Flexible F-Link Manual

Example can be found at https://support.industry.siemens.com/cs/document/109768964/configuring-flexible-f-link-communication?dti=0&lc=en-DK

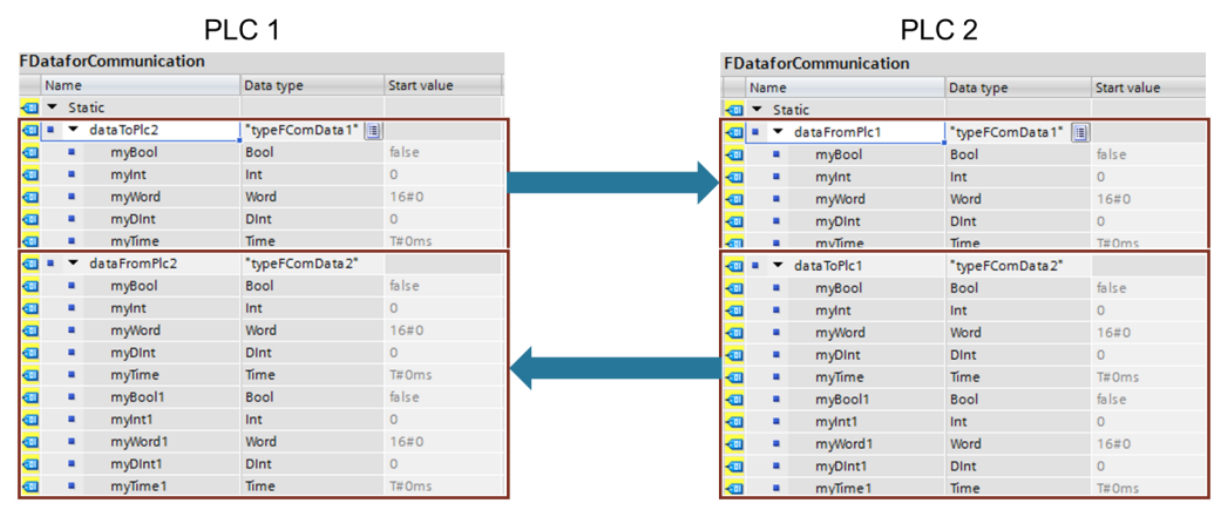
# 

# Configuring the F-Link connection

The maximum size of a safety DB that can be transferred is 100bytes.

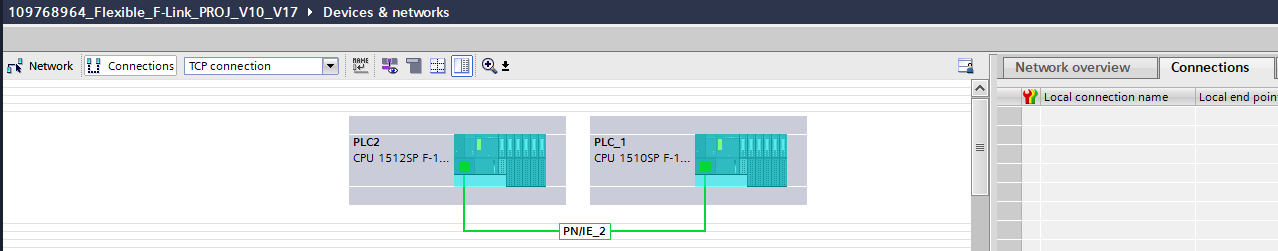
## Configuration for both controllers

1. Configure two identical databases in the PLCs.



If it is necessary to make a lot of databases these databases can be made using custom F-Compliant data types.

1. Configuration of the connections
   1. Configuration in the same project with static connections
      1. Go to Devices and networks 🡪 Network View ->> and push connections.



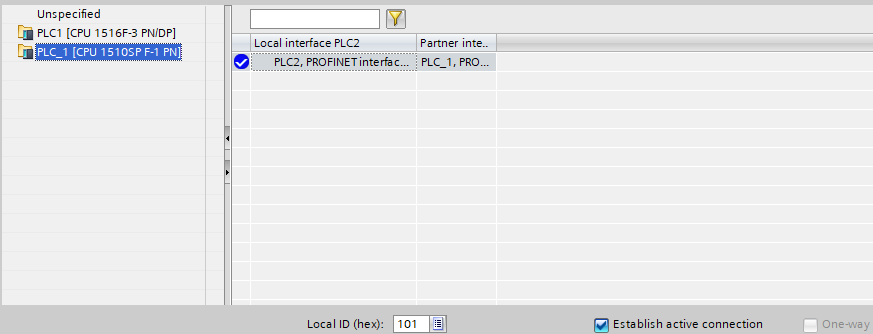
* + 1. Right Click the first PLC and push the Add new connection A screenshot of a computer

       Description automatically generated
    2. Click on the PLC you would like to make the connection too. In this example it is PLC 2

A screenshot of a computer

Description automatically generated

* + 1. Do the same for the other PLC you would like too connect.



* + 1. Der er nu oprettet forbindelse mellem de to PLCer

A screenshot of a computer

Description automatically generated

* 1. Configuration in different projects using TCON and TDICCON
     1. Two connections need to be made for the safety communication. In each of the PLCs
     2. In the first PLC the TCON function can be inserted and set up. If you need more information about this look at the manual for TCON.
     3. The First TCON is a sender whereas the second one is a receiver.

The Pictures below is from PLC 1

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

The Pictures below is from PLC2  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Creating an F-Link communication in the Safety Administration Editor

This should be done to both of the CPUS

Configuration PLC 1

1. Create an F-Link communication in the Safety Administration Editor with the direction "Send" (FLinkSendToPlc2) and one with the direction "Receive" (FLinkRcvFromPlc2). Select the appropriate data type for the data to be transferred and the transfer direction and set the F monitoring time.   
     
   To help you determine the F monitoring time, you can use the table with the S7 reaction times. The table can be found under the following link: <https://support.industry.siemens.com/cs/ww/en/view/93839056>  
     
   The largely unique F communication UUID is generated by the system itself when a communication is created.  
    

A screenshot of a computer

Description automatically generated

A close-up of a sign

Description automatically generated

Configuration PLC 2

* 1. Create an F-Link communication in the Safety Administration Editor with the direction "Receive" (FLinkRcvFromPlc1) and one with the direction "Send" (FLinkSendToPlc1). Select the appropriate data type for the data to be transferred and the desired transfer direction.
  2. Set the same F monitoring time as for PLC 1.
  3. Copy the F communication UUIDs of the F-Link communication from PLC 1. Make sure that the UUIDs are assigned correctly. This ensures the uniqueness of the transmitter/receiver recognition:   
     UUID from PLC 1 Transmit (FLinkSendToPlc2) = UUID from PLC 2 Receive (FLinkRcvFromPlc1) UUID from PLC 1 Receive (FLinkRcvFromPlc2) = UUID from PLC 2 Send (FLinkSendToPlc1)

When this is done, databases for the communication will appear in the F-Communication DBs  
A screenshot of a computer

Description automatically generated

## Programming the Flexible F-Link

The programs in both controllers are structured according to the same principle. They are subdivided into the three areas "Safety", "Send" and "Receive". The following description explains how the PLC 1 works as an example.

The following graphic shows the program structure:A diagram of a computer program

Description automatically generated

### Setting up the send and receive

##### Cyclic OB

The safety data from the other devices should be loaded into the PLC at a certain interval is done with the shown OB.  
A screenshot of a computer

Description automatically generated

Data inconsistencies may occur due to different priorities of the cyclic OB and the F-process group: If the cyclic OB has a higher priority than the F-process group, the F-program can be interrupted by the cyclic OB.

A yellow box with numbers and a black line

Description automatically generated with medium confidence

To ensure data consistency, the data is therefore first written to a receive buffer (DataToSafety.RcvBuffer). At this point, the receive buffer is a byte array with the length of the maximum expected data of a Flexible F-Link communication (100 bytes user data + 22 bytes for UUID and checksum). The data is then transferred to the security program in the preprocessing of the F process group using the non-breakable statement "UMOVE\_BLK".

#### Receiving data

##### Receiving and moving the data to buffer

Receiving the information is done in two parts. The first part is receiving t

he ACK and the second part is receiving the data.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

To receive the information from the other PLC the TRCV block is used.

ID: The ID of the communication. This is the same as the id used in the TCON connection

LEN: The ID with the expected length of information to receive

DATA: The DB where the received information should be stored

After receiving the information, it is moved to the buffer.  
The last block moves the status of the TRCV if it had a error.

All these functions are placed io the function ReceiveFData and are the same in the other PLC. If there are more PLCs, the block should be duplicated and corrected. An example of this can be found in the project.

##### Moving the data from the buffer to the Safety

In the function PreProcessingCopyDataToSafety the information is moved with UMOVE\_BLK from the Buffer to the generated SAFETY\_DB. If there is multiple PLC this will have to be duplicated and corrected.

A screenshot of a computer

Description automatically generated

##### Moving the safety DB to the last destination.

The last move is done in the Main\_Safety\_RTG1 and moved the valid data from the generated safety block to a safety DB. From hereon the information can be used in the safety program.

A screenshot of a computer

Description automatically generated

#### Sending Data

##### Moving data from the program to the safety DB

Sending the data is done in the opposite direction and starts in Main\_Safety\_RTG1. The information is moved from the functional DB to the generated safety DBs.

A screenshot of a computer

Description automatically generated

##### Sending the safety data to the other PLC

The data is not sent cyclically as the information is received. In the function SendFdata TSEND functions are used to transfer the information from the safety DBs directly. In the siemens example the TSEND function is always reset on a scan. This function is kept in the example. The first network send the Data of the transfer and after the ACK data is transmitted.  
  
A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Setup of the TSEND Connections

ID: The ID used is the same used in the TCON

DATA: The safety data that needs to be transferred. Make sure DATA and ACK have not been switched.

#### Adding the Pre and Post to the runtime

A important step to setting up the Communication is adding Pre/Post processing in the F-Runtime as shown on the pictures below. This have to be done in all the PLCs using the communication.

Pre and Post processing of the F-Runtime group in PLC1

A screenshot of a computer

Description automatically generated

Pre and Post processing of the F-Runtime group in PLC2

A screenshot of a computer

Description automatically generated