Authors Jonas Kilian, Roland Loy
Date 25/03/2024
Revision 1.0

FLEXFEEDER FANUC 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 FANUC Plug-in project from Flexfactory AG for operating a Flexfactory flexfeeder X with a FANUC robot.

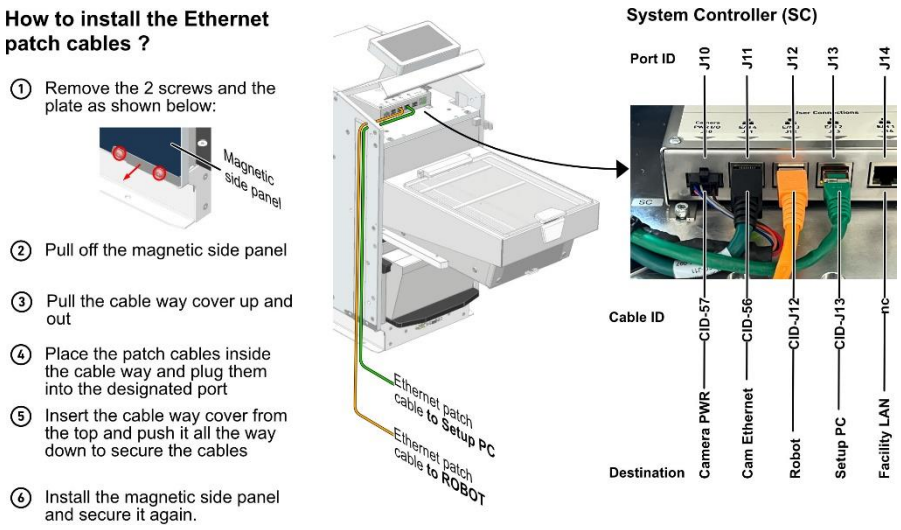
In case of an error please refer to chapter Trouble Shooting.

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.

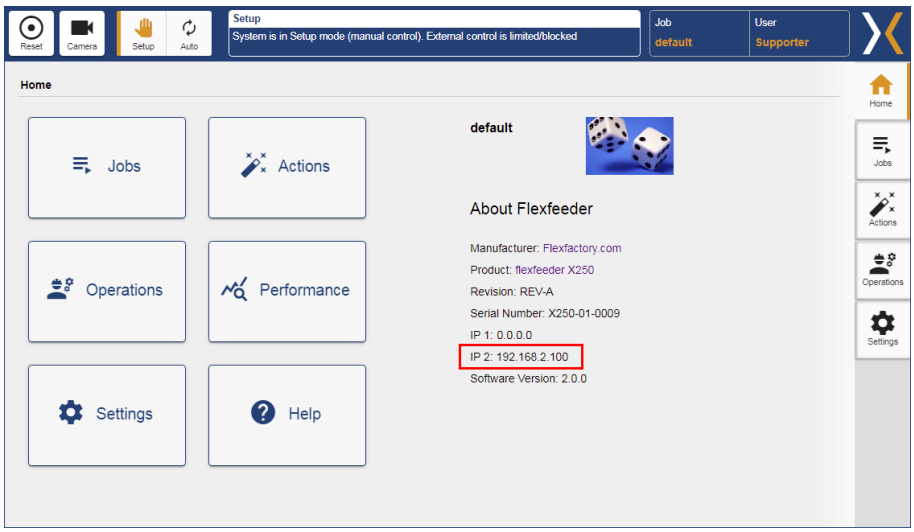
For more information and to register, please visit our training page at: https://www.flexfactory.com/en/contact-and-support/trainings

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 POSITIONS

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

Name of the robot pose: *Out of FOV*

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

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

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 of the pose: *PickPos*

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 FANUC PLUG-IN

The "flexfeeder FANUC Plug-In" [Version: 1.0] has been developed and tested with the development environment ROBOGUIDE [Version: v9.4.0 ZF].

To install and set up the flexfeeder FANUC Plug-In, it is assumed that ROBOGUIDE is installed and functional. The robot is in the delivery state and ready for operation.

The following option packages may need to be purchased and installed:

- R648 ! User Socket Msg
- R809 ! Motion Package

Follow the steps below to create the project and then transfer it to your robot controller:

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

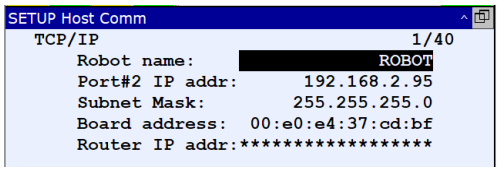
2. Start the ROBOGUIDE development environment and import the source code into the project you have created that matches your robot and controller setup.

3. Transfer the generated project files to your robot controller.

4. In order to use the transferred program files, they must be loaded individually on the robot controller.

In order to run the example program correctly, the following adjustments must be made:

1. Activate the *"Split screen"* option to be able to observe the output on the user screen while the program is running.



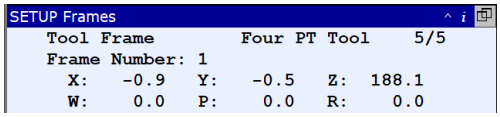
2. Set the ***robot IP address*** in *"Host Comm Menu"*.

   Example: The Ethernet cable is plugged into port 2 of the robot controller. The IP address set for the robot is 192.168.2.95.

3. In the *"Config Menu"*, activate the ***"Multi Program selection"*** option *(true)*.

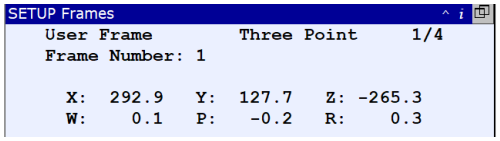
4. Set up the ***payload*** of your EOAT (= End Of Arm Tool) according to your gripper system.
   Payload 1 is used for this sample application.

5. Set up the ***TCP (=Tool Center Point)*** of your EOAT according to your gripper system.
   For this sample application, *"user tool 1"* is used.



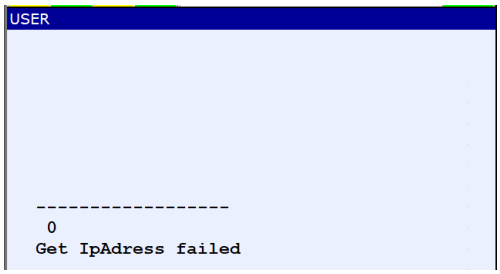
   Example: The TCP is parallel to the robot flange and in the original orientation.

6. Set up a ***user frame*** for the flexfeeder surface.
   For this example application, *"user frame 1"* is used.



   Example: The user frame is almost in the same orientation as the robot's world frame.

7. For the initial preparation of the sample application and to establish communication between the robot and flexfeeder, please follow the steps below:

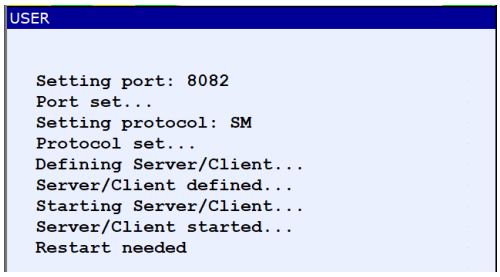


   a. Start the *"FFX_CHECK_INIT_COM"* program. The first time this program is executed, the comments for the signals/registers/flags are set. The program will abort with the message *"Get IpAdress failed"*.

   b. Please check that the robot IP address has been set correctly ("Host Comm Menu").

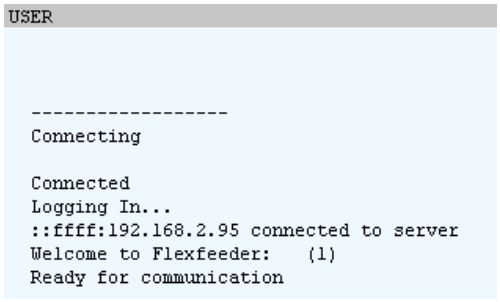
c. Enter the ***flexfeeder IP address*** in the string register *SR[13] IpAdress*.

Note: The IP address of the flexfeeder is shown on the home page of the flexfeeder display.

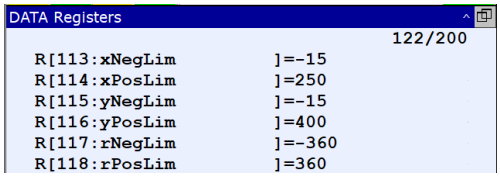


d. Start the *"FFX_CHECK_INIT_COM"* program again. In the second run of this program, the variables associated with the socket communication are set. The program will abort with the message *"Restart needed"*.

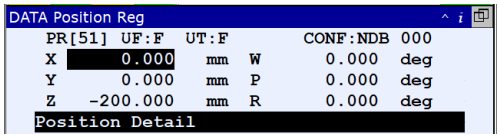
e. Please restart the robot controller.



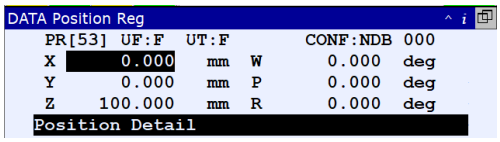
f. Now start the *"FFX_CHECK_INIT_COM"* program for the last time. If the connection has been successfully established, the message *"Ready for communication."* is displayed on the user screen.



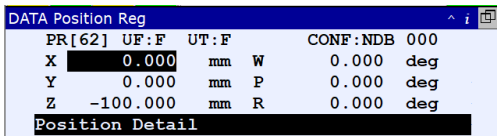
8. Set the registers *R[113-118]* (x/y/r) ***to limit the robot coordinates*** in relation to the *user frame 1* that has been set up.



9. Please set up the following ***position registers***:
   a. *PR[51] CalNestPre*
   [Tool Frame Offset]
   Offset in the direction of approach while picking the calibration object at its place position. This offset is used during the calibration process.



b. *PR[53] CalFfUOfsZ*
[User Frame Offset]
Offset in the direction of approach while picking the calibration object on the feeder surface (placement and pickup of the calibration object). This offset is used during the calibration process.



c. *PR[62] PickPosPre*
[Tool Frame Offset]
Offset in the direction of approach while picking the part on the feeder surface (pickup of the part). This offset is used during productive operation.

10. Adaptation of the ***robot poses***:
    a. To ***adjust the calibration poses***, please open the following file: *"FFX_CAL_MAIN"*. Please adapt the following robot poses to your conditions.

       Note: If you are asked to create a new id while setting up the robot poses, please answer "No".

       i. *Start pos*
          The start position from where the robot should start.

       ii. *Out of FOV*
          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.

       iii. *Nest pos calibration part*
          Pick up and place pose of the calibration object that will be used for the calibration.

       iv. *Four calibration poses*
          The placement poses in the four corners of the feeder surface.

Note: See list of important robot poses → Calibration Poses

b. To ***adapt the poses for productive operation***, please open the following file *"FFX_MAIN"*. Adapt the following robot poses to your conditions.

   Note: If you are asked to create a new id while setting up the robot poses, please answer "No".

   i. *Start pos*
   The start position from where the robot should start.

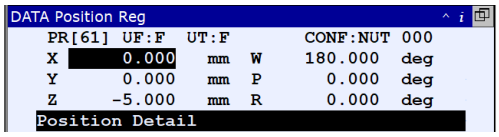
   ii. *Out of FOV*
   The robot pose outside the camera's field of view, which is approached to trigger a new image capture by the camera.

   Note: See list of important robot poses.

   iii. *Pick pose*
   Please follow the steps below in the "FFX_GET_PRT_POS" function to set up, adjust and test the *PickPos* pose:

   (1) Move the robot near the origin of the flexfeeder *user frame 1* and make sure that both the orientation and the configuration of the robot arm are suitable for your setup situation. Please teach this robot position in *PickPos* in the position register.

   (2) Run the test and adjustment area in the source code starting on line 10 to teach the *PickPos* and then test and, if necessary, adjust the picking height.



Example: The parameters in this example configuration are for the arm configuration *CONF: NUT*, the orientation is *W: 180.0, P: 0.0, R: 0.0*. The pickup height has been compensated by *5 mm*.

iv. *Place Pos*

This placement pose is used to drop the successfully gripped part in a well-defined location.

5. CALIBRATION

To perform the calibration process, please execute the program file *"FFX_CAL_MAIN"* on the robot controller.

Before running the program for the first time, please ensure that the following preparations and basic settings 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 configured as described in the Installation and Setup chapter.

3. The camera job was specified in the string register*SR[11] CalJobName*.

   Note: Please ensure that the calibration camera job is specified with the ".job" file extension.

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

5. The gripper settings have been configured for your gripping system as described in the Installation and Setup chapter, and the *"FFX_PICK"* and *"FFX_PLACE"* functions in order to close and open the gripping system.

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

To run the example program for productive operation, please open the program file *"FFX_MAIN"* on the robot controller.

Before running the program for the first time, please ensure 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 configured as described in the Installation and Setup chapter.

3. The project-specific configuration parameters were adjusted as described in the Installation and Setup chapter.

4. Make sure that the feeder job has been specified in the string register*SR[12] JobName*.

   Note: Please note that the feeder job must be specified without the ".job" file extension.

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

   Note: Please note that the pose *PickPos* must be correctly teached for your gripping system and the part.

5. The gripper settings have been configured for your gripping system as described in the Installation and Setup chapter, and the *"FFX_PICK"* and *"FFX_PLACE"* functions in order to close and open the gripping system.

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

7. PROGRAM STRUCTURE AND SIGNAL OVERVIEW

The following section provides an overview of the signal, position and variable registers used in this sample application.

If you use this sample application as the basis for your use case, please note that if you change the register assignment, these changes must also be adapted in the *feedWareX.kl* file.

Register	Name	Description	Type
FLG[11]	ReadyForCom	Ready for communication	Input
FLG[12]	EndCom	End communication	Output
FLG[13]	Lifebit	Lifebit	Input
R[101]	Command	Command	Input/Output
R[102]	CalPntNum	Calibration point number	Input
R[103]	CalAngVal	Calibration angel value	Input
R[104]	CalXPos	Calibration X position	Input
R[105]	CalYPos	Calibration Y position	Input
R[107]	XPos	X position	Input
R[108]	YPos	Y position	Input
R[109]	RPos	Z position	Input
R[113]	xNegLim	X negative limit	Output
R[114]	xPosLim	X positive limit	Output
R[115]	yNegLim	Y negative limit	Output
R[116]	yPosLim	Y positive limit	Output
R[117]	rNegLim	R negative limit	Output
R[118]	rPosLim	R positive limit	Output
PR[51]	CalNestPre	Calibration nest pre (position)	Robot internal
PR[52]	CalFfTOfsR	Calibration toolframe offset R	Robot internal
PR[53]	CalFfUOfsZ	Calibration userframe offset Z	Robot internal
PR[54]	CalPosTmp	Calibration position temporary	Robot internal
PR[61]	PickPos	Pick position	Input
PR[62]	PickPosPre	Pick position pre (position)	Robot internal
SR[11]	CalJobName	Calibration job name	Output
SR[12]	JobName	Job name	Output
SR[13]	IpAdress	IP address (of feeder)	Output

The program file *feedWareX.kl* is of central importance. It is a so-called KAREL program - a source code in the high-level language of FANUC. It implements the following tasks.

The setting of:

- Interfaces Comments
- Communication parameters

Logics and sequences for:

- Establishing communication
- Execution of the calibration process
- Execution of production operation
- Execution of special functions

The FANUC simulation environment ROBOGUIDE is used to generate the machine code.

The following overview shows the relationships between the different program files.

Calibration

FFX_CAL_MAIN is the main program of the flexfeeder sample program for the calibration process. The following subroutines are used in it:

Program	Description
FFX_CECK_INIT_COM	Checks and initialize communication between robot and flexfeeder.
FFX_CAL_START	Starts calibration mode of the flexfeeder.
FFX_CAL_STORE_POS	Stores the current calibration position.
FFX_CAL_SEND_PART_PLACED	Send the signal to the feeder that the part has been successfully placed.
FFX_PICK	Pick action of the gripper.
FFX_PLACE	Place action of the gripper.
FFX_STOP_CYCLE	Sends the stop cycle signal to the flexfeeder.

Production

FFX_MAIN is the main program of the flexfeeder sample program for production mode. The following subroutines are used in it:

Program	Description
FFX_CECK_INIT_COM	Checks and initialize communication between robot and flexfeeder.
FFX_START_PRODUCTION	Starts production mode of the flexfeeder.
FFX_REQ_PRTS	Request a new part position from the flexfeeder.
FFX_GET_PRT_POS	Gets the requested new part position.
FFX_PICK	Pick action of the gripper.
FFX_PLACE	Place action of the gripper.
FFX_STOP_CYCLE	Sends the stop cycle signal to the flexfeeder.

Additional Programs

The following additional programs can be used to execute flexfeeder actions:

Program	Description
FFX_PURGE	Starts flexfeeder purge action.
FFX_REJECT	Starts flexfeeder reject action.
FFX_RESET	Starts flexfeeder reset action.
FFX_UNLOCK_LOCK	Locks or unlocks the dispenser cover of the flexfeeder.

8. TROUBLE SHOOTING

Display of error messages on the user screen.	In the case of errors that occur during the execution of the programs, please observe the output in the user screen. The error and debug messages displayed there are used to analyze errors and determine the cause of the error. Note: How to switch on the user screen is described in chapter Installation and Setup → *Split screen.*
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 → *flexfeeder IP address*) The IP of the feeder is displayed on its HMI on the home screen (accessible via web browser or touch screen).