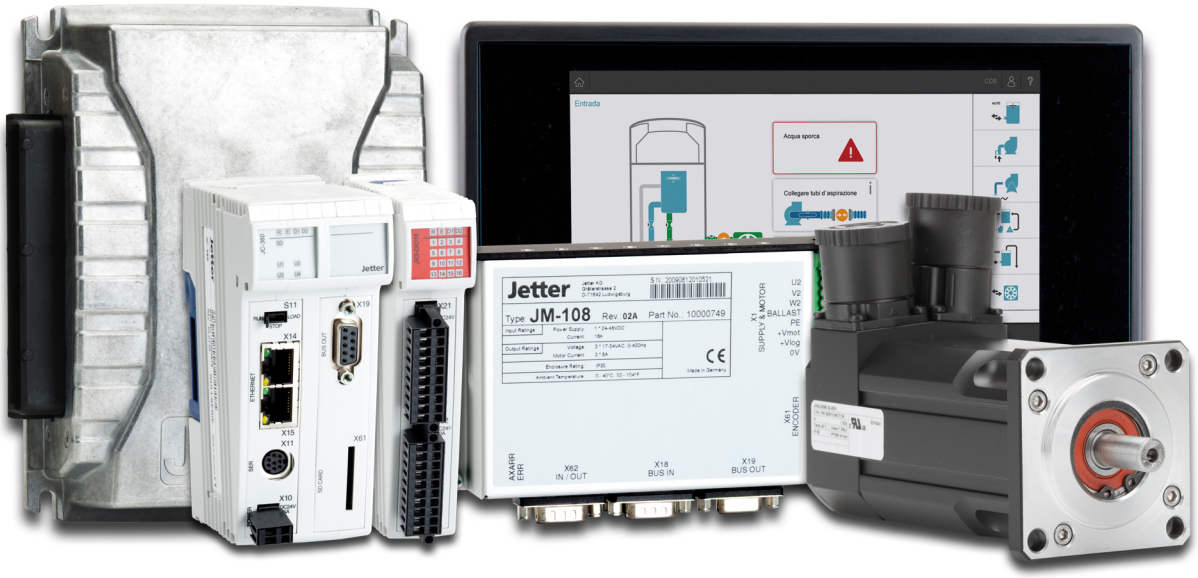
Jetter automation



JC-340

Versionsupdate von V. 1.24 auf V. 1.28

We automate your success.

Version 1.01
Juli 2017 / Printed in Germany

Dieses Dokument hat die Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt.

Bei Änderungen, Weiterentwicklungen oder Erweiterungen bereits zur Verfügung gestellter Produkte wird ein überarbeitetes Dokument nur beigefügt, sofern dies gesetzlich vorgeschrieben oder von der Jetter AG für sinnvoll erachtet wird. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Die im Dokument aufgeführten Logos, Bezeichnungen und Produktnamen sind geschützte Marken der Jetter AG, der mit ihr verbundenen Unternehmen oder anderer Inhaber und dürfen nicht ohne Einwilligung des jeweiligen Inhabers verwendet werden.

Inhaltsverzeichnis

1 Einleitung 4

Betriebssystemupdate 5
JC-340 Versionsupdate Übersicht 6

2 Erweiterungen 13

2.1 Diverse Funktionserweiterungen und Änderungen 14
Melden der IP-Adresse 15
Neue Funktion FileEnd() 16
Neue Funktionen DirLister 17
Neue Funktionen BitSetReg() und BitClearReg() 18
STX-Speichernutzung 19
SD-Karte mit FAT-32 formatieren 20
Taskwechsel bei I/O-Zugriffen aktiv 21
Mehr Verbindungen beim JetIP/TCP-Server 22
Erweitertes Fehlerregister 200009 23
NetConsistency kopiert die Konfigurationsdateien 24
2.2 Freiprogrammierbare IP-Schnittstelle 25
2.2.1 Programmierung 27
Initialisieren der freiprogrammierbaren IP-Schnittstelle 28
Verbindung öffnen 29
Daten senden 33
Daten empfangen 35
Verbindung schließen 38
2.2.2 Register 39
Registernummern 40
Registerbeschreibung 41

3 Beseitigte Software-Bugs 44

Ungültiges Verbindungs-Handle bei der freiprogrammierbaren IP-Schnittstelle 45
Lange Schlüsselnamen führen zum Absturz 46
Umbenennen einer Datei mit maximaler Namenslänge führt zum Absturz 47
Eingangswerte eines Netzwerkteilnehmers sind eingefroren 48
Fehlerhafte Anzeige von Float-Registern im JetSym-Oszilloskop 49
Inkonsistente Register bei der IP-Konfiguration 50

1 Einleitung

Einleitung

Dieses Kapitel zeigt eine Historie der Betriebssystemversionen.

Betriebssystemupdate - wozu?

Ein Betriebssystemupdate bietet Ihnen die folgenden Möglichkeiten:

- Die Funktionalität erweitern
- Software-Bugs beheben
- Einen bestimmten Betriebssystemstand übertragen, z. B. nach Freigabe einer Betriebssystemversion

Inhalt

Thema	Seite
Betriebssystemupdate	5
JC-340 Versionsupdate Übersicht	6

Betriebssystemupdate

OS-Datei zum Betriebssystemupdate

Zum Betriebssystemupdate benötigen Sie die folgende Datei:

OS-Datei	Beschreibung
JC-340_1.28.0.00.os	Betriebssystemdatei für JC-340 mit der Version 1.28

Download eines Betriebssystems

Die Jetter AG stellt die Betriebssysteme auf ihrer **Homepage https://www.jetter.de/downloads.html** zur Verfügung. Sie finden die Betriebssystemdateien zum Download beim jeweiligen Produkt.

Betriebssystemupdate mit JetSym

Führen Sie zum Betriebssystemupdate folgende Schritte aus.

Schritt	Vorgehen
1	Download der OS-Datei von www.jetter.de.
2	Herstellung einer Verbindung zwischen PC und Steuerung.
3	In JetSym: Ausführen des Menüpunkts Build -> Betriebssystemupdate oder Klicken auf die Schaltfläche OS-Update im CPU-Fenster des Hardware-Managers.
4	Auswahl der OS-Datei.
5	Start des Betriebssystemupdates mit OK.
6	**Ergebnis:** Nach Power Off/Power On startet das neue Betriebssystem.

Mindestvoraussetzungen

Für die Programmierung des JC-340 ab Version 1.28 ist JetSym ab Version 5.3.0 erforderlich.

JC-340 Versionsupdate Übersicht

V 1.04

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.04:

Funktion	Neu	Bug
JX2-Systembus:		
Registerüberlagerung der digitalen Ein-/Ausgänge	✓	
Unterstützung von JX-SIO und CANopen®-Geräten weiterer Hersteller	✓	
JX3-Systembus:		
Registerüberlagerung der digitalen Ein-/Ausgänge	✓	
Systembus Spezialregister für Status und Steuerung	✓	
Betriebssystemupdate:		
Über FTP: Bei Fertigmeldung ist das OS wirklich gespeichert		✓
Update auf JX2-Slave bei gleichzeitigem Registerzugriff blockiert die Kommunikation		✓
Anwendungsprogramm:		
Taskwechsel konnte ausbleiben		✓
Fehleranzeige, wenn Datei **\App\start.ini** nicht korrekt		✓
Display-Befehle:		
Nur wenn JX2-PRN1 auch konfiguriert ist, funktioniert die Umleitung auf JX2-SER1		✓

V 1.05

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.05:

Funktion	Neu	Bug
JX2-Systembus: V1.05.0.00		
AS-Interface Gateway BWU1821 wird unterstützt	✓	
Frequenzumrichter 8200 vector wird unterstützt	✓	
JetMove 1xx wird beim Booten nicht gefunden		✓
Automatische Baudratenerkennung funktioniert nicht immer bei unterschiedlichen Baudraten und Konfigurationen der IP67-Module		✓
Wiederholungszähler beim Abfragen der I/O-Module funktioniert nicht		✓
AutoCopy-Funktion:		
Automatisches Kopieren von Steuerungsdaten	✓	
Anwendungsprogramm:		
Anstehende zyklische Tasks werden nach dem Taskunlock sofort gestartet	✓	

Funktion	Neu	Bug
Bei der Funktion pow(x,y) ist als Exponent eine Fließkommazahl möglich	✓	
Debuggen von zyklischen Tasks möglich	✓	
Länge von Projekt- und Programmname > 39 Zeichen		✓
Ein abgelaufener Timer läuft wieder an		✓
DateTimeDecode() lieferte den Tag um 1 zu klein		✓
DateTimeEncode und -IsValid können trotz ungültigem Datum den Wert TRUE zurückliefern		✓
Anwendungsregister:		
Einrichten des Registertyps ohne Starten des Anwendungsprogramms	✓	
Anzeige- und Bediengeräte:		
Vorgabewert beim UserInput als Fließkommawert möglich	✓	
Vorgabewert beim UserInput wird nicht korrekt angezeigt		✓
LED-Registernummern können nicht eingetragen werden		✓

V 1.08

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.08:

Funktion	Neu	Bug
Systemkonfiguration:		
Systemrechte für Konfigurationsdatei	✓	
JX2-Systembus: V1.11.0.00		
Timeout nach CAN-PRIM-Message		✓
Register von LJX7-CSL-Modulen		✓
Beschreiben der Analogausgänge von CANopen®-Modulen		✓
Zustand der digitalen Eingänge beim Einschalten		✓
Digitale Ausgänge bei JX-SIO oder CANopen®-Geräten		✓
Ein-/Ausgang 64 bei JX-SIO oder CANopen®-Geräten		✓
Freiprogrammierbare CAN-Schnittstelle		✓
Anwendungsprogramm:		
NetCopyList-Funktionen	✓	
StrCopy()		✓
Absturz bei "ungültigem" Anwendungsprogramm		✓
NetCopyVarFromReg()		✓
JX3-Systembus:		
Modulregister für digitale Ein-/Ausgänge	✓	

Funktion	Neu	Bug
Anzeige- und Bediengeräte:		
UserInput()		✓

V 1.09

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.09:

Funktion	Neu	Bug
System:		
Systemkommandoregister	✓	
JX2-Systembus: V1.13.0.00		
Eingangsänderungen am JX2-ID8		✓
Schnelle Eingangsänderungen		✓
Anwendungsprogramm:		
FTP-Client	✓	
Achsbefehle		✓
Taskrestart bei Delay()		✓
Absturz bei fehlender Bibliothek		✓
Fließkommazahlregister in Datendateien		✓
NetCopyVarToReg mit Fließkommazahlregistern		✓
JX3-Systembus:		
Dummy-Module	✓	
AutoCopy:		
FTP-Kommandos	✓	
Serielle Schnittstelle:		
Initialisierung nach dem Booten		✓

V 1.10

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.10:

Funktion	Neu	Bug
System:		
LED-Register		✓
SD-Karte		✓
JX2-Systembus: V1.17.0.00		
Weitere Module	✓	
CAN-PRIM	✓	

Funktion	Neu	Bug
Anwendungsprogramm:		
Taskbefehle mit Variablenparameter	✓	
UserInput()		✓
NetCopyListSend()		✓
Taskstatusregister		✓
Echtzeituhr:		
Zusätzliches Register für Millisekunden	✓	
Freiprogrammierbare IP-Schnittstelle:		
Mehr Verbindungen	✓	

V 1.12

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.12:

Funktion	Neu	Bug
System:		
Systemkommandoregister	✓	
JX2-Systembus: V1.21.0.00		
Initialisierung		✓
CAN-PRIM		✓
CANopen®-Sync-Intervall		✓
CANopen®-Anwendungsregister		✓
CANopen®-Typ "String"		✓
CANopen®-Ausgang setzen		✓
CANopen®-Versionsnummer		✓
Wago 750		✓
JX3-Systembus:		
Registerzugriffe		✓
Anwendungsprogramm:		
Programmsteuerung	✓	✓
Zuweisungen von Strukturen	✓	
Daten sortieren	✓	
Variablenanzeige in JetSym		✓
HTTP-Server:		
Neuer Dateityp	✓	
Serielle Schnittstelle:		
Fehlererkennung		✓

V 1.14

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.14:

Funktion	Neu	Bug
JX2-Systembus: V1.22.0.00		
Betriebssystemupdate		✓
Anwendungsprogramm:		
Neue Befehle	✓	

V 1.16

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.16:

Funktion	Neu	Bug
JX2-Systembus: V1.23.0.00		
CANopen®-Register		✓
Anwendungsprogramm:		
Neue Datentypen	✓	
Neue Funktionen	✓	
Speicherschutz	✓	
Zyklische Tasks		✓
NetCopyVarToReg		✓
Zykluszeitregister		✓

V 1.18

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.18:

Funktion	Neu	Bug
Anwendungsprogramm:		
Debugging	✓	
Speicherschutz	✓	✓
Teildownload		✓
StrFormat()		✓
Exceptions		✓
Ethernet-Systembus:		
Erweiterte Diagnose	✓	
Modulunterstützung		✓
JX3-Systembus:		
Register	✓	

Funktion	Neu	Bug
Initialisierung		✓
OS-Update		✓
STX-Debug-Server:		
TCP-Verbindungsverwaltung	✓	

V 1.22

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.22:

Funktion	Neu	Bug
Anwendungsprogramm:		
Debugging	✓	
Neue Funktionen	✓	
Tasklock	✓	
Initialisierung von Variablen	✓	
Teildownload		✓
Exceptions		✓
Ethernet-Systembus:		
IP-Adresseinstellung	✓	
JetSync-Blocker	✓	
JX2-Systembus:		
Fehlerzähler/-bits	✓	
CAN-PRIM	✓	
Fehleranzeige		✓
Register für Ausgangsüberlagerung		✓
DNS-Client:		
IP-Adresse des DNS-Servers	✓	
Diagnose	✓	

V 1.24

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.24:

Funktion	Neu	Bug
Anwendungsprogramm:		
Taskbearbeitung (Systemkommando 170/171)	✓	
Taskcontinue	✓	
Ethernet-Systembus:		
NetConsistency	✓	
System:		
Startverzögerung	✓	
IP-Konfiguration	✓	

V 1.28

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystemversion 1.28:

Funktion	Neu	Bug
Anwendungsprogramm:		
Speicherverwaltung	✓	
Neue Funktionen	✓	
Kommunikation:		
STX-Debug-Server	✓	✓
JetIP-Server	✓	
ARP	✓	
NetConsistency	✓	
Dateisystem:		
SD-Karte	✓	
Rename		✓
Freiprogrammierbare IP-Schnittstelle:		
Puffer-Verwaltung	✓	
Senden/Empfangen	✓	✓
System:		
Fehleranzeige	✓	
I/O-Zugriffe	✓	

2 Erweiterungen

Einleitung

Die Entwicklung der Jetter AG erweitert ständig die Funktionen der Steuerung JC-340. Durch ein Betriebssystemupdate lässt sich die Funktionalität der Steuerung um die neuen Funktionen erweitern. Sie benötigen dazu ...

- eine OS-Datei
- das Softwaretool JetSym
- eine Verbindung zwischen dem PC und der Steuerung

Inhalt

Thema	Seite
Diverse Funktionserweiterungen und Änderungen	14
Freiprogrammierbare IP-Schnittstelle	25

2.1 Diverse Funktionserweiterungen und Änderungen

Einleitung

Dieses Kapitel beschreibt diverse Funktionserweiterungen und Änderungen.

Inhalt

Thema	Seite
Melden der IP-Adresse	15
Neue Funktion FileEnd()	16
Neue Funktionen DirLister	17
Neue Funktionen BitSetReg() und BitClearReg()	18
STX-Speichernutzung	19
SD-Karte mit FAT-32 formatieren	20
Taskwechsel bei I/O-Zugriffen aktiv	21
Mehr Verbindungen beim JetIP/TCP-Server	22
Erweitertes Fehlerregister 200009	23
NetConsistency kopiert die Konfigurationsdateien	24

Melden der IP-Adresse

Funktion

Die Steuerung JC-340 meldet ihre aktuelle IP-Adresse den anderen Netzwerkteilnehmern. Dazu versendet sie ein spezielles Ethernet-Telegramm ("Gratuitous ARP"), jeweils einmalig, bei folgenden Gelegenheiten:

- während der Boot-Phase, nachdem die IP-Einstellungen aktiv sind.
- während der Laufzeit, wenn sich die IP-Einstellungen geändert haben.

Vorteile

Es ergeben sich folgende Vorteile:

- Die anderen Netzwerk-Teilnehmer und die Infrastruktur-Geräte (z. B. Switche) können sich sofort auf den "neuen" Teilnehmer einstellen.
- Die eigentliche Kommunikation mit diesem Teilnehmer kann i.d.R. schneller aufgenommen werden.

Neue Funktion FileEnd()

Einleitung	Ab dieser Betriebssystemversion der Steuerung JC-340 wird die STX-Funktion `FileEnd()` unterstützt.
Voraussetzungen	Für die Verwendung dieser Funktionen ist die Programmierumgebung JetSym in der Version 5.3 oder höher erforderlich.
Deklaration	`Function FileEnd(Ref F:File):Int`
Verweis	Eine detaillierte Beschreibung des Befehls und seiner Anwendung finden Sie in der Online-Hilfe von JetSym.

Neue Funktionen DirLister

Einleitung	Ab dieser Betriebssystemversion der Steuerung JC-340 werden die STX-Funktionen des DirListers unterstützt.
Voraussetzungen	Für die Verwendung dieser Funktionen ist die Programmierumgebung JetSym in der Version 5.3 oder höher erforderlich.

Deklaration	`Type`
	`    DirKind : Enum (Files = 0, Folders = 1);`
	`End_Type`
	
	`Function DirListPath(enKind: DirKind, Const Ref strPath : String, Const Ref strFilter : String := '*.*') : Bool;`
	`Function DirListGetEntry(Ref strEntry : String) : Bool;`
	`Function DirListClose();`

Verweis	Eine detaillierte Beschreibung der Befehle und ihrer Anwendung finden Sie in der Online-Hilfe von JetSym.

Neue Funktionen BitSetReg() und BitClearReg()

Einleitung	Ab dieser Betriebssystemversion der Steuerung JC-340 werden die STX-Funktionen `BitSetReg()` und `BitClearReg()` unterstützt.
Voraussetzungen	Für die Verwendung dieser Funktionen ist die Programmierumgebung JetSym in der Version 5.3 oder höher erforderlich.
Deklaration	`Function BitSetReg(RegNr:Int, BitNr:Int);` `Function BitClearReg(RegNr:Int, BitNr:Int);`
Verweis	Eine detaillierte Beschreibung der Befehle und ihrer Anwendung finden Sie in der Online-Hilfe von JetSym.

STX-Speichernutzung

Einleitung

Ab dieser Betriebssystem-Version der Steuerung JC-340 wird die Anzeige der Speichernutzung von JetSym STX unterstützt.

Voraussetzungen

Für die Anzeige der Speichernutzung ist die Programmierumgebung JetSym in der Version 5.3.1 oder höher erforderlich.

Registerübersicht

In diesen Registern ist die Speichernutzung des Anwenderprogramms in Bytes lesbar. Im CPU-Fenster des Hardware-Managers von JetSym finden Sie eine grafische Darstellung der Werte.

Register	Beschreibung
R 211010	Gesamter Speicher: Insgesamt
R 211011	Gesamter Speicher: Verwendet
R 211012	Gesamter Speicher: Frei
R 211013	Systemspeicher: Insgesamt
R 211014	Systemspeicher: Verwendet
R 211015	Systemspeicher: Frei
R 211016	Anwenderspeicher: Insgesamt
R 211017	Anwenderspeicher: Verwendet
R 211018	Anwenderspeicher: Frei
R 211019	Verwendeter Speicher: Programm
R 211020	Verwendeter Speicher: Daten
R 211021	Verwendeter Speicher: Konstanten
R 211022	Verwendeter Speicher: Stack
R 211023	Verwendeter Speicher: JIT-Compiler
R 211024	Verwendeter Speicher: System

Verweis

Eine detaillierte Beschreibung der Anzeige der STX-Speichernutzung finden Sie in der Online-Hilfe von JetSym.

SD-Karte mit FAT-32 formatieren

Bisherige Funktion

Die Steuerung JC-340 unterstützt folgende Funktionen mit einer SD-Karte:

- Lesen und schreiben auf einer SD-Karte im Format FAT-16.
- Lesen und schreiben auf einer SD-Karte im Format FAT-32.
- Formatieren einer SD-Karte im Format FAT-16.

Neue Funktion

Die Steuerung JC-340 unterstützt zusätzlich folgende Funktion mit einer SD-Karte:

- Formatieren einer SD-Karte im Format FAT-32.

Formatieren

Gehen Sie zum Formatieren einer SD-Karte in der Steuerung wie folgt vor:

Schritt	Vorgehen
1	Beschreiben Sie Register 202936 mit dem Wert 1179923250 (0x46543332).
2	Schalten Sie die Stromversorgung der Steuerung aus.
3	Schalten Sie die Stromversorgung der Steuerung wieder ein. **Ergebnis**: Während des Boot-Vorgangs wird die SD-Karte mit FAT-32 formatiert.

Taskwechsel bei I/O-Zugriffen aktiv

Bisherige Funktion	Nach dem Einschalten der Steuerung JC-340 ist die Funktion Taskwechsel bei I/O-Zugriffen deaktiviert.
Neue Funktion	Nach dem Einschalten der Steuerung JC-340 ist die Funktion Taskwechsel bei I/O-Zugriffen aktiviert.
Einstellen der Funktion	Die Funktion Taskwechsel bei I/O-Zugriffen stellen Sie über das Systemkommandoregister 202961 ein. ▪ Wenn Sie die Funktion Taskwechsel bei I/O-Zugriffen deaktivieren wollen verwenden Sie das Systemkommando 160. ▪ Wenn Sie die Funktion Taskwechsel bei I/O-Zugriffen aktivieren wollen verwenden Sie das Systemkommando 161. Der Zustand der Funktion wird in Bit 0 vom Systemstatusregister 202962 angezeigt. ▪ Wenn Bit 0 gesetzt (= 1) ist, erfolgt ein Taskwechsel bei einem I/O-Zugriff. ▪ Wenn Bit 0 nicht gesetzt (= 0) ist, erfolgt kein Taskwechsel bei einem I/O-Zugriff.

Mehr Verbindungen beim JetIP/TCP-Server

Bisherige technische Daten

Parameter	Beschreibung
Anzahl Verbindungen	4

Neue technische Daten

Parameter	Beschreibung
Anzahl Verbindungen	8

Grund der Änderung

Es können mehr Verbindungen gleichzeitig geöffnet sein.

Erweitertes Fehlerregister 200009

R 200009

Die in dieser Version neu hinzugekommenen Fehlerbits sind grau unterlegt:

Bedeutung der Bits	
Bit 3	Fehler in Datei Modconfig.da
Bit 5	Interner fataler Fehler der Ausführungseinheit des Anwenderprogramms
Bit 10	Fehlermeldung von Teilnehmer am Jetter-Ethernet-Systembus
Bit 12	Fehlermeldung von JetIPScan
Bit 16	Fehlermeldung von NetConsistency
Bit 20	Interner Fehler der Speicherverwaltung des Betriebssystems
Bit 21	Interner Fehler der Speicherverwaltung des Anwenderprogramms
Bit 22	Zum Boot-Zeitpunkt ist der System-Logger aktiv (Register 209700 = 213)
Bit 24	IP-Adresskonflikt erkannt

Modulregistereigenschaften	
Zugriff	Lesen
Wert nach Reset	0

NetConsistency kopiert die Konfigurationsdateien

Bisherige Funktion	Die Funktion NetConsistency prüft die IP-Einstellungen der konfigurierten Netzwerkteilnehmer und stellt sie gegebenenfalls ein.
Neue Funktion	Die Funktion NetConsistency kopiert zusätzlich die Konfigurations- und Parameterdateien der konfigurierten Netzwerkteilnehmer und startet sie neu.
Einschränkung	Die Netzwerkteilnehmer müssen sternförmig vernetzt sein.

2.2 Freiprogrammierbare IP-Schnittstelle

Die freiprogrammierbare IP-Schnittstelle

Die freiprogrammierbare IP-Schnittstelle bietet die Möglichkeit, beliebige Daten mit TCP/IP oder UDP/IP über die Ethernet-Schnittstelle des Geräts zu versenden oder zu empfangen. Die Verarbeitung der Daten geschieht dabei ganz im Anwendungsprogramm.

Anwendungen

Die freiprogrammierbare IP-Schnittstelle erlaubt dem Programmierer Daten über Ethernet-Verbindungen auszutauschen, die sich nicht Standard-protokollen - wie z. B. FTP, HTTP, JetIP oder Modbus/TCP - bedienen. Folgende Anwendungen sind dabei möglich:

- Server
- Client
- TCP/IP
- UDP/IP

Voraussetzungen an den Programmierer

Die Funktionalität der freiprogrammierbaren IP-Schnittstelle setzt folgende Kenntnisse der Datenübertragung über IP-Netzwerke voraus:

- IP-Adressierung (z. B. IP-Adresse, Port-Nummer, Subnetzmaske)
- TCP (z. B. Verbindungsaufbau / -abbau, Datastream, Datensicherung)
- UDP (z. B. Datagram)

Einschränkungen

Für die Kommunikation über die freiprogrammierbare IP-Schnittstelle darf der Programmierer keine Ports verwenden, die schon das Betriebssystem verwendet. Verwenden Sie deshalb folgende Ports nicht:

Protokoll	Port-Nummer	Standardwert	Benutzer
TCP	Abhängig vom FTP-Client	20	FTP-Server (Daten)
TCP	21		FTP-Server (Steuerung)
TCP	23		System-Logger
TCP	80		HTTP-Server
TCP	Aus der Datei /EMAIL/email.ini	25, 110	E-Mail-Client
TCP	502		Modbus/TCP-Server
TCP, UDP	1024 - 2047		Diverse
TCP, UDP	IP-Konfiguration	50000, 50001	JetIP
TCP	IP-Konfiguration	52000	Debug-Server

Inhalt

Thema	Seite
Programmierung	27
Register	39

2.2.1 Programmierung

Einleitung

Bei der freiprogrammierbaren IP-Schnittstelle werden die Daten über TCP/IP- oder UDP/IP-Verbindungen zwischen dem Anwendungsprogramm und einem Netzwerkteilnehmer ausgetauscht. Verwendung finden hierbei Aufrufe von Funktionen, die im Sprachumfang des Geräts enthalten sind. Führen Sie zur Programmierung folgende Schritte aus:

Schritt	Vorgehen
1	Freiprogrammierbare IP-Schnittstelle initialisieren
2	Verbindungen öffnen
3	Daten übertragen
4	Verbindungen schließen

Technische Daten

Die technischen Daten der freiprogrammierbaren IP-Schnittstelle:

Funktion	Beschreibung
Anzahl Verbindungen	20
Maximale Datengröße	4.000 Byte
Anzahl Empfangspuffer pro Verbindung	4

Einschränkungen

Während das Gerät gerade eine der Funktionen der freiprogrammierbaren IP-Schnittstelle bearbeitet, dürfen die Tasks, die die Funktionen aufgerufen haben, nicht durch `TaskBreak` angehalten oder durch `TaskRestart` neu gestartet werden.

Dies kann zu folgenden Fehlern führen:

- Verbindungen öffnen sich nicht
- Datenverlust beim Senden oder Empfangen
- Verbindungen bleiben ungewollt offen
- Verbindungen werden ungewollt geschlossen

Inhalt

Thema **Seite**
Initialisieren der freiprogrammierbaren IP-Schnittstelle ... 28
Verbindung öffnen ... 29
Daten senden ... 33
Daten empfangen ... 35
Verbindung schließen ... 38

Initialisieren der freiprogrammierbaren IP-Schnittstelle

Einleitung

Die Initialisierung muss mindestens einmal bei jedem Anwendungsprogrammstart ausgeführt werden.

Funktionsdeklaration

```
Function ConnectionInitialize():Int;
```

Rückgabewert

Folgender Rückgabewert ist möglich:

Rückgabewert	
0	Immer

Verwenden der Funktion

So wird die Funktion verwendet und der Rückgabewert einer Variablen zur weiteren Auswertung zugewiesen:

```
Result := ConnectionInitialize();
```

Funktionsweise

Das Gerät arbeitet die Funktion in folgenden Stufen ab:

Stufe	Beschreibung
1	Das Gerät schließt alle geöffneten Verbindungen der freiprogrammierbaren IP-Schnittstelle.
2	Das Gerät initialisiert alle betriebssysteminternen Datenstrukturen der freiprogrammierbaren IP-Schnittstelle.

Verwandte Themen

- **Verbindung öffnen** (siehe Seite 29)
- **Verbindung schließen** (siehe Seite 38)
- **Daten senden** (siehe Seite 33)
- **Daten empfangen** (siehe Seite 35)

Verbindung öffnen

Einleitung

Bevor Daten gesendet oder empfangen werden können, muss eine Verbindung geöffnet werden. Dabei gilt Folgendes zu unterscheiden:

- Welches Transportprotokoll (TCP oder UDP) ist zu verwenden?
- Ist ein Client oder ein Server einzurichten?

Funktionsdeklaration

```
Function ConnectionCreate(ClientServerType:Int,
                          IPType:Int,
                          IPAddr:Int,
                          IPPort:Int,
                          Timeout:Int):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
`ClientServerType`	Client = 1 = CONNTYPE_CLIENT Server = 2 = CONNTYPE_SERVER	
`IPType`	UDP/IP = 1 = IPTYPE_UDP TCP/IP = 2 = IPTYPE_TCP	
`IPAddr`	Gültige IP-Adresse	Nur bei TCP/IP-Client erforderlich
`IPPort`	Gültige IP-Port-Nummer	Wird bei UDP/IP-Client ignoriert
`Timeout`	0 ... 1.073.741.824 [ms]	0 = unendlich

Rückgabewert

Bei einem positiven Rückgabewert konnte die Verbindung geöffnet werden. Bei einem negativen Rückgabewert ist ein Fehler aufgetreten und die Verbindung konnte nicht geöffnet werden.

Rückgabewert	
> 0	Ein positiver Rückgabewert muss in einer Variablen gesichert werden. Er muss als Handle beim Aufruf der Funktionen für Daten senden, Daten empfangen und Verbindung schließen mitgegeben werden.
-1	Fehler beim Verbindungsaufbau
-2	Interner Fehler
-3	Ungültiger Parameter
-8	Zeitüberschreitung

Verwenden der Funktion bei einem TCP/IP-Client

Wenn ein Client eine TCP/IP-Verbindung zu einem Server aufbauen soll, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionCreate(CONNTYPE_CLIENT,
                           IPTYPE_TCP,
                           IP#192.168.75.123,
                           46000,
                           T#10s);
```

Funktionsweise bei einem TCP/IP-Client

Der Task bleibt bei der Programmzeile stehen, bis die Verbindung aufgebaut wurde oder der angegebene Timeout abgelaufen ist. Die Funktion läuft in folgenden Stufen ab:

Stufe	Beschreibung	
1	Das Gerät versucht eine TCP/IP-Verbindung zu Port 46000 zu dem Netzwerkteilnehmer mit der IP-Adresse 192.168.75.123 aufzubauen.	
2	**Wenn ...**	**... dann ...**
	... der Netzwerkteilnehmer die Verbindung akzeptiert hat,	... wird die Funktion beendet und ein positiver Wert als Handle zum weiteren Zugriff auf die Verbindung zurückgeliefert.
	... die Verbindung nicht aufgebaut werden konnte und die Timeout-Zeit von 10 Sekunden noch nicht abgelaufen ist,	... wird mit Stufe 1 fortgefahren.
	... ein Fehler aufgetreten ist oder der Timeout abgelaufen ist,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Verwenden der Funktion bei einem TCP/IP-Server

Wenn ein Server eine TCP/IP-Verbindung mit einem Client aufbauen soll, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionCreate(CONNTYPE_SERVER,
                           IPTYPE_TCP,
                           0,
                           46000,
                           T#100s);
```

Funktionsweise bei einem TCP/IP-Server

Der Task bleibt bei der Programmzeile stehen, bis die Verbindung aufgebaut wurde oder der angegebene Timeout abgelaufen ist. Die Funktion läuft in folgenden Stufen ab:

Stufe	Beschreibung	
1	Das Gerät richtet den TCP/IP-Port 46000 zum Empfang von Verbindungsanfragen ein.	
2	**Wenn ...**	**... dann ...**
	... der Client eines Netzwerkteilnehmers eine Verbindung aufgebaut hat,	... werden keine weiteren Verbindungsanfragen auf diesen Port akzeptiert, die Funktion beendet und ein positiver Wert als Handle zum weiteren Zugriff auf die Verbindung zurückgeliefert.
	... die Verbindung nicht aufgebaut werden konnte und die Timeout-Zeit von 100 Sekunden noch nicht abgelaufen ist,	... wird auf einen Verbindungsaufbau gewartet.
	... ein Fehler aufgetreten ist oder der Timeout abgelaufen ist,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Verwenden der Funktion bei einem UDP/IP-Client

Wenn ein Client eine UDP/IP-Verbindung aufbauen soll, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionCreate(CONNTYPE_CLIENT,
                           IPTYPE_UDP,
                           0,
                           0,
                           0);
```

Funktionsweise bei einem UDP/IP-Client

UDP ist eine verbindungslose Art der Kommunikation. Deshalb öffnet das Gerät nur einen Kommunikationskanal, über den Daten an einen Netzwerkteilnehmer gesendet werden. Die Funktion läuft in folgenden Stufen ab:

Stufe	Beschreibung	
1	Das Gerät richtet einen UDP/IP-Verbindungskanal zum Senden von Daten ein.	
2	**Wenn ...**	**... dann ...**
	... kein Fehler aufgetreten ist,	... wird die Funktion beendet und ein positiver Wert als Handle zum weiteren Zugriff auf die Verbindung zurückgeliefert.
	... ein Fehler aufgetreten ist,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Verwenden der Funktion bei einem UDP/IP-Server

Wenn ein Server eine UDP/IP-Verbindung aufbauen soll, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionCreate(CONNTYPE_SERVER,
                           IPTYPE_UDP,
                           0,
                           46000,
                           0);
```

Funktionsweise bei einem UDP/IP-Server

UDP ist eine verbindungslose Art der Kommunikation. Deshalb öffnet das Gerät nur einen Kommunikationskanal über den Daten von einem Netzwerkteilnehmer empfangen werden. Die Funktion läuft in folgenden Stufen ab:

Stufe	Beschreibung	
1	Das Gerät richtet einen UDP/IP-Verbindungskanal am Port 46000 zum Empfangen von Daten ein.	
2	**Wenn ...**	**... dann ...**
	... kein Fehler aufgetreten ist,	... wird die Funktion beendet und ein positiver Wert als Handle zum weiteren Zugriff auf die Verbindung zurückgeliefert.
	... ein Fehler aufgetreten ist,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Verwandte Themen

- **Verbindung schließen** (siehe Seite 38)
- **Daten senden** (siehe Seite 33)
- **Daten empfangen** (siehe Seite 35)
- **Initialisieren der freiprogrammierbaren IP-Schnittstelle** (siehe Seite 28)

Daten senden

Einleitung

Daten können über eine zuvor geöffnete Verbindung gesendet werden.

Funktionsdeklaration

```
Function ConnectionSendData(IPConnection:Int,
                            IPAddr:Int,
                            IPPort:Int,
                            Const Ref SendData,
                            DataLen:Int):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
`IPConnection`	Handle	Rückgabewert der Funktion `ConnectionCreate()`
`IPAddr`	Gültige IP-Adresse	Nur bei UDP/IP-Client erforderlich
`IPPort`	Gültige IP-Port-Nummer	Nur bei UDP/IP-Client erforderlich
`SendData`	Adresse des zu sendenden Datenblocks	
`DataLen`	1 ... 4.000	Länge des Datenblocks in Byte

Rückgabewert

Folgende Rückgabewerte sind möglich:

Rückgabewert	
0	Daten erfolgreich gesendet
-1	Fehler beim Senden, z. B. Verbindung abgebrochen
-3	Ungültiges Handle, z. B. senden über einen UDP/IP-Server

Verwenden der Funktion bei einer TCP/IP-Verbindung

Wenn Daten über eine TCP/IP-Verbindung gesendet werden sollen, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionSendData(hConnection,
                             0,
                             0,
                             SendBuffer,
                             SendLen);
```

Funktionsweise bei einer TCP/IP-Verbindung

Bei TCP/IP werden die Daten über eine Verbindung übertragen, die vorher geöffnet wurde. Deshalb ist die Angabe von IP-Adresse und IP-Port-Nummer nicht mehr erforderlich und werden in der Funktion ignoriert.

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

- Die Daten wurden gesendet und ihr Empfang bestätigt.
- Ein Fehler ist aufgetreten.

Verwenden der Funktion bei einer UDP/IP-Verbindung

Wenn Daten bei einem Client über eine UDP/IP-Verbindung gesendet werden sollen, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionSendData(hConnection,
                             IP#192.168.75.123,
                             46000,
                             SendBuffer,
                             SendLen);
```

Funktionsweise bei einer UDP/IP-Verbindung

Da bei UDP/IP keine Verbindung zwischen zwei Netzwerkteilnehmern besteht, können Daten bei jedem Funktionsaufruf an einen anderen Teilnehmer oder anderen Port gesendet werden. Der Task bleibt bei diesem Funktionsaufruf stehen, bis die Daten gesendet wurden.

Sie erhalten keine Bestätigung, dass der andere Netzwerkteilnehmer die Daten empfangen hat.

UDP/IP-Client und -Server

Über eine UDP/IP-Client-Verbindung kann nur gesendet werden. Der Sende-Port wird vom Betriebssystem festgelegt.

Über eine UDP/IP-Server-Verbindung kann gesendet und empfangen werden. Als Sende-Port wird der bei der Verbindungseröffnung angegebene Port verwendet.

Verwandte Themen

- **Initialisieren der freiprogrammierbaren IP-Schnittstelle** (siehe Seite 28)
- **Verbindung öffnen** (siehe Seite 29)
- **Verbindung schließen** (siehe Seite 38)
- **Daten empfangen** (siehe Seite 35)

Daten empfangen

Einleitung

Daten können über eine zuvor geöffnete TCP/IP-Verbindung oder über die UDP/IP-Verbindung eines Servers empfangen werden.

Über die UDP/IP-Verbindung eines Clients können keine Daten empfangen, sondern nur gesendet werden.

Einschränkungen

Wenn Datenpakete über das Netzwerk empfangen werden, müssen diese vom Anwendungsprogramm abgeholt werden. Vom Betriebssystem der Steuerung werden pro Verbindung maximal vier Pakete zwischengespeichert. Alle weiteren Pakete werden verworfen.

Funktionsdeklaration

```
Function ConnectionReceiveData(IPConnection:Int,
                               Ref IPAddr:Int,
                               Ref IPPort:Int,
                               Ref ReceiveData,
                               DataLen:Int,
                               Timeout:Int):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
`IPConnection`	Handle	Rückgabewert der Funktion `ConnectionCreate()`
`IPAddr`	Adresse einer Variablen, um die IP-Adresse des Senders zu speichern	Nur bei UDP/IP-Server erforderlich
`IPPort`	Adresse einer Variablen, um die IP-Port-Nummer des Senders zu speichern	Nur bei UDP/IP-Server erforderlich
`ReceiveData`	Adresse des Empfangsdatenblocks	
`DataLen`	1 ... 4.000	Maximale Länge des Datenblocks in Byte
`Timeout`	0 ... 1.073.741.824 [ms]	0 = unendlich

Rückgabewert

Folgende Rückgabewerte sind möglich:

Rückgabewert	
> 0	Anzahl empfangener Datenbyte
-1	Fehler beim Empfang, z. B. Verbindung abgebrochen
-3	Ungültiges Handle, z. B. empfangen über einen UDP/IP-Client
-8	Timeout

Verwenden der Funktion bei einer TCP/IP-Verbindung

Wenn Daten über eine TCP/IP-Verbindung empfangen werden sollen, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionReceiveData(hConnection,
                                Dummy,
                                Dummy,
                                ReceiveBuffer,
                                sizeof(ReceiveBuffer),
                                T#10s);
```

Funktionsweise bei einer TCP/IP-Verbindung

Bei TCP/IP werden die Daten über eine Verbindung übertragen, die vorher geöffnet wurde. Deshalb ist die Angabe von IP-Adresse und IP-Port-Nummer nicht mehr erforderlich und werden in der Funktion ignoriert.

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

- Die Daten wurden empfangen.
- Ein Fehler ist aufgetreten.

Die Daten werden bei einer TCP/IP-Verbindung als Datenstrom übertragen.

Das Gerät arbeitet die Funktion in folgenden Stufen ab:

Stufe	Beschreibung	
1	Das Gerät wartet, bis Daten empfangen wurden, längstens jedoch die mitgegebene Timeout-Zeit.	
2	**Wenn ...**	**... dann ...**
	... der Timeout abgelaufen oder die Verbindung geschlossen wurde,	... wird die Funktion mit einer Fehlermeldung verlassen.
	... Daten empfangen wurden,	... werden sie in den mitgegebenen Empfangspuffer kopiert (höchstens jedoch bis zur mitgegebenen Anzahl) und bei Stufe 3 fortgefahren.
3	**Wenn ...**	**... dann ...**
	... mehr Daten empfangen wurden als in den Empfangspuffer kopiert werden konnten,	... werden diese vom Gerät gepuffert und können durch weitere Funktionsaufrufe abgeholt werden.
4	Die Funktion wird verlassen und die Anzahl der in den Empfangspuffer kopierten Daten zurückgegeben.	

Verwenden der Funktion bei einem UDP/IP-Server

Wenn Daten bei einem Server über eine UDP/IP-Verbindung empfangen werden sollen, können Sie die Funktion so aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionReceiveData(hConnection,
                                IPAddr,
                                IPPort,
                                ReceiveBuffer,
                                sizeof(ReceiveBuffer),
                                T#10s);
```

Funktionsweise bei einem UDP/IP-Server

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

- Alle Daten wurden empfangen.
- Ein Fehler ist aufgetreten.

Die Daten werden bei einer UDP/IP-Verbindung als Datagramme übertragen. Das Gerät arbeitet die Funktion in folgenden Stufen ab:

Stufe	Beschreibung	
1	Das Gerät wartet bis alle Daten eines Datagramms empfangen wurden, längstens jedoch die mitgegebene Timeout-Zeit.	
2	**Wenn ...**	**... dann ...**
	... der Timeout abgelaufen oder die Verbindung geschlossen wurde,	... wird die Funktion mit einer Fehlermeldung verlassen.
	... Daten empfangen wurden,	... werden sie in den mitgegebenen Empfangspuffer kopiert (höchstens jedoch bis zur mitgegebenen Anzahl) und bei Stufe 3 fortgefahren.
3	**Wenn ...**	**... dann ...**
	... mehr Daten empfangen wurden als in den Empfangspuffer kopiert werden konnten, also das gesendete Datagramm zu groß ist,	... werden diese verworfen.
4	IP-Adresse und IP-Port-Nummer des Senders werden in die mitgegebenen Variablen übertragen.	
5	Die Funktion wird verlassen und die Anzahl der in den Empfangspuffer kopierten Daten zurückgegeben.	

Verwandte Themen

- **Initialisieren der freiprogrammierbaren IP-Schnittstelle** (siehe Seite 28)
- **Verbindung öffnen** (siehe Seite 29)
- **Verbindung schließen** (siehe Seite 38)
- **Daten senden** (siehe Seite 33)

Verbindung schließen

Einleitung

Schließen Sie nicht mehr benötigte Verbindungen, da die Anzahl gleichzeitig geöffneter Verbindungen begrenzt ist.

Funktionsdeklaration

```
Function ConnectionDelete(IPConnection:Int):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
`IPConnection`	Handle	Rückgabewert der Funktion `ConnectionCreate()`

Rückgabewert

Folgende Rückgabewerte sind möglich:

Rückgabewert	
0	Verbindung geschlossen und gelöscht
-1	Ungültiges Handle

Verwenden der Funktion

So können Sie die Funktion aufrufen und den Rückgabewert einer Variablen zur weiteren Auswertung zuweisen:

```
Result := ConnectionDelete(hConnection);
```

Verwandte Themen

- **Verbindung öffnen** (siehe Seite 29)
- **Daten senden** (siehe Seite 33)
- **Daten empfangen** (siehe Seite 35)
- **Initialisieren der freiprogrammierbaren IP-Schnittstelle** (siehe Seite 28)

2.2.2 Register

Einleitung

Dieses Kapitel beschreibt die Register des Geräts, in denen die aktuelle Verbindungsliste der freiprogrammierbaren IP-Schnittstelle enthalten ist. Die Register können zu Debug- oder Diagnosezwecken verwendet werden. Weitere Funktionen, wie Verbindung öffnen oder schließen, können hierüber nicht ausgelöst werden.

Inhalt

Thema	Seite
Registernummern	40
Registerbeschreibung	41

Registernummern

Einleitung

Die Daten jeweils einer Verbindung erscheinen in den Registern eines zusammenhängenden Registerblocks. Die Basisregisternummer dieses Blocks ist steuerungsabhängig.

Registernummern

Basisregisternummer	Registernummern
350000	350000 ... 350007

Registernummer ermitteln

In diesem Kapitel ist jeweils nur die letzte Ziffer der Registernummer angegeben, z. B. MR 1. Addieren Sie zu dieser Ziffer die Basisregisternummer des jeweiligen Geräts, z. B. 350000, um die vollständige Registernummer zu ermitteln.

Registerübersicht

Register	Beschreibung
MR 0	Verbindungsauswahl
MR 1	Verbindungstyp
MR 2	Transportprotokoll
MR 3	IP-Adresse
MR 4	IP-Port-Nummer
MR 5	Zustand
MR 6	Anzahl gesendeter Bytes
MR 7	Anzahl empfangener Bytes
MR 8	Anzahl verworfener Bytes
MR 9	Anzahl verworfener Pakete

Registerbeschreibung

Einleitung

Das Betriebssystem verwaltet die geöffneten Verbindungen in einer Liste. Mit Hilfe des Modulregisters MR 0 *Verbindungsauswahl* werden die Verbindungsdaten einer Verbindung in die anderen Register des Registerblocks kopiert.

MR 0

Verbindungsauswahl

Die Anwahl der Verbindungen geschieht, indem Werte in dieses Register geschrieben werden. Dieses Register zeigt, ob die folgenden Register Verbindungsdaten enthalten.

Modulregistereigenschaften		
Werte lesen	0	Verbindung vorhanden
	-1	Verbindung nicht vorhanden

Modulregistereigenschaften		
Werte schreiben	0	Erste Verbindung in der Liste anwählen
	> 0	Nächste Verbindung in der Liste anwählen
	< 0	Vorherige Verbindung in der Liste anwählen

MR 1

Verbindungstyp

Der Wert in diesem Register zeigt an, ob es sich um eine Client- oder eine Serververbindung handelt.

Modulregistereigenschaften		
Werte	1	Client
	2	Server

MR 2

Transportprotokoll

Der Wert in diesem Register zeigt die Art des Transportprotokolls an, UDP oder TCP.

Modulregistereigenschaften		
Werte	1	UDP
	2	TCP

MR 3 **IP-Adresse**

Der Wert in diesem Register zeigt die konfigurierte IP-Adresse an.

Modulregistereigenschaften	
Werte	0.0.0.0 ... 255.255.255.255

MR 4 **IP-Port-Nummer**

Der Wert in diesem Register zeigt die konfigurierte IP-Port-Nummer an.

Modulregistereigenschaften	
Werte	0 ... 65.535

MR 5 **Zustand**

Der Wert in diesem Register zeigt an, in welchem Zustand sich die Verbindung befindet.

Modulregistereigenschaften		
Werte	0	Verbindung geschlossen
	1	Verbindung wird geöffnet
	2	Verbindung ist geöffnet
	3	TCP/IP-Server: Warten auf Verbindungsanfrage von Client
	4	Interne Verwendung

MR 6 **Anzahl gesendeter Bytes**

Der Wert in diesem Register zeigt die Anzahl der über diese Verbindung gesendeten Datenbytes an. Da es sich um ein vorzeichenbehaftetes 32-Bit-Register handelt und die gesendeten Bytes jeweils hinzuaddiert werden, können Zahlenüberläufe vom positiven zum negativen Maximalwert auftreten.

Modulregistereigenschaften	
Werte	-2.147.483.648 ... 2.147.483.647

MR 7 **Anzahl empfangener Bytes**

Der Wert in diesem Register zeigt die Anzahl der über diese Verbindung empfangenen Datenbytes an. Da es sich um ein vorzeichenbehaftetes 32-Bit-Register handelt und die empfangenen Bytes jeweils hinzuaddiert werden, können Zahlenüberläufe vom positiven zum negativen Maximalwert auftreten.

Modulregistereigenschaften	
Werte	-2.147.483.648 ... 2.147.483.647

MR 8 **Anzahl verworfener Bytes**

Der Wert in diesem Register zeigt die Anzahl der Datenbytes an, die nicht empfangen werden konnten, weil das Anwendungsprogramm die zwischengespeicherten Datenbytes nicht abgeholt hat.

Modulregistereigenschaften	
Werte	0 ... 2.147.483.647

MR 9 **Anzahl verworfener Pakete**

Der Wert in diesem Register zeigt die Anzahl der Datenpakete an, die nicht empfangen werden konnten, weil das Anwendungsprogramm die zwischengespeicherten Datenpakete nicht abgeholt hat.

Modulregistereigenschaften	
Werte	0 ... 2.147.483.647

3 Beseitigte Software-Bugs

Einleitung

Dieses Kapitel beschreibt die in der neuen Betriebssystemversion beseitigten Software-Bugs.

Inhalt

Thema	Seite
Ungültiges Verbindungs-Handle bei der freiprogrammierbaren IP-Schnittstelle	45
Lange Schlüsselnamen führen zum Absturz	46
Umbenennen einer Datei mit maximaler Namenslänge führt zum Absturz	47
Eingangswerte eines Netzwerkteilnehmers sind eingefroren	48
Fehlerhafte Anzeige von Float-Registern im JetSym-Oszilloskop	49
Inkonsistente Register bei der IP-Konfiguration	50

Ungültiges Verbindungs-Handle bei der freiprogrammierbaren IP-Schnittstelle

Fehlerbild

Wenn bei `ConnectionReceiveData()`, `ConnectionSendData()` oder `ConnectionDelete()` ein ungültiges Verbindungs-Handle mitgegeben wird stürzt die Steuerung ab.

Betroffene Versionen/Revisionen

Der Fehler tritt in folgenden Versionen/Revisionen auf:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	< 1.28.0.00 < 1.28.0.00 < 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Abhilfe/Workaround

Prüfen Sie vor der Verwendung das Verbindungs-Handle auf Gültigkeit.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen/Revisionen behoben:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	1.28.0.00 1.28.0.00 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Lange Schlüsselnamen führen zum Absturz

Fehlerbild

Nachdem für mehrere Schlösser/Schlüssel in der Datei **/System/keys.ini** längere Namen zugeordnet wurden stürzt die Steuerung ab.

Betroffene Versionen/Revisionen

Der Fehler tritt in folgenden Versionen/Revisionen auf:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	< 1.28.0.00 < 1.28.0.00 < 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Abhilfe/Workaround

Addieren Sie die Längen (Anzahl Zeichen) aller Namen in der Datei **/System/keys.ini.** Stellen Sie sicher, dass diese Summe den Wert 224 nicht übersteigt.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen/Revisionen behoben:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	1.28.0.00 1.28.0.00 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Umbenennen einer Datei mit maximaler Namenslänge führt zum Absturz

Fehlerbild

Wenn eine Datei umbenannt wird und der neue Namen hat die maximal zulässige Länge von 63 Zeichen stürzt die Steuerung ab.

Betroffene Versionen/Revisionen

Der Fehler tritt in folgenden Versionen/Revisionen auf:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	< 1.28.0.00 < 1.28.0.00 < 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Abhilfe/Workaround

Stellen Sie sicher, dass die Länge des neuen Dateinamens 62 Zeichen nicht übersteigt.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen/Revisionen behoben:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	1.28.0.00 1.28.0.00 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Eingangswerte eines Netzwerkteilnehmers sind eingefroren

Fehlerbild

Wenn mehrere Netzwerkteilnehmer mit einer Steuerung über Publish/ Subscribe kommunizieren ändern sich die Eingangswerte eines der Teilnehmer nicht mehr. Die Ausgänge zu diesem Teilnehmer funktionieren weiterhin. Ebenfalls die Eingänge und Ausgänge zu den anderen Teilnehmern. Es werden keine Fehler angezeigt. Das Problem lässt sich durch einen Neustart des Subscribers auf der Steuerung beheben.

Betroffene Versionen/Revisionen

Der Fehler tritt in folgenden Versionen/Revisionen auf:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	< 1.28.0.00 < 1.28.0.00 < 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Abhilfe/Workaround

Verbinden Sie bei jedem Netzwerkteilnehmer einen nicht verwendeten Ausgang mit einem nicht verwendeten Eingang. Schalten Sie im Anwendungsprogramm den Ausgang ein und aus. Prüfen Sie ob der Eingang dem Zustand des Ausgangs folgt. Wenn dies nicht mehr der Fall ist, starten Sie den Subscriber auf der Steuerung neu.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen/Revisionen behoben:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	1.28.0.00 1.28.0.00 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Fehlerhafte Anzeige von Float-Registern im JetSym-Oszilloskop

Fehlerbild

Bei der Oszilloskop-Aufzeichnung in JetSym werden bei Float-Registern falsche Werte angezeigt.

Betroffene Versionen/Revisionen

Der Fehler tritt in folgenden Versionen/Revisionen auf:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	< 1.28.0.00 < 1.28.0.00 < 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Abhilfe/Workaround

Verwenden Sie im JetSym-Oszilloskop nur Integer-Register.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen/Revisionen behoben:

Betriebssystemversion	JC-340/350 JC-360/365 (MC) JC-940MC	1.28.0.00 1.28.0.00 1.10.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Inkonsistente Register bei der IP-Konfiguration

Fehlerbild

Beim Ändern der IP-Einstellungen in den Register 101200 bis 101202 werden die neuen Werte sofort in den Register 104531 bis 104533 sichtbar. Sie werden jedoch nicht sofort wirksam.

Betroffene Versionen/Revisionen

Der Fehler tritt in folgenden Versionen/Revisionen auf:

Betriebssystemversion	JC-340/350 JC-360/365 (MC)	< 1.28.0.00 < 1.28.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Abhilfe/Workaround

Eine Abhilfe oder ein Workaround ist in den betroffenen Version nicht möglich.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen/Revisionen behoben:

Betriebssystemversion	JC-340/350 JC-360/365 (MC)	1.28.0.00 1.28.0.00
Hardwarerevision	Nicht relevant	
Konfiguration oder Betriebsart	Nicht relevant	

Jetter automation

Jetter AG
Gräterstraße 2
71642 Ludwigsburg | Germany

Tel +49 7141 2550-0
Fax +49 7141 2550-425
info@jetter.de
www.jetter.de

We automate your success.