

FB Job_List_SCL and FB Job_List_STL

General

With the function block Job_List there is an additional block for MODBUS/TCP client communication available. The block is available in SCL and STL.

The functionality of both blocks is the same.

Principle of operation

By means of the block "Job_List" a job list for the Modbus/TCP blocks MODBUSPN and MODBUSCP is realizable. It provides the possibility to start different Modbus jobs cyclically.

The outputs *QDATA_TYPE*, *QSTART_ADDRESS*, *QLENGTH* and *QWRITE_READ* have to be connected to the corresponding inputs of the Modbus/TCP block. The outputs *DONE* and *ERROR* of the Modbus/TCP block have to be connected to the block Job_List as well.

The different jobs resp. requests are parameterized at the inputs *Job1_x* to *Job5_x*.

The time at the input *CYCLICAL* defines the time interval in milliseconds in which the job list is executed cyclically. When the time *CYCLICAL* elapses, the jobs parameterized at *Job_x* are executed sequentially.

If the time *CYCLICAL* elapses while the job list is executed, the information A089 is displayed at the output *STATUS*. The execution of the jobs already running is carried on. As soon as the last job is finished, the job list is started immediately with the first job .

With *CYCLICAL* = 0ms the job list is not executed. When the time *CYCLICAL* is set to 0ms during runtime, the actual executed job list will be completed. After the last job is finished the execution of the job list is stopped.

By setting *Jobx_DATA_TYPE* = 0 a job can be skipped.

With *ABORT* = TRUE the job list is cancelled. The running job will be completed, no further jobs are started until the time *CYCLICAL* elapses. Then the job list starts with the first job.

It is recommended to set *ABORT* in OB100 so that the FB "Job_List" is reset when the CPU is restarted.

The function block provides 5 jobs by default. The number of jobs can be increased if necessary. To achieve this, the following changes are mandatory:

- Open the source of the job list block.
- Copy the inputs *Job5_DATA_TYPE*, *Job5_START_ADDRESS*, *Job5_LENGTH* and *Job5_WRITE_READ* and insert them below job5.
- Rename the inputs to *Job6_DATA_TYPE*, *Job6_START_ADDRESS*, *Job6_LENGTH* and *Job6_WRITE_READ* etc.
- Adjust the static variable *Count_of_Jobs* according to your changes.

Input parameters

Parameter	Data type	Description
CYCLICAL	TIME	> 0ms: Cyclical execution of the job list
DONE	BOOL	Positive acknowledgement of the Modbus/TCP block
ERROR	BOOL	Negative acknowledgement of the Modbus/TCP block
Job1_DATA_TYPE	BYTE	1. Job: Data type, 0 = not carried out
Job1_START_ADDRESS	WORD	1. Job: Start address
Job1_LENGTH	WORD	1. Job: Length
Job1_WRITE_READ	BOOL	1. Job: Write/read
Job2_DATA_TYPE	BYTE	2. Job: Data type, 0 = not carried out
Job2_START_ADDRESS	WORD	2. Job: Start address
Job2_LENGTH	WORD	2. Job: Length
Job2_WRITE_READ	BOOL	2. Job: Write/read
Job3_DATA_TYPE	BYTE	3. Job: Data type, 0 = not carried out
Job3_START_ADDRESS	WORD	3. Job: Start address
Job3_LENGTH	WORD	3. Job: Length
Job3_WRITE_READ	BOOL	3. Job: Write/read
Job4_DATA_TYPE	BYTE	4. Job: Data type, 0 = not carried out
Job4_START_ADDRESS	WORD	4. Job: Start address
Job4_LENGTH	WORD	4. Job: Length
Job4_WRITE_READ	BOOL	4. Job: Write/read
Job5_DATA_TYPE	BYTE	5. Job: Data type, 0 = not carried out
Job5_START_ADDRESS	WORD	5. Job: Start address
Job5_LENGTH	WORD	5. Job: Length
Job5_WRITE_READ	BOOL	5. Job: Write/read
ABORT	BOOL	TRUE: Cancel the running job list

Output parameters

Parameter	Data type	Description
QENQ	BOOL	Start the execution of the Modbus/TCP block
QDATA_TYPE	BYTE	DATA_TYPE of the current job
QSTART_ADDRESS	WORD	START_ADDRESS of the current job
QLENGTH	WORD	LENGTH of the current job
QWRITE_READ	BOOL	WRITE_READ of the current job
BUSY	BOOL	Job list is running
STATUS	WORD	Status information of the block

Status information

Status	Meaning	Remedy / notes
16#A089	The parameterized time CYCLICAL has elapsed while the job list is still in work.	The job list starts immediately after the last job of the previous list was finished.

Example

