MelsecQJ-Link / MelsecQR2-Link

DLL - Library to communicate with Mistubishi Melseq Q-Series
Version 2.0.4.7

Operating system

Windows

- 10
- 8
- 7
- Vista
- XP

Programming languages

- C/C++
- VB
- C#
- VB.net
- Delphi
- Excel

Functionality

MelsecQJ-Link / MelsecRQ2-Link provide access for C / C ++, Delphi etc. to the Mitsubishi Melsec-Q series PLCs.
QJ-Link works over Ethernet (TCP / IP) - MelsecRQ2-Link uses the serial connection via RS232.

The library provides functions for reading and writing.

Data area

Code	PLC-Memory	AccessMode
'X'	Input relay	bit
'Y'	Output relay	bit
'B'	Link relay	bit
'M'	Internal relay	bit
'm'	Special relay	bit (! note the notation of the small 'm')
'W'	Link register	word / dword / float
'D'	Data register	word / dword / float
'R'	File register	word / dword / float
'w'	Special link register	word / dword / float
'T'	TS Timer Contact	word
't'	TC Timer Coil	Bit
'I'	TN Timer current value	word
'S'	SS Retenitive Timer Contact	word
's'	SC Retenitive Timer Coil	Bit
'i'	SN Retenitive Timer current Value	word
'C'	CS Counter Contact	word

Code	PLC-Memory	AccessMode
'c'	CC Counter Coil	Bit
'J'	CNCounter current value	word
'L'	Latch relay	bit (! only Q-Series)

Return Values

Only the function “MQR2GetLastErrCode” returns a special value from internal data.
All other functions return a result.

A result with the value of zero means, the everything is okay.
If the value is other than 0 you have to check the return code as follows:

MQR2 - error values

Name	value	meaning
MQR2_E_NOERR	0	everything OK
MQR2_E_TIMEOUT	-1	timeout error occurred
MQR2_E_BADBAUD	-2	for the selected COM, here was elected a different Baud-rate before“;
MQR2_E_NOCOM	-4	The selected COM port does not exist or is already in use
MQR2_E_BADCHAR	-3	receive error, I received a bad character.
MQR2_E_GENERAL	-5	general error, (should not occur)
MQR2_E_NODATA	-6	data area in PLC not available, e.g. you tried to read M 30000
MQR2_E_DATACNT	-7	protocol error. The count of nett data is corrupted. (should not occur)
MQR2_E_SIZE	-8	You tried to read/write to much data on block. The greatest block is 960 16-Bit words
MQR2_E_ZEROSIZE	-9	You tried to read/write a block with the length of zero. The Count parameter is set to 0!
MQR2_E_DATATYP	-10	you have selected an other type than above described
MQR2_E_UNKNOWN_ERRCODE	-11	I received an unknown ErrorCode. Please call 'MQR2GetLastErrCode' and report the value to the developer of the driver
MQR2_E_BADREF	-99	the parameter <Ref> is invalid or to the PLC manufacturer.

When using QJ functions the following errors can appear.

MQJ - error values

MQJ_E_NOERR	eq. to MQR2_E_NOERR
MQJ_E_TIMEOUT	eq. to MQR2_E_TIMEOUT
MQJ_E_GENERAL	eq. to MQR2_E_GENERAL
MQJ_E_NODATA	eq. to MQR2_E_NODATA
MQJ_E_DATACNT	eq. to MQR2_E_DATACNT
MQJ_E_SIZE	eq. to MQR2_E_SIZE
MQJ_E_ZEROSIZE	eq. to MQR2_E_ZEROSIZE
MQJ_E_DATATYP	eq. to MQR2_E_DATATYP
MQJ_E_UNKNOWN_ERRCODE	eq. to MQR2_E_UNKNOWN_ERRCODE
MQJ_E_BADREF	eq. to MQR2_E_BADREF
MQJ_E_DEMOEND	eq. to MQR2_E_DEMOEND
MQJ_E_NOMEM	-12 non memory available

MQJ_E_NOCON_AVAIL	-13 the maximal count (64) of connections are reached max = 256
MQJ_E_SOCKERR	-14 Socket error happened, check MQJGetSockErr (long Ref)
MQJ_E_CHECK_ERRCODE	-15 check Error-code in user manual, appended in as pdf-File, to get Error code call MQJGetLastErrCode (Ref);

Funktionen

Open

Opens the serial Com Port for the connection specified by the parameters.

MQR2Open

```
long WINAPI
MQR2Open (DWORD Com, DWORD PLCType, DWORD Timeout, DWORD StationNr, DWORD NetworkNr, DWORD
PCNr, DWORD Baudrate, DWORD Parity, DWORD StopBits);
```

Parameter

Name	Purpose
Com	number of the comport the device name is normally created with “sprintf (ComDev, ”/dev/ttyS%d”, Com)“. If you want to use an other device name for this COM-Port, you can call 'MQR2SetDevName' before to register an other device name for the selected port
PLCType	0 = normal Q-PLC, 1 = A-PLC
Timeout	Timeoutvalue in ms to wait for response of the PLC normally 3000 ms
StationNr	station number of the PLC (normally 0, see configuration of the PLC)
NetworkNr	Network number of the PLC (normally 0, see configuration of the PLC)
PCNr	normally 0xFF, (not tested with other values)
Baudrate	the baud rate, can be: 1200, 1800, 2400, 4800, 9600,19200, 38400,57600, 115200 depends on the PLC
Parity	can be 'N' for none, 'O' for odd, and 'E' for even
StopBits	count of Stopbits (1 or 2)

Result

The result value is the reference for calling all further functions.
If the value is negative (less than 0). Check the error codes above.
If the value is greater or equal to zero, all is OK.
You can open up to 16 channels.

MQJOpen

```
long WINAPI
MQJOpen (LPCSTR IPAdr, DWORD Port, DWORD PLCType, DWORD NetworkNr, DWORD PCNr, DWORD
RxTimeout, DWORD TxTimeout, DWORD ConnectTimeout);
```

Name	meaning
IPAdr	IP Address of the QJ71-71 device e.g. 10.0.0.200
Port	Port number of the TCP/IP-connection normally use 5002 (!) here.
PLCType	0 = normal Q-PLC, 1 = A-PLC
NetWorkNr	Network number normally use 0
PCNr	normally 0xFF, (not tested with other values)
RxTimeout	Timeoutvalue in ms to wait for response of the PLC normally 5000 ms

Name	meaning
TxTimeout	Timeoutvalue in ms to wait for sending TCP/IP packets
ConnectTimeout	Timeoutvalue in ms to wait for the TCP/IP connection becomes established

Result

The result value is the reference for calling all further functions.
If the value is negative (less than 0), check the error codes above.
If the value is greater or equal to zero, all is OK.
You can open up to 16 channels.

Close

Closes the COM-Port referenced by <Ref>.

MQR2Close

```
long WINAPI
MQR2Close (long Ref);
```

MQJClose

```
long WINAPI
MQJClose (long Ref);
```

Result

For the result, check the error code described above.

Read / Write

Word

Read or write words from the PLC

MQR2RdW / MQR2WrW

```
//Read
long WINAPI
MQR2RdW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPWORD Buffer);

//Write
long WINAPI
MQR2WrW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPWORD Buffer);
```

MQJRdW / MQJWrW

```
//Read
long WINAPI
MQJRdW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPWORD Buffer);

//Write
long WINAPI
MQJWrW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPWORD Buffer);
```

Name	meaning
Ref	the reference generated by the “Open function”

Name	meaning
Type	Data type, see the list above 'M', 'X' and so on, see data area
Start	number of the first element, that should be read
Cnt	the count of elements to be read. Note your <Buffer> must be big enough!
Buffer	pointer to the memory in the PC. normally this is an array of elements

Result

For the result, check the error code described above.

DWORD

Read or write double words from the PLC
The Parameter description see “MQR2RdW”.

MQR2RdDW / MQR2WrDW

```
//Read
long WINAPI
MQR2RdDW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPDWORD Buffer);

//Write
long WINAPI
MQR2WrDW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPDWORD Buffer);
```

MQJRdDW / MQJWrDW

```
//Read
long WINAPI
MQJRdDW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPDWORD Buffer);

//Write
long WINAPI
MQJWrDW (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPDWORD Buffer);
```

REAL

Read or write doubles from the PLC
The Parameter description see “MQR2RdW”.

MQR2RdReal / MQR2WrReal

```
//Read
long WINAPI
MQR2RdReal (long Ref, DWORD Type, DWORD Start, DWORD Cnt, double *Buffer);

//Write
long WINAPI
MQR2WrReal (long Ref, DWORD Type, DWORD Start, DWORD Cnt, double *Buffer);
```

MQJRdReal / MQJWrReal

```
//Read
long WINAPI
MQJRdReal (long Ref, DWORD Type, DWORD Start, DWORD Cnt, double *Buffer);

//Write
long WINAPI
MQJWrReal (long Ref, DWORD Type, DWORD Start, DWORD Cnt, double *Buffer);
```

BIT

Read or write bit (boolean) from the PLC
The Parameter description see “MQR2RdW”.
Every Byte represents one bit. 1 = Bit is set. 0 = bit is not set. e.g.

```
  BYTE Buffer[10];

  MQR2RdBit (Ref, 'M', , 10, Buffer);

  // Read
  now: Buffer[] = M0
       Buffer[1] = M1
       Buffer[2] = M2 and so on.


   // Write
   to set  M1: Buffer[1] = 1;
   to reset M2: Buffer[2] = ;

   MQR2WrBit (Ref, 'M', , 10, Buffer);
```

MQR2RdBit / MQR2WrBit

```
//Read
long WINAPI
MQR2RdBit (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPBYTE Buffer);

//Write
long WINAPI
MQR2WrBit (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPBYTE Buffer);
```

MQJRdBit / MQJWrBit

```
//Read
long WINAPI
MQJRdBit (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPBYTE Buffer);

//Write
long WINAPI
MQJWrBit (long Ref, DWORD Type, DWORD Start, DWORD Cnt, LPBYTE Buffer);
```

GetLastErrCode

This function can be called to determine the last error code in the protocol.
See the error list above!
Call it, when you get the result MQJ_E_CHECK_ERRCODE or MQR2_E_UNKNOWN_ERRCODE

MQR2GetLastErrCode

```
long WINAPI
MQR2GetLastErrCode (long Ref);
```

MQJGetLastErrCode

```
long WINAPI
MQJGetLastErrCode (long Ref);
```

SetDevName

The device name normally is created with “sprintf (ComDev, ”/dev/ttyS%d”, Com)“.

MQR2SetDevName

```
int
MQR2SetDevName (DWORD Com, const char *DevName);
```

If you want to use an other device name for this COM-Port, you can call 'MQR2SetDevName' before to register an other device name for the selected port.

e.g. to change the name to ”/dev/tty/mytty“ on the port COM1 call:

```
MQR2SetDevName (COM1,  "/dev/tty/mytty");
```

Now all Open-calls with COM1 are using these device.

Result

A Result -1 means, you have selected a COM number greater than 15!

MQJGetSockErr

```
long WINAPI
MQJGetSockErr (long Ref);

#define MQJ_E_NOERR       MQR2_E_NOERR
#define MQJ_E_TIMEOUT     MQR2_E_TIMEOUT
#define MQJ_E_BADBAUD     MQR2_E_BADBAUD
#define MQJ_E_NOCOM       MQR2_E_NOCOM
#define MQJ_E_BADCHAR     MQR2_E_BADCHAR
#define MQJ_E_GENERAL     MQR2_E_GENERAL
#define MQJ_E_NODATA      MQR2_E_NODATA
#define MQJ_E_DATACNT     MQR2_E_DATACNT
#define MQJ_E_SIZE        MQR2_E_SIZE
#define MQJ_E_ZEROSIZE    MQR2_E_ZEROSIZE
#define MQJ_E_DATATYP     MQR2_E_DATATYP
#define MQJ_E_UNKNOWN_ERRCODE  MQR2_E_UNKNOWN_ERRCODE
#define MQJ_E_BADREF     MQR2_E_BADREF
#define MQJ_E_DEMOEND    MQR2_E_DEMOEND

#define MQJ_E_NOMEM         -12             non memory available
#define MQJ_E_NOCON_AVAIL   -13             the maximal count of connections are reached
max = 256
#define MQJ_E_SOCKERR       -14             Socket error happened, check
#define MQJ_E_CHECK_ERRCODE -15             see complete codes user manual Chapter 11.3,
appended in as pdf-File,
```

Q-PLC Error codes / complete codes

This section explains the end codes (complete codes) that are added to responses.

End code	Description	Processing
00_H	Normal completion	—
02_H	Designation of device range of devices to be read/written from/to is incorrect	Check and correct the designated head device and number of points
50_H	Codes for command/response type of sub-header are not within the specifications In communication using the fixed buffer, if the data length setting is less than the actual data count, the remaining data is determined as the second data and processed. In this case, a subheader undefined command type error may occur	Check and correct command/response type set by an external device. (The Ethernet module automatically adds command/response type; the user does not need to set these.) Check and correct the data length
51_H	In communication using the random access buffer, the head address designated by an external device is set outside the range from 0 to 6143	Check and correct the designated head address
54_H	When “ASCII code communication” is selected in [Operational settings] – [Communication data code] with GX Developer, ASCII code data that cannot be converted to binary code was received from an external device	Check and correct the send data of the external device
55_H	When [Operational setting] – [Enable Write at RUN time] is set to disable (no check mark) with GX Developer, an external device requests to write data while the PLC CPU is in the RUN status. An external device requests writing a parameter, sequence program or microcomputer program while the PLC CPU is in the RUN status. (Not related to the settings in [Operational settings] – [Enable Write at RUN time with GX Developer])	Write data by setting [Enable Write at RUN time] to enable (with check mark). However, writing a parameter, sequence program, or microcomputer program is not allowed while the CPU is in the RUN status. Write such data by placing the PLC CPU in the STOP status
56_H	Device designation from an external side is incorrect	Correct the device designated
57_H	The number of points for a command designated by an external device exceeds the maximum number of processing points (number of processing that can be executed per communication) for each processing Addresses from the head address (head device number and head step number) to the designated points exceed the maximum addresses (device number and step number)	Correct the designated points or the head address (device number and step number)
	Byte length of a command does not conform to the specifications When writing data, the set number of data points written is different from the value of the designated number	Check the data length of the command and redo the data setting
	Monitoring was requested even though monitor data is not registered	Register the monitor data
	When reading/writing in a microcomputer program, an address after the end of parameter setting range is designated	Addresses after the last address cannot be read/written from/to. Correct the address designated
	In the block number designation of the extension file register, a block number exceeding the range of corresponding memory cassette size is designated	Correct the block number

End code	Description	Processing
58_H	A head address (head device number and head step number) of a command designated by an external device is set outside the range that can be designated. When reading from/writing to a microcomputer program or file register (R), values exceeding the PLC CPU's parameter setting range was designated	Designate the appropriate values within the range that are allowed for each processing
	A block number designated for an extension file register does not exist	Correct the block number
	Cannot designate a file register (R)	Check the device
	A word device is designated for a command for bit devices. The head number of bit devices is designated by a value other than a multiple of 16 in a command for word devices	Correct the command or the designated device
59_H	Cannot designate an extension file register	Check the device
$5B_H$	The PLC CPU and the Ethernet module cannot communicate. The PLC CPU cannot process requests from an external device	Fix the faulty parts by referring to the abnormal codes appended to the end codes
60_H	Communication time between the Ethernet module and the PLC CPU exceeded CPU monitoring timer value	Increase the CPU monitoring timer value

A-PLC Error codes / complete codes

Error Code	Error	Description error	Corrective action
10_H	PC number error (PLC number error)	A station with the specified PC number does not exist. (1) The PC number designated with a command is neither "FF" of the local station nor any of the station numbers designated with the MELSECNET link parameters	(1) Change the PC number to "FF" of the local station or a station number set in the link parameter, and communicate again
11_H	Mode error	Poor communication between the Ethernet module and the PLC CPU (1) After the Ethernet module receives a request successfully from an external device, the Ethernet module and the PLC CPU could not communicate for some reason (noise, etc.)	(1) Communicate again. If an error occurs again, check noise, etc. and replace the Ethernet module, then communicate again
12_H	Intelligent function module designation error	Intelligent function module error (1) The intelligent function module with the buffer memory that can be communicated with, does not exist at the location designated by the intelligent function module number. (For example, the corresponding place is for an input/output module or an empty slot.)	(1) Change the data content designated in the control procedure or change the installation location of the intelligent function module, and communicate again

Error Code	Error	Description error	Corrective action
18_H	Remote error	Remote RUN/STOP not accessible. Another module (other Ethernet module, etc.) has already executed remote STOP/PAUSE	(1) Check whether or not other module has executed remote STOP/PAUSE. If one of these commands has been executed, cancel it and communicate again
$1F_H$	Device error	Invalid device specification	(1) Review the specified device. (2) Does not access any non-existent device
20_H	Link error	The CPU module of the request destination is disconnected from the data link	Check whether or not the PLC CPU with the station number set as PC number is disconnected. Remove the cause of disconnection and connect again
21_H	Intelligent function module bus error	Cannot access the memory of the intelligent function module. (1) The control bus to the intelligent function module is faulty. (2) The intelligent function module is faulty	One of the following hardware is faulty: the PLC CPU, base unit, intelligent function module, or Ethernet module. Please consult your nearest dealer

Table of Contents