SIEMENS
Ingenuity for life

Open User Communication with TSEND_C and TRCV_C

SIMATIC S7-1200 CPU

https://support.industry.siemens.com/cs/ww/en/view/67196808

Siemens Industry Online Support

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com.

Table of contents

1 Introduction

You can use the open user communication by means of the TSEND_C and TRCV_C instructions for data exchange over the integrated PROFINET interface of the S7-1200 CPU.

In STEP 7 (TIA Portal) you will find the TSEND_C and TRCV_C instructions in the "Instructions" task card in the "Communication > Open User Communication" palette.

Table 1-1

Instruction	Description
TSEND_C	The TSEND_C instruction is executed asynchronously and has the functions below: • Configure and establish communication connection • Send data through the existing communication connection • Disconnect communication connection
TRCV_C	The TRCV_C instruction is executed asynchronously and has the functions below: • Configure and establish communication connection • Receive data through the existing communication connection • Disconnect communication connection

The protocols below are supported for this:

- ISO-on-TCP
- TCP
- UDP

Description of the sample program

The sample program was created in STEP 7 (TIA Portal V14). The project consists of one S7-1200 CPU and one S7-1500 CPU including hardware configuration and user program.

Marker byte 10 (MB10) is configured as clock marker byte in both CPUs.

The S7 program contains the call of the "TSEND_C" and "TRCV_C" instructions and the parameterization of the ISO-on-TCP connection for data exchange between the S7-1200 CPU and S7-1500 CPU. The connection parameters for establishing the ISO-on-TCP connection are saved in the "GeneralData" data block.

The user program consists of the components below.

Table 1-2

Block	Symbolic name	Description
OB100	Startup	Startup OB
OB1	Main	The functions FC100 "FC_SEND" and FC200 "FC_RECV" are called in OB1.
FC100	FC_SEND	The FC100 "FC_SEND" function calls the TSEND_C instruction internally to send data through an ISO-on-TCP connection.
FC200	FC_RECV	The FC200 "FC_RECV" function calls the TRCV_C instruction internally to receive data through an ISO-on-TCP connection.
DB3	SendData	The sent data is stored in the DB3 data block.
DB5	RecvData	The received data is stored in the DB5 data block.

2 Sample Program

2.1 OB100

The OB100 is a startup OB and is run when the S7-1200 CPU is restarted (warm start). The following variables are set to value “1”, to establish two ISO-on-TCP connections and enable the job to receive data.

- “GeneralData”.contSend
- “GeneralData”.contRecv
- “GeneralData”.enable

2.2 OB1

OB1 is called cyclically. The functions FC100 "FC_SEND" and FC200 "FC_RECV" are called in OB1.

Figure 2-1

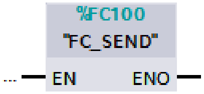
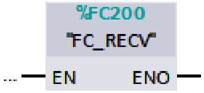


Figure 2-2

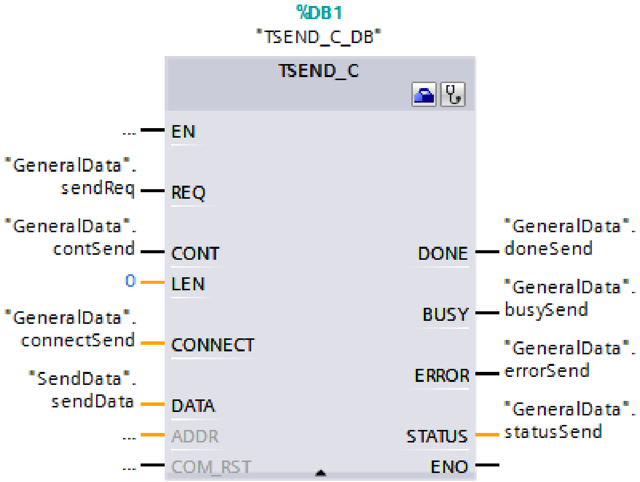


2.3 FC100 "FC_SEND"

The FC100 "FC_SEND" function calls the TSEND_C instruction internally to establish a communication connection over ISO-on-TCP, TCP or UDP and send data through the communication connection.

The following figure shows the call of the TSEND_C instruction.

Figure 2-3


%DB1
"TSEND_C_DB"
TSEND_C
... EN
"GeneralData".sendReq REQ
"GeneralData".contSend CONT
0 LEN
"GeneralData".connectSend CONNECT
"SendData".sendData DATA
... ADDR
... COM_RST
DONE "GeneralData".doneSend
BUSY "GeneralData".busySend
ERROR "GeneralData".errorSend
STATUS "GeneralData".statusSend
ENO

2.3.1 Inputs and Outputs of the TSEND_C Instruction

Inputs

The table below gives an overview of

- the inputs of the TSEND_C instruction
- the variables assigned to the inputs

Table 2-1

Input	Data type	Variable	Description
REQ	BOOL	"GeneralData".sendReq	Starts the send job at a rising edge.
CONT	BOOL	"GeneralData".contSend	Controls the communication connection: 0: Disconnects the communication connection. 1: Establishes and maintains the communication connection. If CONT=1, the TSEND_C instruction configures and establishes a communication connection. Once the connection has been configured and established, it is maintained and monitored automatically by the S7-1200 CPU.

Input	Data type	Variable	Description
LEN	UINT	0	Number of bytes that can be sent with the job.
CONNECT	VARIANT	"GeneralData".connectSend	Pointer to the structure of the connection description: • For ISO-on-TCP, use the TCON_IP_RFC system data type. For a description, refer to chapter 2.5 • For TCP, use the TCON_IP_V4 system data type. For a description, refer to chapter 2.6
DATA	VARIANT	“SendData”.sendData	Pointer to the send area that contains the address and length of the data to be sent.

Outputs

The table below gives an overview of

- the outputs of the TSEND_C instruction
- the variables assigned to the outputs

Table 2-2

Output	Data type	Variable	Description
DONE	BOOL	"GeneralData".doneSend	Status parameter with the values below: 0: Job not yet started or still being executed. 1: Job executed error-free.
BUSY	BOOL	"GeneralData".busySend	Status parameter with the values below: 0: Job not yet started or already terminated. 1: Job has not yet terminated. A new job cannot be started.
ERROR	BOOL	"GeneralData".errorSend	Status parameter with the values below: 0: No error 1: Error occurred.
STATUS	WORD	"GeneralData".statusSend	Status of the instruction

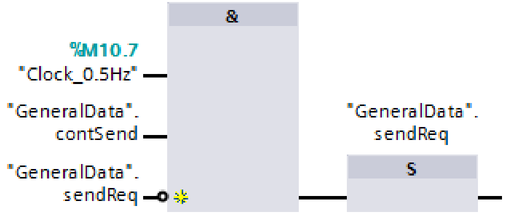
2.3.2 Start Send Job

The send job is started by the clock marker M10.7 and controlled with the "GeneralData".sendReq and "GeneralData".contSend variables.

When data is sent, the CONT input of the TSEND_C instruction must be set to the value "1" in order to establish the communication connection. The CONT input is set to the value "1" by means of the "GeneralData".contSend variable. This means that when "GeneralData".contSend is set to the value "1", the send job can be started.

When the send job is running, "GeneralData".reqSend is set to the value "1", which means that no new send job can be started as long as this send job is running.

Figure 2-4



2.3.3 Establish and Maintain the Communication Connection

The "GeneralData".contSend variable is set permanently to the value "1" when the S7-1200 CPU is restarted (warm restart).

The CONT input of the TSEND_C instruction is set permanently to the value "1" by means of the "GeneralData".contSend variable in order to establish and maintain the communication connection.

If the S7-1200 CPU goes into STOP mode, the existing communication connection is aborted and the configured communication connection is removed. You must once again execute the TSEND_C instruction to reconfigure and re-establish the communication connection.

2.3.4 LEN

A send area with optimized access is used at the DATA parameter in this sample program. For this reason the value “0” is used at the LEN parameter.

2.3.5 Send Area

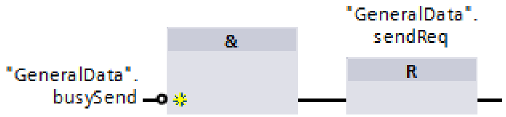
In this sample program, the send area “SendData”sendData is defined with optimized access.

The send area is 100 bytes long and the data to be sent is contained in data block DB3.

2.3.6 Reset "GeneralData".sendReq

If there is no send job running, the BUSY output of the TSEND_C instruction has the value "0" and "GeneralData".sendReq is reset to the value "0". This means that a new send job can only be triggered once the previous job has been completed.

Figure 2-5



2.3.7 Save STATUS of the TSEND_C Instruction

If the TSEND_C instruction is executed successfully or with errors, the status of the TSEND_C instruction is saved in the "GeneralData".statusSendSave variable. The status informs you of the cause if the send job is not running.

Figure 2-6

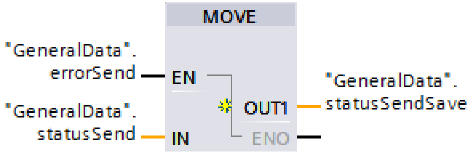
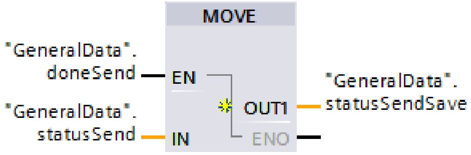


Figure 2-7

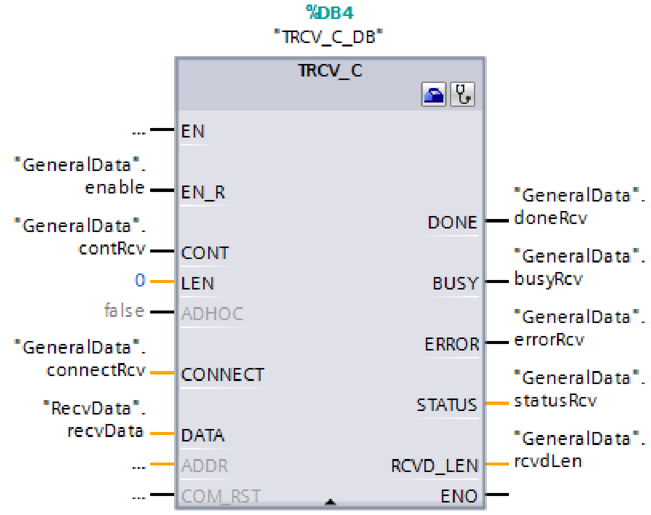


2.4 FC200 "FC_RECV"

The FC200 "FC_RECV" function calls the TRCV_C instruction to establish a communication connection over ISO-on-TCP, TCP or UDP and receive data through the communication connection.

The following figure shows the call of the TRCV_C instruction.

Figure 2-8



2.4.1 Inputs and Outputs of the TRCV_C Instruction

Inputs

The table below gives an overview of

- the inputs of the TRCV_C instruction
- the variables assigned to the inputs

Table 2-3

Input	Data type	Variable	Description
EN_R	BOOL	"GeneralData".enable	Enable receive with EN_R = 1
CONT	BOOL	"GeneralData".contRcv	Controls the communication connection: 0: Disconnects the communication connection. 1: Establishes and maintains the communication connection. If CONT=1, the TRCV_C instruction configures and establishes a communication connection. Once the connection has been configured and established, it is maintained and monitored automatically by the S7-1200 CPU.

Input	Data type	Variable	Description
LEN	UINT	0	Maximum number of bytes that can be sent with the job. **Note** If you use a receive area with optimized access at the DATA parameter, the value “0” must be used at the LEN parameter.
CONNECT	VARIANT	“GeneralData”.connectRcv	Pointer to the structure of the connection description: • For ISO-on-TCP, use the TCON_IP_RFC system data type. For a description, refer to chapter 2.5 • For TCP, use the TCON_IP_V4 system data type. For a description, refer to chapter 2.6
DATA	VARIANT	“RecvData”.recvData	Pointer to the receive area that contains the address and length of the data to be received.

Outputs

The table below gives an overview of

- the outputs of the TRCV_C instruction
- the variables assigned to the outputs

Table 2-4

Output	Data type	Variable	Description
DONE	BOOL	"GeneralData".doneRcv	Status parameter with the values below: 0: Job not yet started or still being executed. 1: Job executed error-free.
BUSY	BOOL	"GeneralData".busyRcv	Status parameter with the values below: 0: Job not yet started or already terminated. 1: Job has not yet terminated. A new job cannot be started.
ERROR	BOOL	"GeneralData".errorRcv	Status parameter with the values below: 0: No error 1: Error occurred.
STATUS	WORD	"GeneralData".statusRcv	Status of the instruction
RCVD_LEN	UINT	"GeneralData".rcvdLen	Volume of data actually received in bytes.

2.4.2 Enable Receiving of Data

The "GeneralData".enable variable is set permanently to the value "1" when the S7-1200 CPU is restarted (warm restart). The EN_R input of the TRCV_C instruction is set permanently to the value "1" by means of the "GeneralData".enable variable in order to enable receiving of data.

2.4.3 Establish and Maintain the Communication Connection

The "GeneralData".contRcv variable is set permanently to the value "1" when the S7-1200 CPU is restarted (warm restart). The CONT input of the TRCV_C instruction is set permanently to the value "1" by means of the "GeneralData".contRcv variable in order to establish and maintain the communication connection.

If the S7-1200 CPU goes into STOP mode, the existing communication connection is aborted and the configured communication connection is removed. You must once again execute the TRCV_C instruction to reconfigure and re-establish the communication connection.

2.4.4 LEN

A receive area with optimized access is used at the DATA parameter in this sample program. For this reason the value “0” is used at the LEN parameter.

2.4.5 Receive Area

In this sample program, the receive area “RecvData”.recvData is defined with optimized access.

The receive area is 100 bytes long and the data received is contained in data block DB5.

2.4.6 Save STATUS of the TRCV_C Instruction

If the TRCV_C instruction is executed successfully or with errors, the status of the TRCV_C instruction is saved in the "GeneralData".statusRcvSave variable. The status informs you of the cause if the data is not received successfully.

Figure 2-9

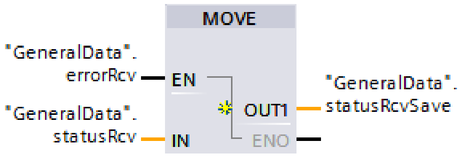
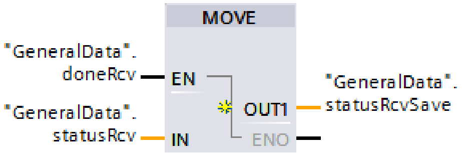


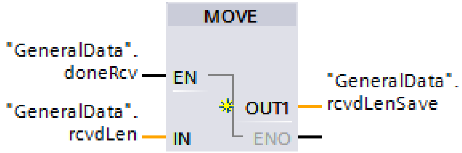
Figure 2-10



2.4.7 Save Length of the Data Received

If the TRCV_C instruction is executed successfully, the length of the data received is saved in the "GeneralData".rcvdLenSave variable.

Figure 2-11



2.5 Connection Parameters with Structure according to TCON_IP_RFC

A connection description DB with a structure according to TCON_IP_RFC is used for CPUs of S7-1200 V4.0 and higher and S7-1500 to assign parameters to ISO-on-TCP communication connections. The fixed data structure of the TCON_IP_RFC contains all parameters that are required to establish the connection.

The following table shows the structure of the connection description according to TCON_IP_RFC.

Table 2-5

Byte	Parameter	Data Type	Value	Description
0 and 1	InterfaceId	HW_ANY	64	Hardware identifier of the integrated PROFINET interface of the CPU
2 and 3	ID	CONN_OUC	1	Reference to this connection (value range: 1 to 4095). You must specify the value of this parameter for the TSEND_C and TRCV_C instruction under ID.
4	ConnectionType	BYTE	16#0C	Connection type: • 12: ISO-on-TCP (16#0C hex = 12 dez)
5	ActiveEstablished	BOOL	False	Identifier for the type of connection establishment • False: passive connection establishment • True: active connection establishment

Byte	Parameter	Data Type	Value	Description
6 to 9	RemoteAddress	ARRAY [1..4] of BYTE	-	IP address of the partner endpoint, e. g. for 192.168.0.3 • addr[1] = 192 • addr[2] = 168 • addr[3] = 0 • addr[4] = 3
10 to 43	RemoteTSelector	TSelector	-	TSelector of the remote connection partner: • TSelLength = value range 0 to 32 as UINT • TSel[1-32] = value range each 0 to 255 in bytes
44 to 77	LocalTSelector	TSelector	-	TSelector of the local connection partner: • TSelLength = value range 0 to 32 as UINT • TSel[1-32] = value range each 0 to 255 in bytes

NOTE

You have to use different connection numbers for the TSEND_C and TRCV_C. Otherwise two connections with the same connection number will be establish.

The same TSelector is used for the remote and local connection partner in this sample program.

2.6 Connection Parameters with Structure according to TCON_IP_V4

A connection description DB with a structure according to TCON_IP_V4 is used for CPUs of S7-1200 V4.0 and higher and S7-1500 to assign parameters to TCP or UDP communication connections. The fixed data structure of the TCON_IP_V4 contains all parameters that are required to establish the connection.

The following table shows the structure of the connection description according to TCON_IP_V4.

Table 2-6

Byte	Parameter	Data Type	Value	Description
0 and 1	InterfaceId	HW_ANY	64	Hardware identifier of the integrated PROFINET interface of the CPU
2 and 3	ID	CONN_OUC	1	Reference to this connection (value range: 1 to 4095). You must specify the value of this parameter for the TSEND_C and TRCV_C instruction under ID.

Byte	Parameter	Data Type	Value	Description
4	ConnectionType	BYTE	16#0B	Connection type: • 11: TCP (16#0B hex = 11 dez) • 19: UDP (16#13 hex = 19 dez)
5	ActiveEstablished	BOOL	False	Identifier for the type of connection establishment • False: passive connection establishment • True: active connection establishment
6 to 9	RemoteAddress	ARRAY [1..4] of BYTE	-	IP address of the partner endpoint, e. g. for 192.168.0.3 • addr[1] = 192 • addr[2] = 168 • addr[3] = 0 • addr[4] = 3
10 and 11	RemotePort	UINT	2000	Port address of the remote connection partner (value range: 1 to 49151)
12 and 13	LocalPort	UINT	2000	• Port address of the local connection partner (value range: 1 to 49151)

NOTE

You have to use different connection numbers for the TSEND_C and TRCV_C. Otherwise two connections with the same connection number will be establish.

The same port address is used for the remote and local connection partner in this sample program.