

MelsecQJ-Link / MelsecQR2-Link

DLL - Library zur Kommunikation mit Mistubishi Melseq Q-Serie
Version 2.0.4.7

© **by TIS**

Betriebssystem

Windows

- 10
- 8
- 7
- Vista
- XP

Programmiersprachen

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

Funktionsweise

MelsecQJ-Link / MelsecRQ2-Link ermöglichen den Zugriff für C/C++, Delphi etc. auf die SPSen der Mitsubishi Melsec-Q-Serie.

QJ-Link arbeitet über Ethernet (TCP/IP) - MelsecRQ2-Link verwendet die serielle Verbindung über RS232.

Die Library stellt Funktionen zum Lesen und Schreiben bereit.

Datenbereiche

Code	PLC-Memory	AccessMode	AddressMode	PLC-Syntax	read/write
'X'	X Input relay	bit	Hex	X xxxxx	r
'Y'	Y Output relay	bit	Hex	Y xxxxx	r
'B'	B Link relay	bit	Hex	B xxxxx	r/w
'M'	M Internal relay	bit	Dec	M ddddd	r/w
'm'	SM Special relay (only Q-Series)	bit	Dec	SM ddddd	r
'W'	W Link register	word / dword / float	Hex	W xxxxx	r/w
'D'	D Data register	word / dword / float	Dec	D ddddd	r/w
'R'	R File register	word / dword / float	Dec	R ddddd	r/w
'w'	SW Special link register	word / dword / float	Hex	SW xxxxx	r/w
'T'	TS Timer Contact	word	Dec	TS ddddd	r/w
't'	TC Timer Coil	Bit	Dec	TC ddddd	r/w
'I'	TN Timer current value	word	Dec	TN ddddd	r/w
'S'	SS Retentive Timer Contact	word	Dec	SS ddddd	r/w
's'	SC Retentive Timer Coil	Bit	Dec	SC ddddd	r/w
'i'	SN Retentive Timer current Value	word	Dec	SN ddddd	r/w
'C'	CS Counter Contact	word	Dec	CS ddddd	r/w

Code	PLC-Memory	AccessMode	AddressMode	PLC-Syntax	read/write
'c'	CC Counter Coil	Bit	Dec	CC ddddd	r/w
'j'	CN Counter current value	word	Dec	CN ddddd	r/w
'L'	L Latch relay (only Q-Series)	bit	Dec	L ddddd	r/w

AddressMode:

- Hex = hexadezimale Operandennummer
 - z.B Eingang 20 = X 14
- Dec = dezimale Operandennummer
 - z.B. Merker 20 = M 20

Syntax Bitzugriff bei Nicht-Bit Operanden
 z.B Datenregister 'D' Adresse 10, Bit 8 = D 10.8



Return Values

Nur die Funktion „MQR2GetLastErrCode“ gibt einen speziellen Wert aus internen Daten zurück.
 Alle anderen Funktionen geben ein Ergebnis zurück.

Ein Ergebnis mit dem Wert 0 bedeutet, dass alles in Ordnung ist. Wenn der Wert nicht 0 ist, müssen Sie den Rückgabecode wie folgt überprüfen:

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.

Bei Verwendung von QJ-Funktionen können folgende Fehler auftreten.

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 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 the Protocol Error code call MQJGetLastErrCode (Ref);

Voraussetzungen

Wir verwenden das **MC Protokoll** für die Verbindung zur SPS. Dieses muss in der Entwicklungssoftware aktiviert werden.

Folgende Möglichkeiten gibt es:

Build-In Ethernet

Öffnen Sie in Ihrem **Projekt** bei **Parameter** die **SPS-Einstellungen**



Im Reiter **Built-In Ethernet Port Setting**

- stellen Sie die gewünschte IP-Adresse ein, z.B. 192.168.0.57
- aktivieren Sie **Enable online change(FTP, MC Protocol)**
- Klicken Sie anschließend auf 
- Im geöffneten **Built-In Ethernet Port Open Setting** wählen Sie eine Verbindung aus und
 - ändern diese auf **MC Protocol**
 - Port eintragen z.B. 5100



Ethernet-Module QJ71E71

In jedem Fall sind diese Einstellungen vorzunehmen:



Bei dem QJ71E71 Ethernet Modul gibt zwei Möglichkeiten das **MC Protokoll** zu verwenden:

1. QJ71E71_MC Protokoll mit Standard-Port 5001

Wenn auf dem Modul nichts eingestellt ist, wird über den Standard Port 5001 kommuniziert



2. QJ71E71_MC Protokoll mit Port

Wenn TCP mit Port beliebig (z.B 1280) verwendet werden soll, dann diese Einstellung für das MC Protokoll vornehmen



Funktionen

Open

Öffnet den seriellen Com-Port für die Verbindung anhand der angegebene Parameter.

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	Nummer des Com-Ports Der Gerätenamen wird normalerweise mit „sprintf (ComDev, “/dev/ttyS%d”, Com),“ generiert. Wenn Sie für diesen COM-Port einen anderen Device-Namen verwenden möchten, können Sie zuvor 'MQR2SetDevName' aufrufen, um einen anderen Device-Namen für den ausgewählten Port zu registrieren
PLCType	0 = normal Q-PLC, 1 = A-PLC
Timeout	Timeout-Wert in ms, um auf die Antwort der SPS zu warten, Standard 3000 ms
StationNr	Station Nummer der SPS (Standard 0, siehe Konfiguration der SPS)
NetworkNr	Network Nummer der SPS (Standard 0, siehe Konfiguration der SPS)
PCNr	normally 0xFF, (noch nicht mit anderen Werten getestet)
Baudrate	Die Baudrate, folgende Werte sind möglich: 1200, 1800, 2400, 4800, 9600,19200, 38400,57600, 115200 je nach SPS
Parity	kann 'N' für None, 'O' für odd und 'E' für even
StopBits	Anzahl der Stopbits (1 or 2)

Result

Der Rückgabewert ist die Referenz für den Aufruf aller weiteren Funktionen.

Wenn der Wert negativ ist (kleiner als 0), überprüfen Sie die obigen Fehlercodes.

Wenn der Wert größer oder gleich Null ist, ist alles in Ordnung. Sie können bis zu 16 Kanäle öffnen.

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	Networknumber 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

Der Ergebniswert ist die Referenz für den Aufruf aller weiteren Funktionen.
 Wenn der Wert negativ ist (kleiner als 0), überprüfen Sie die obigen Fehlercodes.
 Wenn der Wert größer oder gleich 0 ist, ist alles in Ordnung. Sie können bis zu 256 Kanäle öffnen.

Close

Schließt die COM-Port Verbindung der übergebenen Referenz.

MQR2Close

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

MQJClose

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

Result

Überprüfen Sie für das Ergebnis die oben beschriebenen Fehlercodes.

Read / Write

Word

Lesen und Schreiben von Wörtern (word) von der SPS

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	Bedeutung
Ref	die durch die „Open-Funktion“ erzeugte Referenz
Type	Datentyp, siehe Liste oben 'M', 'X' usw., siehe Datenbereiche

Name	Bedeutung
Start	Nummer des ersten Elements, dass gelesen werden soll
Cnt	die Anzahl der zu lesenden Elemente. Beachten Sie, dass Ihr <Buffer> groß genug sein muss!
Buffer	Zeiger auf den Speicher im PC. Normalerweise ist dies eine Array von Elementen

Result

Überprüfen Sie für das Ergebnis die oben beschriebenen Fehlercodes.

DWORD

Lesen oder schreiben von Doppelworten (double word) von der SPS

Für die Parameterbeschreibung siehe „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

Lesen oder schreiben von double von der SPS

Für die Parameterbeschreibung siehe „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

lesen oder schreiben von bit (boolean) von der SPS

Für die Parameterbeschreibung siehe „MQR2RdW“.

Jedes Byte repräsentiert ein Bit. 1 = Bit ist gesetzt. 0 = Bit ist nicht gesetzt. z.B.

```

BYTE Buffer[10];

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

// Read
now: Buffer[0] = 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);

```

GetLastErrorCode

Diese Funktion kann aufgerufen werden, um den letzten Fehlercode im Protokoll zu ermitteln.

Siehe obige Fehlerliste!

Rufen Sie es auf, wenn Sie das Ergebnis MQJ_E_CHECK_ERRCODE oder MQR2_E_UNKNOWN_ERRCODE erhalten

MQR2GetLastErrorCode

```

long WINAPI
MQR2GetLastErrorCode (long Ref);

```

MQJGetLastErrorCode

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

SetDevName

Der Gerätenamen wird normalerweise mit „sprintf (ComDev, “/dev/ttyS%d“, Com)“, erstellt.

MQR2SetDevName

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

Wenn Sie für diesen COM-Port einen anderen Device-Namen verwenden möchten, können Sie zuvor 'MQR2SetDevName' aufrufen, der für den ausgewählten Port registriert wird.

Z.B. um für den COM1-Port den Name “/dev/tty/mytty“, festzulegen, rufen Sie folgendes auf:

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

Jetzt benutzen alle Open-Calls mit COM1 diesen Device-Namen.

Result

Das Ergebnis -1 bedeutet, dass Sie eine COM-Nummer größer als 15 ausgewählt haben!

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 Fehlercodes / Komplette Codes

Dieser Abschnitt erklärt die Endcodes (vollständige Codes), die den Antworten hinzugefügt werden.

Endcode	Beschreibung	Behandlung
00 _H	Normaler Abschluss	—
02 _H	Die Bezeichnung des Gerätebereichs der zu lesenden / geschriebenen Geräte von / bis ist falsch	Prüfen und korrigieren Sie den vorgesehenen Kopf und die Anzahl der Punkte
50 _H	Codes für Befehl / Antwort-Typ von Subheader sind nicht innerhalb der Spezifikationen In Kommunikation mit dem festen Puffer, wenn die Datenlänge Einstellung kleiner als die tatsächliche Datenzählung ist, werden die verbleibenden Daten als die zweiten Daten bestimmt und verarbeitet. In diesem Fall kann ein Untertitel undefinierter Befehlstypfehler auftreten.	Überprüfen und korrigieren Sie den Befehl / Antworttyp, der von einem externen Gerät festgelegt wurde. (Das Ethernet-Modul fügt automatisch den Befehl / Antwort-Typ hinzu, der Benutzer braucht diese nicht einzustellen.) Prüfen und korrigieren Sie die Datenlänge
51 _H	Bei der Kommunikation unter Verwendung des Direktzugriffspuffers wird die von einer externen Vorrichtung bezeichnete Kopfadresse außerhalb des Bereichs von 0 bis 6143 gesetzt	Prüfen und korrigieren Sie die angegebene Kopfadresse
54 _H	Bei Auswahl von „ASCII-Code-Kommunikation“ in [Betriebseinstellungen] - [Kommunikationsdatencode] mit GX Developer wurden ASCII-Code-Daten, die nicht in Binärcode konvertiert werden können, von einem externen Gerät empfangen	Prüfen und korrigieren Sie die Sendedaten des externen Geräts
55 _H	Wenn [Betriebseinstellung] - [Enable Write at RUN time] auf GX Developer deaktiviert ist (kein Häkchen gesetzt), fordert ein externes Gerät Daten auf, wenn die SPS-CPU im RUN-Zustand ist. Ein externes Gerät fordert das Schreiben eines Parameters, eines Sequenzprogramms oder eines Mikrocomputerprogramms an, während sich die SPS-CPU im RUN-Zustand befindet. (Nicht im Zusammenhang mit den Einstellungen unter [Betriebseinstellungen] - [Enable Write bei RUN mit GX Developer])	Schreiben Sie die Daten, indem Sie [Enable Write at RUN time] aktivieren (mit Häkchen). Das Schreiben eines Parameters, eines Sequenzprogramms oder eines Mikrocomputerprogramms ist jedoch nicht zulässig, während sich die CPU im RUN-Zustand befindet. Schreiben Sie diese Daten, indem Sie die SPS-CPU in den Zustand STOP stellen
56 _H	Gerätebezeichnung von einer externen Seite ist falsch	Korrigieren Sie das angegebene Gerät
	Die Byte-Länge eines Befehls entspricht nicht den Spezifikationen Beim Schreiben von Daten unterscheidet sich die eingestellte Anzahl der geschriebenen Datenpunkte von dem Wert der bezeichneten Nummer	Überprüfen Sie die Datenlänge des Befehls und wiederholen Sie die Dateneinstellung

Endcode	Beschreibung	Behandlung
57 _H	Die Anzahl der Punkte für einen von einem externen Gerät vorgegebenen Befehl überschreitet die maximale Anzahl der Verarbeitungspunkte (Anzahl der pro Kommunikation auszuführenden Verarbeitungen) für jede Verarbeitung Adressen aus der Kopfadresse (Kopfgerätenummer und Kopfschrittnummer) bis Überschreiten die bezeichneten Punkte die Maximaladressen (Gerätenummer und Schrittnummer)	Korrigieren Sie die angegebenen Punkte der Kopfadresse (Gerätenummer und Schrittnummer)
	Eine Überwachung wurde angefordert, obwohl die Überwachungsdaten nicht registriert sind	Registrieren Sie die Monitordaten
	Beim Lesen / Schreiben in einem Mikrocomputerprogramm wird eine Adresse nach dem Ende des Parametrierungsbereichs angegeben	Adressen nach der letzten Adresse können nicht gelesen werden. Korrigieren Sie die angegebene Adresse
	In der Blocknummernbezeichnung des Erweiterungsdateiregisters wird eine Blocknummer bezeichnet, die den Bereich der entsprechenden Speicherkassettengröße überschreitet	Korrigieren Sie die Blocknummer
58 _H	Eine Kopfadresse (Kopfgerätenummer und Kopfschrittzahl) eines von einer externen Vorrichtung bezeichneten Befehls wird außerhalb des Bereichs gesetzt, der bezeichnet werden kann. Beim Lesen von / Schreiben in ein Mikrocomputerprogramm oder Dateiregister (R) wurden Werte überschritten, die den Parametereinstellbereich der SPS-CPU überschritten	Geben Sie die entsprechenden Werte innerhalb des für jede Verarbeitung zulässigen Bereichs an
	Eine für ein Erweiterungsdateiregister vorgesehene Bausteinnummer existiert nicht	Korrigieren Sie die Blocknummer
	Kann ein Datei-Register (R) nicht bestimmen	Überprüfen Sie das Gerät
	Ein Wortgerät ist für einen Befehl für Bit-Geräte ausgelegt. Die Kopfzahl der Bit-Geräte wird in einem Befehl für Wortgeräte mit einem anderen Wert als einem Vielfachen von 16 bezeichnet	Korrigieren Sie den Befehl oder das angegebene Gerät
59 _H	Kann ein Erweiterungsdatei-Register nicht bestimmen	Überprüfen Sie das Gerät
5B _H	Die SPS-CPU und das Ethernet-Modul können nicht kommunizieren. Die SPS-CPU kann keine Requests von einem externen Gerät verarbeiten	Beheben Sie die fehlerhaften Teile, indem Sie auf die an den Endcodes angehängten abnormalen Codes verweisen
60 _H	Die Kommunikationszeit zwischen dem Ethernet-Modul und der SPS-CPU überstieg den Timer-Wert der CPU-Überwachung	Erhöhen Sie den Zeitgeberwert der CPU-Überwachung

A-SPS Fehlercodes / Komplette Codes

Fehlercode	Fehler	Fehlerbeschreibung	Korrekturmaßnahme
10 _H	PC-Nummer Fehler (SPS-Nummer Fehler)	Eine Station mit der angegebenen PC-Nummer existiert nicht. (1) Die mit einem Befehl bezeichnete PC-Nummer ist weder „FF“ der lokalen Station noch eine der mit den MELSECNET-Verbindungsparametern bezeichneten Stationsnummern	(1) Ändern Sie die PC-Nummer auf „FF“ der lokalen Station oder eine im Link-Parameter eingestellte Stationsnummer und kommunizieren Sie erneut

Fehlercode	Fehler	Fehlerbeschreibung	Korrekturmaßnahme
11 _H	Modusfehler	Schlechte Kommunikation zwischen dem Ethernet-Modul und der SPS-CPU (1) Nachdem das Ethernet-Modul eine Anfrage von einem externen Gerät erfolgreich empfangen hat, konnten das Ethernet-Modul und die SPS-CPU aus irgendeinem Grund nicht kommunizieren (Rauschen usw.)	(1) Kommunizieren Sie wieder. Tritt ein Fehler erneut auf, überprüfen Sie Rauschen usw. und tauschen Sie das Ethernet-Modul aus, und kommunizieren Sie dann erneut
12 _H	Intelligenter Funktionsbausteinbezeichnungsfehler	Intelligenter Funktionsbausteinfehler (1) Der intelligente Funktionsbaustein mit dem Pufferspeicher, der mit kommuniziert werden kann, existiert nicht an der von der intelligenten Funktionsbausteinnummer angegebenen Stelle. (Beispielsweise befindet sich die entsprechende Stelle für ein Ein- / Ausgabe-Modul oder einen leeren Steckplatz.)	(1) Ändern Sie den im Steuerungsverfahren festgelegten Dateninhalt oder ändern Sie den Installationsort des intelligenten Funktionsbausteins und kommunizieren Sie erneut
18 _H	Remote Fehler	Remote RUN / STOP nicht erreichbar. Ein anderes Modul (anderes Ethernet-Modul, etc.) hat bereits remote STOP / PAUSE ausgeführt	(1) Überprüfen Sie, ob ein anderes Modul Remote STOP / PAUSE ausgeführt hat. Wenn einer dieser Befehle ausgeführt wurde, können Sie ihn abrechnen und erneut kommunizieren
1F _H	Geräte Fehler	Ungültige Gerätespezifikation	(1) Überprüfen Sie das angegebene Gerät. (2) Zugriff auf nicht vorhandene Geräte
20 _H	Link Fehler	Das CPU-Modul des Anforderungsziels wird von der Datenverbindung getrennt	Prüfen Sie, ob die SPS-CPU mit der als PC-Nummer eingestellten Stationsnummer getrennt ist oder nicht. Die Ursache der Abschaltung entfernen und wieder anschließen
21 _H	Fehlerhafter Funktionsbausteinbus	Zugriff auf den Speicher des intelligenten Funktionsbausteins nicht möglich. (1) Der Steuerbus zum intelligenten Funktionsbaustein ist defekt. (2) Der intelligente Funktionsbaustein ist defekt	Eine der folgenden Hardware ist defekt: die SPS-CPU, Basiseinheit, intelligenter Funktionsbaustein oder Ethernet-Modul. Bitte wenden Sie sich an Ihren Fachhändler

Inhaltsverzeichnis

- Betriebssystem** 2
- Programmiersprachen** 2
- Funktionsweise** 2
- Datenbereiche** 2
 - Return Values 3
- MQR2 - error values 3
- MQJ - error values 3
- Voraussetzungen** 4
 - Build-In Ethernet 4
 - Ethernet-Module QJ71E71 4
- Funktionen** 5
 - Open 5
- MQR2Open 5
- Parameter 5
- Result 5
- MQJOpen 5
- Result 6
 - Close 6
- MQR2Close 6
- MQJClose 6
- Result 6
 - Read / Write 6
- Word 6
- MQR2RdW / MQR2WrW 6
- MQJRdW / MQJWrW 6
- Result 7
- DWORD 7
- MQR2RdDW / MQR2WrDW 7
- MQJRdDW / MQJWrDW 7
- REAL 7
- MQR2RdReal / MQR2WrReal 7
- MQJRdReal / MQJWrReal 7
- BIT 8
- MQR2RdBit / MQR2WrBit 8
- MQJRdBit / MQJWrBit 8
 - GetLastErrCode 8
- MQR2GetLastErrCode 8
- MQJGetLastErrCode 8
 - SetDevName 9
- MQR2SetDevName 9
- Result 9
 - MQJGetSockErr 9
- Q-PLC Fehlercodes / Komplette Codes** 10
- A-SPS Fehlercodes / Komplette Codes** 11

