

flexfeeder@flexfactory.com
flexfactory.com



INTRODUCTION

This file documents the quick steps for setting up and configuring the flexfeeder EPSON Plug-in project from Flexfactory AG for operating a Flexfactory flexfeeder X with an EPSON robot.

Table of content:

1. System Installation
2. Relevant Robot-Poses
3. Preparation flexfeeder
4. Installation and Setup EPSON Plug-in
5. Hand-Eye Calibration
6. Productive Operation
7. Trouble Shooting

Note: The hand-eye calibration is needed one time only to know the mapping between the camera and robot coordinate systems.

In case of an error please refer to chapter [Trouble Shooting](#).

We would like to point out that we recommend participation in one of our one-day training courses. These trainings will provide you with valuable insights and give you the necessary 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>



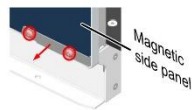
1. SYSTEM INSTALLATION

How to connect the robot to the flexfeeder

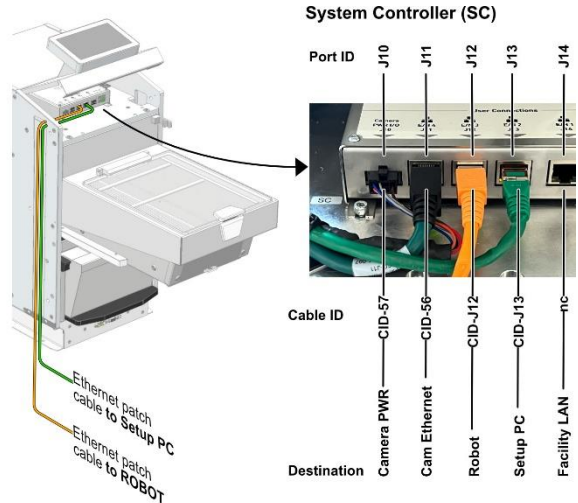
The flexfeeder's System Controller provides ethernet ports to the robot system. A simple patch cable enables the electrical and logical connection of both systems.

How to install the Ethernet patch cables ?

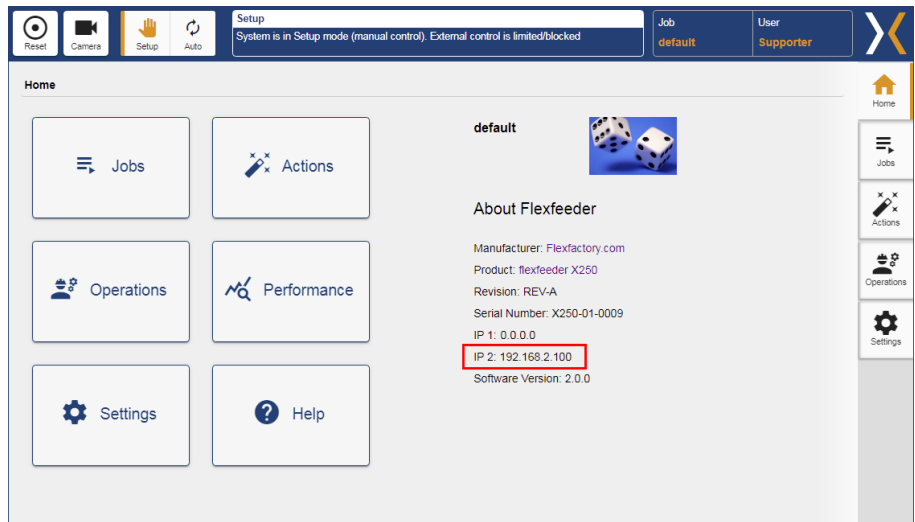
- 1 Remove the 2 screws and the plate as shown below:



- 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 seen on the start page of the display:



2. RELEVANT ROBOT-POSES



Out of view pose: Pose of the robot arm located outside the flexfeeder camera's field of view.

Name of the robot pose: *ffx_out_of_view_pose*

Attention: The gripper and any cables should also be outside the camera's field of view in this pose.



Center of feeder pose: Pose of the robot arm located in the center of the flexfeeders.

This pose is used in context with the set up feeder model to calculate and set a monitored robot area. This monitored area is used in the multi-thread example to realize an optimized triggering of the camera. I.e. the camera is only triggered when the robot has left this monitored viewing area of the camera.

Name of the robot pose: *center_feeder_pose*

Note: For the calculation of the monitored robot area, only the X and Y coordinates of the set up robot pose are used. The Z coordinate is not taken into account.





Calibration poses: The calibration object is placed at four different positions on the feeder surface. Set the corresponding robot poses accordingly.

Names of the robot poses: *ffx_calibration_c1 - ffx_calibration_c4*

Attention: The order of the poses must be either clockwise or counterclockwise.

Important: Do not cross



3. PREPARATION FLEXFEEDER

Please follow the check list to get the flexfeeder ready for production and calibration.

1. Set up flexfeeder electrically and mechanically. Further information can be found in its documentation “assembly instruction”.
2. Switch on flexfeeder, execute a reset command from its touchpad and turn operation mode to “auto”.
3. Teach a user defined part for calibration and production, respectively. See documentation “Anleitung zum Einrichten eines neuen Teils” for further information.
4. Configure appropriate jobs on the flexfeeder for calibration and production, respectively. This includes product specific parameters like feeding force and frequencies.
5. Activate one of those two jobs on the flexfeeder touchpad depending on what you want to do next: run a calibration or a production cycle.

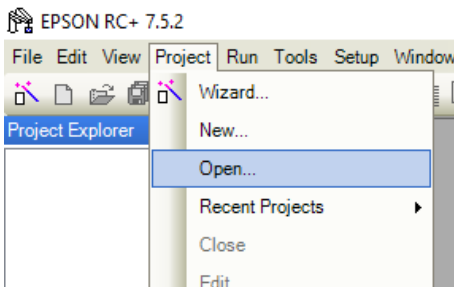


4. INSTALLATION AND SETUP EPSON PLUG-IN

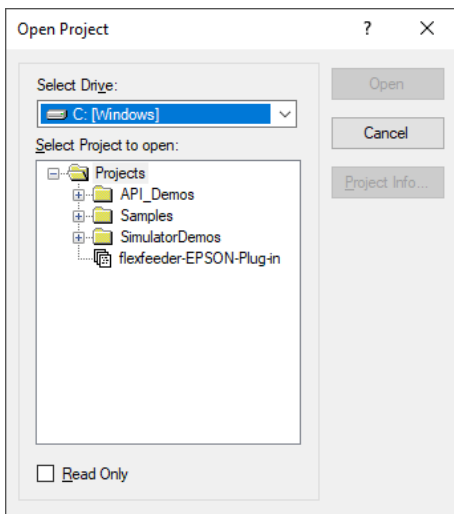
The " flexfeeder EPSON Plug-In" [Version: 1.0] has been developed and tested with the development environment EPSON RC+ 7.0 [Version: 7.5.2].

To install and set up the flexfeeder EPSON Plug-In, it is assumed that EPSON RC+ 7.0 is installed and functional.

Perform the following steps to install and then open the project:



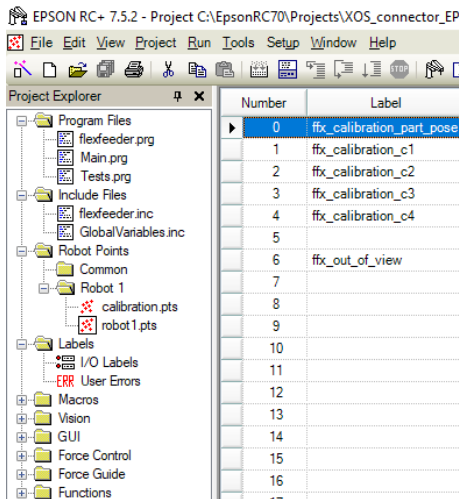
1. Unzip the ZIP file " flexfeeder-EPSON-Plug-In-1.0.zip".
2. Move or copy the unpacked directory "flexfeeder-EPSON-Plug-In" to "C:\EpsonRC70\projects\".
3. Start the EPSON RC+ 7.0 development environment.



4. Open the project:
Menu: Project → Open... → flexfeeder-EPSON-Plug-In select and click "Open".



In order to run the sample programs correctly, the following adjustments must be made:



1. Teaching the robot poses:

a. For the adjustment of the calibration poses please open the file "calibration.pts" and adjust the following robot poses to your conditions:

i. **ffx_calibration_part_pose**

Pickup and place down pose of the calibration object that is to be used for the calibration process.

ii. **ffx_calibration_c1** bis **ffx_calibration_c4**

The place down poses in the four corners of the feeder surface.

Note: See List of Robot poses → Calibration poses

iii. **ffx_out_of_view_pose**

The robot pose outside the camera's field of view, which is approached to trigger a new image capture of the camera.

iv. The poses **ffx_part_pose1** to **ffx_part_pose10** are runtime variables that are used to temporarily store the detected pick poses. Therefore, they do not need to be set up.

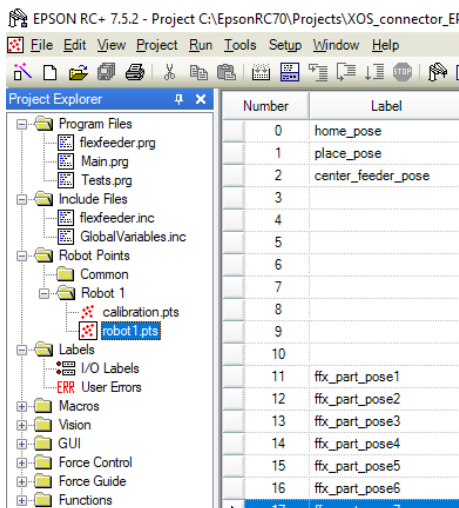
b. For the teaching of the robot poses of the productive operation please open the file "robot1.pts" and adapt the following robot poses to your conditions:

i. **home_pose:**

The start pose of the robot.

ii. **place_pose**

This place down pose is used to place the gripped component in a defined way.



iii. **center_feeder_pose**

This robot pose places the robot in the center of the feeder surface and is used in conjunction with the set up 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. I.e. the camera is only triggered when the robot has left this monitored viewing area of the camera.

2. Configuration of the connection parameters in the functions *main_calibration*, *main_multi_thread* and *main_single_thread*.

Please replace the connection parameters suitable for your flexfeeder in the following function calls:

```
Call ffx_setup_connection(201, "192.168.2.100", 8082, 40)
```

Explanation of the function parameters:

a. ***comm_port***

EPSON specific internal connection port.

Value range: 201 – 216

Note: This value should only be adjusted if the preset port (201) is already used by another communication channel.

b. ***address_ip\$***

The IP address of the flexfeeder to connect to.

Default value: „192.168.2.100“

c. ***address_port***

TCP port of the flexfeeder communication interface.

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



d. ***timeout***

Time in seconds to wait during a communication process until a timeout error occurs.

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

3. Setting up the project-specific configuration parameters in the functions *main_multi_thread* and *main_single_thread*.

Please replace the configuration parameters suitable for your flexfeeder in the corresponding function calls:

Call `ffx_setup_feeder("X250", "ffX-demo", -100.0, False, True)`

Explanation of the function parameters:

a. ***feeder_model\$***

Choose your flexfeeder model.

Possible values: "X185", "X250", "X350"

b. ***feeder_job_name\$***

Name of the feeder job to be used.

c. ***z_on_feeder***

This parameter sets the Z coordinate with which the parts are to be picked up from the feeder surface.

Attention! There is a risk of a collision! This Z coordinate depends on the gripping system used and may have to be adjusted accordingly.

d. ***search_all_parts***

This flag sets whether all found components (max. 10) should be determined (= True), or whether only the next available component should be determined (= False).

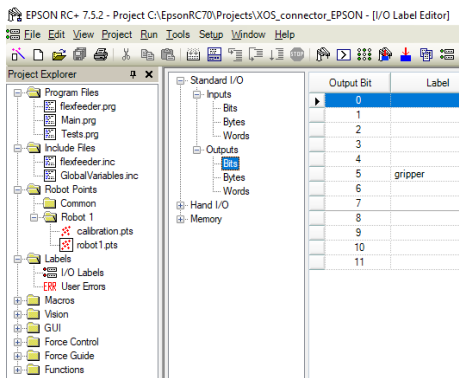
Possible values: "True", "False"

e. ***use_shaker_vertical_cleared***

This flag sets whether the "Shaker Vertically Cleared" command should be used.



Possible values: "True", "False"

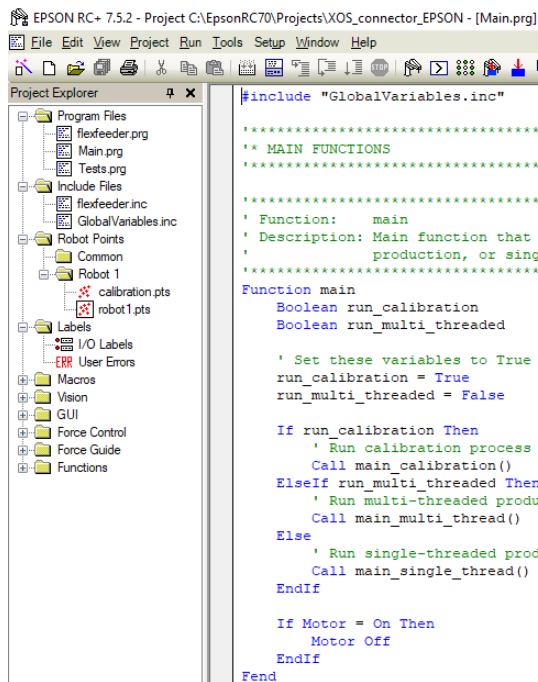


4. Adjustment of the gripper settings and functions.

- a. At *Labels* → *E/A-Labels* → *Ausgänge* → *Bits* you can customize the inputs and outputs for your gripping system.
- b. Please apply these changes in the *grab_part* and *release_part* functions to close or open your gripping system.



5. HAND-EYE CALIBRATION



The screenshot shows the EPSON RC+ 7.5.2 software interface. The Project Explorer on the left displays a tree view of the project files, including Program Files, Include Files, Robot Points, Labels, Macros, Vision, GUI, Force Control, Force Guide, and Functions. The main window shows the code for Main.prg, which includes GlobalVariables.inc and defines the main function. The code sets run_calibration to True and run_multi_threaded to False, and then calls main_calibration and main_multi_thread based on the run_multi_threaded variable.

```
#include "GlobalVariables.inc"

*****
** MAIN FUNCTIONS
*****

' Function:  main
' Description: Main function that
'             production, or sing
*****

Function main
  Boolean run_calibration
  Boolean run_multi_threaded

  ' Set these variables to True
  run_calibration = True
  run_multi_threaded = False

  If run_calibration Then
    ' Run calibration process
    Call main_calibration()
  ElseIf run_multi_threaded Then
    ' Run multi-threaded produ
    Call main_multi_thread()
  Else
    ' Run single-threaded prod
    Call main_single_thread()
  EndIf

  If Motor = On Then
    Motor Off
  EndIf

End
```

Open the program file "Main.prg" and adjust the two variables *run_calibration* and *run_multi_threaded* in the *main* function as follows to execute the calibration sequence:

```
run_calibration = True
run_multi_threaded = False
```

Before executing the program for the first time, please make sure that the following preparations and basic configuration have already been made:

1. A calibration object is set up in the flexfeeder vision system.
Note: Please refer to the documentation „Anleitung zum Einrichten eines neuen Teils“ for this.
2. The configuration of the connection parameters was done in the function *main_calibration* and in the function call:

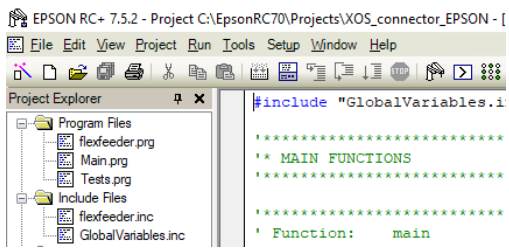
```
Call ffx_setup_connection(201, "192.168.2.100", 8082, 40)
```

3. The corresponding camera job is specified in the *main_calibration* function and in the function call:

```
Call ffx_calibration_start("ffCalib.job")
```

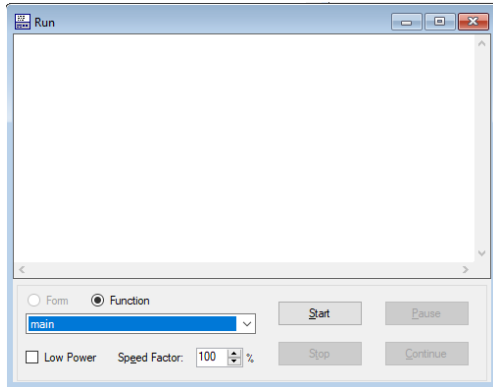
4. The robot poses have been adjusted as described in the chapter [Installation and Setup](#).
Note: Please make sure that the Z coordinates of the calibration poses are set up to match your gripping system and the calibration object.
5. The gripper settings have been modified to your gripper system as described in the chapter [Installation and Setup](#) and the functions *grab_part* and *release_part* perform a closing and opening of the gripper system.





6. If necessary, please modify the functions *pick_up_calibration_part* and *place_down_calibration_part* to your conditions in order to pick up or to place the calibration object.

Open the Run window by clicking...



Please select the *main* function in the drop-down box and click “Start” to execute the calibration procedure.



6. PRODUCTIVE OPERATION

Open the program file "Main.prg" and adjust the two variables *run_calibration* and *run_multi_threaded* in the *main* function as follows to either:

1. run the simple sequential pick-and-place sequence:

```
run_calibration = False  
run_multi_threaded = False
```

or

2. Run the multi-threaded pick-and-place sequence optimized for runtime:

```
run_calibration = False  
run_multi_threaded = True
```

Before executing the program for the first time, please make sure that the following preparations and basic configuration have already been made:

1. The part to be gripped is set up with the flexfeeder's vision system.
Note: Please refer to the documentation „Anleitung zum Einrichten eines neuen Teils“ for this.

2. The configuration of the connection parameters was done in the function *main_multi_thread*, resp. *main_single_thread* and in the function call:

```
Call ffx_setup_connection(201, "192.168.2.100", 8082, 40)
```

3. The adaptation of the project-specific configuration parameters is done in the function *main_multi_thread*, resp. *main_single_thread* and in the function call:

```
Call ffx_setup_feeder("X250", "ffX-demo", -100.0, False, True)
```

4. The robot poses were set up as described in the chapter [Installation and Setup](#).



Note: Please note that the Z coordinate of the gripping poses must be set to match your gripping system and the part. Please check if the configuration parameter *z_on_feeder* is set correctly.

5. The gripper settings have been modified to your gripper system as described in the chapter [Installation and Setup](#) and the functions *grab_part* and *release_part* perform a closing and opening of the gripper system.

Open the Run window by clicking....



Please select the *main* function in the drop-down box and click “Start” to execute the calibration procedure.



7. TROUBLE SHOOTING

A TCP/IP connection to the Flexfeeder cannot be established.	Check ethernet infrastructure including patch cables and switches. Check if flexfeeder's IP address is set correctly and check the <i>ffx_setup_connection address_ip</i> and <i>address_port</i> parameters. The IP of the feeder is displayed on its HMI on the home screen (accessible via web browser or touch screen).
The camera is not triggering well (at the wrong time etc.)	<ol style="list-style-type: none">1. Please check if the robot pose <i>center_feeder_pose</i> is set correctly.2. Please check if you have parameterized the correct flexfeeder model <i>feeder_model</i>.

