Author Roland Loy
Date 10/04/2024
Revision 1.0

FLEXFEEDER KUKA PLUG-IN

Quickstart & Reference Guide

flexfeeder GmbH
Gerhard-Kindler-Straße 8
72770 Reutlingen

Phone +49 7121 8933-663

flexfeeder@flexfactory.com
flexfactory.com

Contents

INTRODUCTION

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

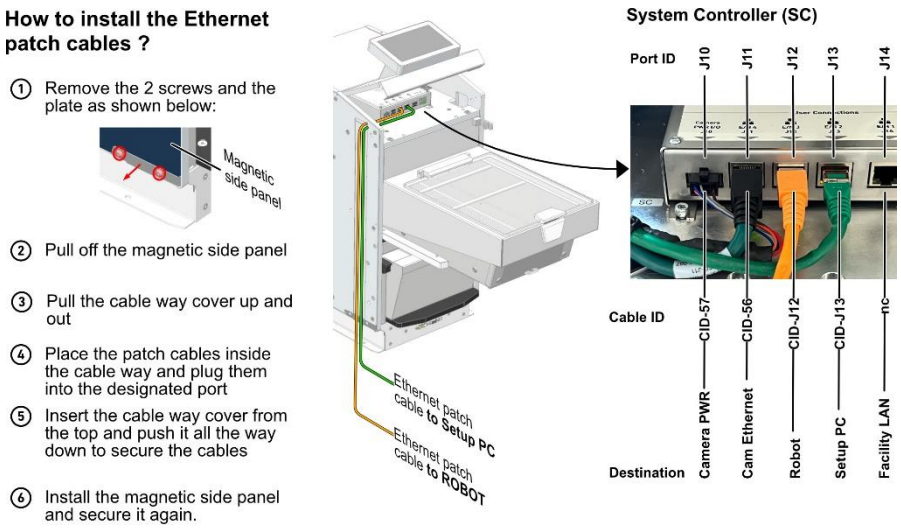
In case of an error please refer to chapter Trouble Shooting.

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.

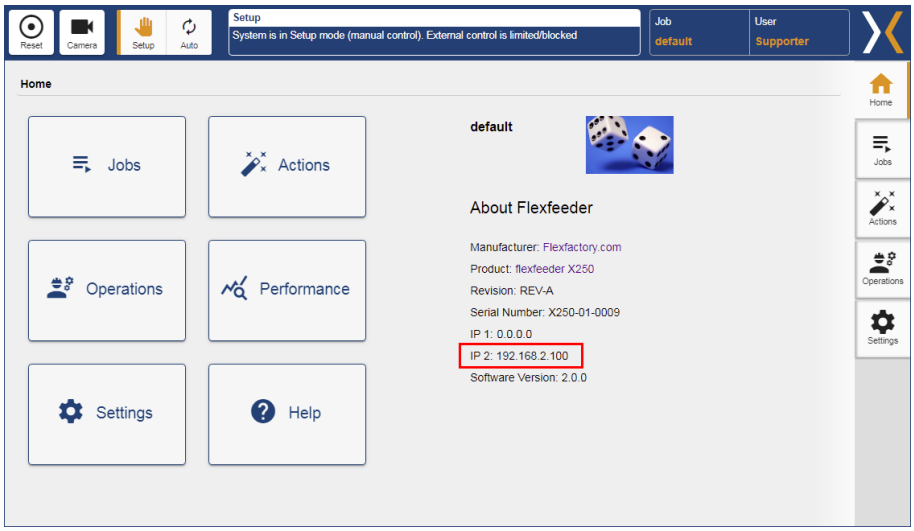
For more information and to register, please visit our training page at: https://www.flexfactory.com/en/contact-and-support/trainings

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.



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



2. IMPORTANT ROBOT POSITIOINS

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

Name of the robot pose: *Xffx_out_of_view_pose*

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

Above shaker pose: After gripping a part from the surface, this robot pose is approached. E. g. teach pose above a corner of the feeder surface.

Name of the robot pose: *Xffx_above_shaker*

Attention! This pose must be accessible without collision from any gripping poses on the feeder.

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: *XCPP1 – XCPP4*

Attention! The order of the poses must be either clockwise or counterclockwise.
Important! Do not cross

Pick pose: Set this pose as if an object is picked from anywhere on the feeder surface. This pose is adjusted automatically during productive operation according to detected objects.

Name der Pose: *Xpick_pose*

Attention! Feeder surface needs to be perpendicular to camera's optical axis.

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 KUKA PLUG-IN

The „flexfeeder KUKA Plug-in“ [Version: 1.0] has been developed and tested with the development environment WorkVisual 6.0 [Version: 6.0.28].

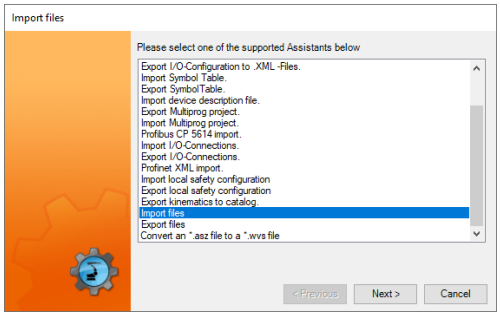
To install and set up the flexfeeder KUKA plug-in, it is required that the WorkVisual 6.0 software has been installed and is functional.

Follow the steps below to install and then open the project:

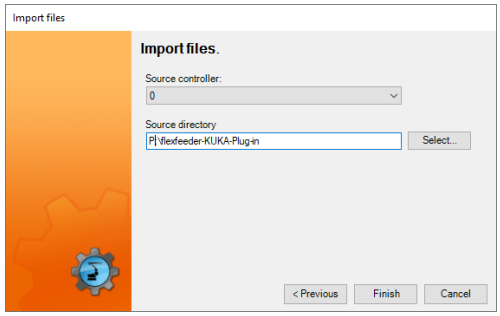
1. **Unzip** the ZIP file „flexfeeder-KUKA-Plug-in-1.0.zip“.
   It contains the following directories:
   - *docs:*
     Documentation in German and English.
   - *flexfeeder-KUKA-Plug-in:*
     Source code of the flexfeeder KUKA Plug-in and the sample applications.

2. **Start** the **WorkVisual 6.0** development environment.

3. Create a new project with the desired configuration, or open the project into which you want to **import the flexfeeder sample programs**:
   Menu *File* → *Import / Export*

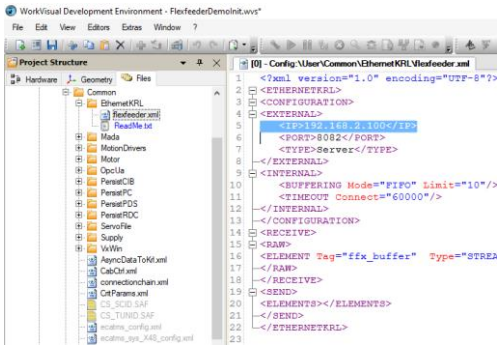


4. In the "Import files" dialog, scroll down the list, select "Import files" and click next.



5. **Select the controller and the source directory** and click on "Finish".

   Select the flexfeeder-KUKA-Plug-in source code directory of the zip file you have just unzipped as the source directory.



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

Note: For easy teaching and testing of the poses, two functions *setup_poses()* and *dry_run_poses()* have been integrated into the *.src files.

1. Configuration of the **connection parameters** of the **Ethernet KRL interface**:
   a. Open the XML file "flexfeeder.xml" in the directory "/Config/User/Common/EthernetKRL/".
      i. Please configure the IP address of the flexfeeder to which the robot should connect.
      ii. If possible, the remaining parameters should not be changed.

2. **Adjustment** of the **robot poses**:
   a. To adjust the **general robot positions**, please open the following files:
   „/R1/Program/flexfeeder/flexfeeder_poses.dat“
   „/R1/Program/flexfeeder/flexfeeder_poses.src“

   and adjust the following robot positions to your circumstances:
      i. *Xffx_above_shaker*
      The robot pose above the pick-up area, which is approached in order to leave the pick-up area safely without collision.

      Note: See list of important robot poses.

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

      Note: See list of important robot poses.

b. To adjust the **calibration poses**, please open the following files:
„/R1/Program/flexfeeder/calibration/calibration.dat“
„/R1/Program/flexfeeder/calibration/calibration.src“

and adjust the following robot positions to your circumstances:

i. *Xpick_up_calib_part*
Pick-up and place pose of the calibration object to be used for calibration.

ii. *XCPP1* bis *XCPP4*
The placement poses in the four corners of the feeder surface.

Note: See list of important robot poses → Calibration poses.

c. To adjust the **poses** of the **productive operation**, please open the following files:
„/R1/Program/flexfeeder/examples/single_thread.dat“
„/R1/Program/flexfeeder/examples/single_thread.src“

or
„/R1/Program/flexfeeder/examples/multi_thread.dat“
„/R1/Program/flexfeeder/examples/multi_thread.src“

and adjust the following robot positions to your circumstances:

i. Xpick_pose
This pose acts as a template for all other gripping poses. In particular, the Z coordinate of this pose is used during productive operation to determine the pick-up height.

Note: See list of important robot poses.

ii. *Xdrop_part*
This place pose is used to place the gripped part in a defined position.

3. Adjust the **configuration parameters** for the **calibration** in the file:
„/R1/Program/flexfeeder/calibration/calibration.src“

Please change the **configuration parameters** suitable for your flexfeeder in the *calibration()* function:

a. *camera_job_name[]*
Name of the camera calibration job.

b. *above_part_height_mm*
This parameter sets the distance in mm above which the parts are to be approached before the calibration part is placed or picked up.

Attention! There is a risk of collision! This distance in the Z direction depends on the gripping system or calibration part used and may need to be adjusted.

4. Adjust the **project-specific configuration parameters** in the files *"single_thread.src"* or *"multi_thread.src"*.

Please change the configuration parameters suitable for your flexfeeder in the *single_thread()* or *multi_thread()* function.

Explanation of the **function parameters**:

a. *feeder_job_name[]*
Name of the feeder job to use.

b. *above_part_height_mm*
This parameter sets the distance in mm above which the parts are to be approached before the calibration part is placed or picked up.

Attention! There is a risk of collision! This distance in the Z direction depends on the gripping system or calibration part used and may need to be adjusted.

c. *search_all_parts*
This flag sets whether all parts found (max. 10) are to be detected (= *TRUE*) or whether only the next available part is to be detected (= *FALSE*).

Permitted values: "*TRUE*", "*FALSE*"

d. *use_vertical_cleared*
This flag sets whether the "Shaker Vertical Cleared" command should be used.

Permitted values: "*TRUE*", "*FALSE*"

5. Adjust the **gripper settings and functions**. In the example programs, it is expected that the following two functions have been implemented and made available:
   a. *gripper_open*
   Opens the gripping system.

   b. *gripper_close*
   Closes the gripping system.

6. To be able to execute the *multi_thread.src* example program, the following **subtask** must be set up and made executable:

   „/R1/Program/flexfeeder/flexfeeder_task.sub“

   A detailed description of how to add and start this subtask on the robot controller can be found in the chapter Add Subtask.

5. CALIBRATION

To execute the calibration procedure, select the following program file
"/R1/Program/flexfeeder/calibration/calibration.src"
on the robot controller and execute it.

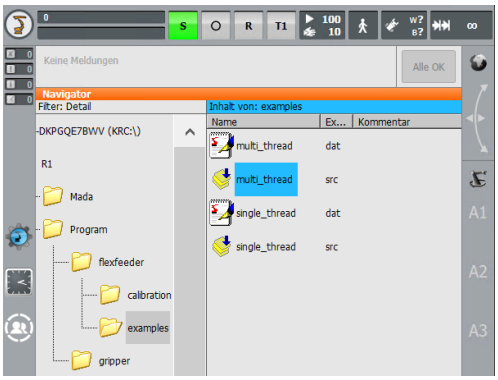
Before running the program for the first time, please make sure that the following preparations and basic adjustments have already been made:

1. A calibration object has been set up with the flexfeeder vision system.
   Note: Please refer to the documentation "Anleitung zum Einrichten eines neuen Teils".
2. The connection parameters were adjusted as described in the chapter Installation and Setup.
3. The camera job was specified in the *calibration()* function in the *camera_job_name[]* variable.
4. The robot poses have been adjusted as described in the chapter Installation and Setup.
   Note: Please ensure that the Z-coordinates of the calibration poses are set to match your gripping system and the calibration object.
5. The gripper settings have been adjusted to your gripper system as described in the chapter Installation and Setup and the function's *gripper_close()* and *gripper_open()* close and open the gripper system.
6. If necessary, please adjust the *handle_calibration_part()* function to your circumstances in order to pick up or place the calibration object.

If the above requirements are met, please execute the calibration procedure.

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

6. PRODUCTIVE OPERATION



Two different sample programs are available to run the sample programs for productive operation.

1. *Sequential pick-and-place process (single thread)*
   The flexfeeder request for new parts is processed sequentially. This means that when new parts are requested, the robot system must wait for the parts to be provided until a response from the flexfeeder has been received.

   Example program:
   „/R1/Program/flexfeeder/examples/single_thread.src“

2. *Runtime optimized pick-and-place process (multi thread)*
   The flexfeeder request for new parts is processed in a separate task parallel to the main program sequence. This means that when requesting new parts, the robot system does not have to wait for the parts to be provided until a response from the flexfeeder has been received and can therefore take care of other tasks during this time.

   Example program:
   „/R1/Program/flexfeeder/examples/multi_thread.src“

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

1. The part to be gripped has been set up with the flexfeeder vision system.
   Note: Please refer to the documentation "Anleitung zum Einrichten eines neuen Teils".

2. The connection parameters were adjusted as described in the chapter Installation and Setup.

3. The project-specific configuration parameters were adjusted in the *multi_thread()* or *single_thread()* function, as described in the chapter Installation and Setup.

4. The robot poses have been adjusted as described in the chapter Installation and Setup.

   Note: Please note that the *Xpick_pose* pose must be taught to match your gripping system and the part. (See: project-specific configuration parameters → *Xpick_pose*)

5. The gripper settings have been adjusted to your gripper system as described in the chapter Installation and Setup and the function's *gripper_close()* and *gripper_open()* perform closing and opening of the gripper system.

6. To execute the *runtime-optimized pick-and-place process (multi_thread)*, ensure that the *flexfeeder_task.sub* subtask is set up and executed correctly (See chapter: Add Subtask).

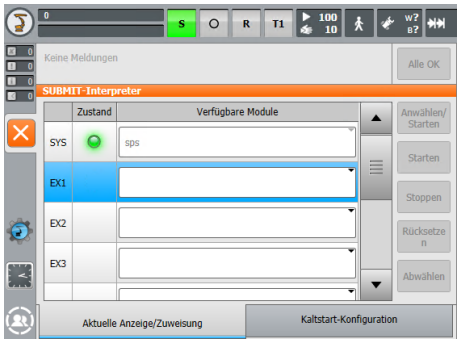
If the above requirements are met, please run the selected sample program for productive operation.

7. ADD SUBTASK

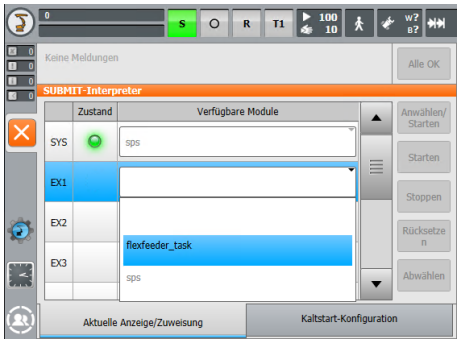
In order to properly run the example *"Runtime optimized pick-and-place sequence" (multi thread)*, it is necessary to run the subtask *flexfeeder_task.sub* on the robot controller.

Please follow the steps below to set up and start this subtask. Make sure that the project files have already been installed on the robot controller.

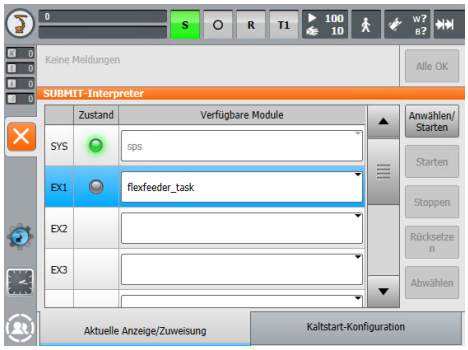
1. Select in the main menu **Configuration** → **SUBMIT-Interpreter.**



2. Select a free slot. E.g. the slot EX1.



3. Select the ***flexfeeder_task*** from the drop-down menu.



4. Click on *Select / Start* in the button bar on the right.

5. Please check the cold start configuration for this newly created subtask.

8. TROUBLE SHOOTING

A TCP/IP connection to the Flexfeeder cannot be established.	Check the ethernet infrastructure, including the patch cable and the switches. Check whether the IP address of the flexfeeder is set correctly. (See Installation and Setup → *Adjustment connection parameters*) Note: The IP of the feeder is displayed on its HMI on the home screen (accessible via web browser or touch screen).