JC-340

Versions-Update

von V. 1.08 auf V. 1.09



Versions-Update

Jetter

Version 1.01

Januar 2011 / Printed in Germany

Die Firma Jetter AG behält sich das Recht vor, Änderungen an Ihren Produkten vorzunehmen, die der technischen Weiterentwicklung dienen. Diese Änderungen werden nicht notwendigerweise in jedem Einzelfall dokumentiert.

Dieses Versions-Update und die darin enthaltenen Informationen wurden mit der gebotenen Sorgfalt zusammengestellt. Die Firma Jetter AG übernimmt jedoch keine Gewähr für Druckfehler oder andere Fehler oder daraus entstehende Schäden.

Die in diesem Buch genannten Marken und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhalter.

Inhalt

1 Einleitung 5

Betriebssystem-Update ... 6
JC-340 Version-Update Übersicht ... 7

2 Erweiterungen 10

2.1 Diverse Funktionserweiterungen und Änderungen ... 11
Dummy-Module am JX3-Systembus ... 12
2.2 FTP-Client ... 13
2.2.1 Programmierung ... 14
Initialisieren des FTP-Client ... 15
Verbindung zum FTP-Server öffnen ... 16
Verbindung schließen ... 18
Datei vom FTP-Server lesen ... 19
Datei auf dem FTP-Server schreiben ... 21
Datei auf dem FTP-Server löschen ... 23
Verzeichnis auf dem FTP-Server wechseln ... 25
Verzeichnis auf dem FTP-Server anlegen ... 27
Verzeichnis auf dem FTP-Server löschen ... 29
Aktuelles Verzeichnis auf dem FTP-Server ermitteln ... 31
2.2.2 Register ... 33
Registernummern ... 34
Registerbeschreibung ... 35

3 Automatisches Kopieren von Steuerungsdaten 38

3.1 Funktionsweise ... 40
AutoCopy-Funktion starten ... 41
AutoCopy-Funktion ausführen ... 42
Betriebsart AutoCopy beenden ... 44
3.2 Die Datei 'autocopy.ini' ... 45
Die Sektion [OPTIONS] ... 46
Die Kommandosektionen ... 47
Beispiel für eine Kommandodatei ... 55
3.3 Protokolldatei ... 58
Dateiinhalt ... 59
3.4 Datendateien ... 60
Dateiformat ... 61
3.5 Systemkommandoregister ... 62
Beschreibung der Systemkommandoregister ... 63
Beschreibung der Systemkommandos ... 65

4 Beseitigte Software-Bugs 67

Achsbefehle Pos, ActualPos, AxArr ... 68
Taskrestart bricht einen Delay-Befehl nicht ab ... 69
Absturz bei fehlender Bibliothek ... 70
NetCopyVarToReg funktioniert mit Float-Variablen nicht richtig ... 71
Serielle Schnittstelle wird nach dem Booten nicht richtig initialisiert ... 72

Speichern von Fließkommazahlregistern in Datendateien funktioniert nicht 73
Eingangsänderung vom JX2-ID8 werden nicht erkannt..... 74
Schnelle Zustandsänderung von digitalen Eingängen am JX2-Systembus 75

1 Einleitung

Einleitung

Dieses Kapitel zeigt eine Historie der Betriebssystem-Versionen der Steuerung JC-340.

Betriebssystem-Update - wozu?

Ein Betriebssystem-Update bietet Ihnen die folgenden Möglichkeiten:

- die Funktionalität erweitern
- Software-Bugs beheben
- einen bestimmten Betriebssystem-Stand übertragen, beispielsweise bei kundenspezifischer Freigabe einer Betriebssystem-Version

Inhalt

Thema	Seite
Betriebssystem-Update	6
JC-340 Version-Update Übersicht	7

Betriebssystem-Update

OS-Datei zum Betriebssystem-Update

Zum Betriebssystem-Update benötigen Sie die folgende Datei:

OS-Datei	Beschreibung
JC-340_1.09.0.0.os	Betriebssystem-Datei für JC-340 mit der Version 1.09

Download der OS-Datei

Die Jetter AG stellt OS-Dateien zum Betriebssystem-Download auf unserer **Homepage http://www.jetter.de** zur Verfügung. Sie finden die OS-Dateien im Support-Bereich bzw. über die Quicklinks der Steuerung JC-340.

Betriebssystem-Update mit JetSym

Führen Sie zum Betriebssystem-Update 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 -> Betriebssystem-Update oder Klicken auf den Button OS-Update im CPU-Fenster des Hardware-Managers
4	Auswahl der OS-Datei
5	Start des Betriebssystem-Updates mit OK
6	**Ergebnis:** Nach Power Off / Power On startet das neue Betriebssystem.

Mindestvoraussetzungen

Für die Programmierung des JC-340 ab Version 1.09 ist JetSym ab Version 4.2 nötig.

JC-340 Version-Update Übersicht

V 1.04

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystem-Version 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	✓	
Betriebssystem-Update:		
Über FTP: Bei Fertigmeldung ist das OS wirklich gespeichert		✓
Update auf JX2-Slave bei gleichzeitigem Registerzugriff blockiert die Kommunikation		✓
Anwenderprogramm:		
Task-Wechsel konnte ausbleiben		✓
Fehleranzeige wenn Datei "/app/start.ini" nicht korrekt		✓
Display-Befehle:		
Umleitung auf JX2-SER1 funktioniert nur, wenn JX2-PRN1 auch konfiguriert ist		✓

V 1.05

Die folgende Tabelle zeigt eine Übersicht über neu hinzugekommene Funktionen und die beseitigten Software-Bugs der Betriebssystem-Version 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	✓	
Anwenderprogramm:		
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		✓
Anwenderregister:		
Einrichten des Registertyps ohne Starten des Anwenderprogramms	✓	
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 Betriebssystem-Version 1.08:

Funktion	Neu	Bug
System-Konfiguration:		
System-Rechte 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		✓
Anwenderprogramm:		
NetCopyList-Funktionen	✓	
StrCopy()		✓
Absturz bei "ungültigem" Anwenderprogramm		✓
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 Betriebssystem-Version 1.09:

Funktion	Neu	Bug
System:		
Systemkommandoregister	✓	
JX2-Systembus: V1.13.0.00		
Eingangsänderungen am JX2-ID8		✓
Schnelle Eingangsänderungen		✓
Anwenderprogramm:		
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		✓

2 Erweiterungen

Einleitung

Dieses Kapitel beschreibt die in der neuen Betriebssystem-Version hinzugekommenen oder erweiterten Funktionen.

Inhalt

Thema	Seite
Diverse Funktionserweiterungen und Änderungen	11
FTP-Client	13
Automatisches Kopieren von Steuerungsdaten	38
Systemkommandoregister	62

2.1 Diverse Funktionserweiterungen und Änderungen

Einleitung

Dieses Kapitel beschreibt diverse Funktionserweiterungen und Änderungen.

Inhalt

Thema	Seite
Dummy-Module am JX3-Systembus	12

Dummy-Module am JX3-Systembus

Dummy-Module

Ein Dummy-Modul ist ein Modul am JX3-Systembus, das tatsächlich gar nicht vorhanden ist. Die Steuerung JC-340 behandelt Dummy-Module bei der Vergabe der I/O-Modulnummern wie vorhandene Module.

Durch ein Dummy-Modul kann der Anwender ein nicht vorhandenes Modul zwischen vorhandene Module einfügen. So können Platzhalter zum Beispiel für zukünftige Erweiterungen oder Anlagenoptionen geschaffen werden.

Registerübersicht

Der Wert in dem Register zur Konfiguration der Dummy-Module ist remanent. Eine Änderung wird erst nach einem Neustart der Steuerung JC-340 wirksam.

Register	Beschreibung
R 100002023	I/O-Dummy-Module

Konfiguration von Dummy-Modulen

Schritt	Vorgehen
1	Ändern Sie die Konfiguration der Dummy-Module in der Steuerung über R 100002023.
2	Schalten Sie die Steuerung aus.
3	Schalten Sie die Steuerung wieder ein. **Ergebnis:** Die Steuerung hat den JX3-Systembus mit den konfigurierten Dummy-Modulen in Betrieb genommen. Die Dummy-Module sind über das Modul-Array lesbar.

2.2 FTP-Client

Der FTP-Client

Der FTP-Client in der Steuerung JC-340 ermöglicht den Zugriff auf Dateien und Verzeichnisse eines fernen Netzwerkteilnehmers aus dem Anwenderprogramm. Der FTP-Client kommuniziert hierzu mit dem FTP-Server dieses Netzwerkteilnehmers.

Funktionen

Folgende Funktionen sind dabei möglich:

- Anlegen von Verzeichnissen im fernen Dateisystem
- Löschen von Verzeichnissen im fernen Dateisystem
- Kopieren von Dateien vom lokalen Dateisystem zum fernen Dateisystem
- Kopieren von Dateien vom fernen Dateisystem zum lokalen Dateisystem

Voraussetzungen

- Die Funktionalität des FTP-Clients setzt grundsätzliche Kenntnisse über FTP-Verbindungen und Dateisysteme voraus.
- Die IP-Adresse des FTP-Servers muss bekannt sein.
- Wenn die IP-Adresse des FTP-Servers nicht bekannt ist, muss die Namensauflösung über einen DNS-Server möglich sein.
- Die Parameter für Anmeldung am FTP-Server (User-Name und Passwort) müssen bekannt sein.
- Für die Programmierung muss JetSym ab der Version 4.3 verwendet werden.

Verarbeitung im Anwenderprogramm

- Die Steuerung führt immer nur einen FTP-Zugriff aus, bis dieser abgeschlossen ist.
- Der entsprechende Task im Anwenderprogramm bleibt an dem Befehl so lange stehen, bis der Zugriff abgeschlossen ist.
- Während dieser Zeit werden andere Tasks des Anwenderprogramms bearbeitet.
- Während ein FTP-Zugriff eines Task gerade läuft, werden andere Tasks, die einen FTP-Befehl aufrufen, solange blockiert, bis der laufende Zugriff abgeschlossen ist.

Inhalt

Thema	Seite
Programmierung	14
Register	33

2.2.1 Programmierung

Einleitung

Der FTP-Client ermöglicht, vom Anwenderprogramm auf Dateien und Verzeichnisse eines Netzwerkteilnehmers zuzugreifen. Verwendung finden hierbei Aufrufe von Funktionen, die im Sprachumfang der Steuerung enthalten sind. Führen Sie zur Programmierung folgende Schritte aus:

Schritt	Vorgehen
1	FTP-Client initialisieren
2	Verbindung(en) zu FTP-Server(n) öffnen
3	Daten übertragen
4	Verbindungen(en) schließen

Einschränkungen

Im Anwenderprogramm sollten die Tasks, die den FTP-Client bedienen, nicht durch `TaskBreak` angehalten oder durch `TaskRestart` neu gestartet werden, während von der Steuerung gerade eine dieser Funktionen bearbeitet wird. Sonst kann es vorkommen, dass die Steuerung diese Funktion nicht vollständig beendet und damit alle weiteren Funktionsaufrufe des FTP-Client blockiert werden.

Inhalt

Thema	Seite
Initialisieren des FTP-Client	15
Verbindung zum FTP-Server öffnen	16
Verbindung schließen	18
Datei vom FTP-Server lesen	19
Datei auf dem FTP-Server schreiben	21
Datei auf dem FTP-Server löschen	23
Verzeichnis auf dem FTP-Server wechseln	25
Verzeichnis auf dem FTP-Server anlegen	27
Verzeichnis auf dem FTP-Server löschen	29
Aktuelles Verzeichnis auf dem FTP-Server ermitteln	31

Initialisieren des FTP-Client

Einleitung

Die Initialisierung muss mindestens einmal bei jedem Start des Anwenderprogramms durchgeführt werden.

Funktionsdeklaration

```
Function FtpInitialize():Int;
```

Funktionsergebnis

Folgender Rückgabewert ist möglich:

Funktionsergebnis	
0	immer

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Result := FtpInitialize();
```

Funktionsweise

Die Steuerung arbeitet die Funktion in folgenden Stufen ab:

Stufe	Beschreibung
1	Alle geöffneten Verbindungen des FTP-Client werden geschlossen.
2	Alle betriebssysteminternen Datenstrukturen des FTP-Client werden initialisiert.

Verbindung zum FTP-Server öffnen

Einleitung

Bevor Daten gesendet oder empfangen werden können, muss eine Verbindung zu einem FTP-Server geöffnet werden. Hierbei erfolgt auch die Anmeldung mit User-Name und Passwort (Login).

Funktionsdeklaration

```
Function FtpConnect(Const Ref ServerAddr: String,
                    Const Ref UserName: String,
                    Const Ref PassWord: String):Handle;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
ServerAddr	IP-Adresse oder Name	Namensauflösung über DNS-Server
UserName	Anmeldename	Login
PassWord	Anmeldepasswort	Login

Funktionsergebnis

Bei einem positiven Funktionsergebnis konnte die Verbindung geöffnet und die Anmeldung erfolgreich durchgeführt werden. Bei einem Funktionsergebnis von '0' ist ein Fehler aufgetreten, ohne das die Verbindung geöffnet werden konnte.

Funktionsergebnis	
> 0	Ein positives Funktionsergebnis muss in einer Variablen gesichert werden, da es als Handle bei den Funktionen zum Empfangen und Senden von Daten über diese Verbindung und Schließen der Verbindung mitgegeben werden muss
0	Fehler beim Verbindungsaufbau oder der Anmeldung beim FTP-Server

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
    End_Var;

    ....
    FtpHandle := FtpConnect('192.168.123.45',
                            'Administrator',
                            'AdminPassword');
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis die Verbindung aufgebaut werden konnte oder der Timeout des FTP-Client abgelaufen ist. Die Funktion wird in folgenden Stufen abgearbeitet:

<table>
<tr><th>Stufe</th><th>Beschreibung</th></tr>
<tr><td>1</td><td>Die Steuerung versucht eine TCP/IP-Verbindung zum FTP-Server mit der IP-Adresse 192.168.123.45 aufzubauen.</td></tr>
<tr><td>2</td><td>
<table>
<tr><th>Wenn ...</th><th>... Dann ...</th></tr>
<tr><td>der Netzwerkteilnehmer die Verbindung akzeptiert hat</td><td>wird mit Stufe 3 fortgefahren</td></tr>
<tr><td>die Verbindung nicht aufgebaut werden konnte und die Time-outzeit noch nicht abgelaufen ist</td><td>wird mit Stufe 1 fortgefahren</td></tr>
<tr><td>ein Fehler aufgetreten ist oder der Timeout abgelaufen ist</td><td>wird die Funktion beendet und der Wert 0 zurückgeliefert</td></tr>
</table>
</td></tr>
<tr><td>3</td><td>Die Steuerung meldet sich mit dem User-Namen "Administrator" und dem Passwort "AdminPassword" am FTP-Server an</td></tr>
<tr><td>4</td><td>
<table>
<tr><th>Wenn ...</th><th>... Dann ...</th></tr>
<tr><td>der FTP-Server die Anmeldung akzeptiert hat</td><td>wird die Funktion beendet und ein positiver Wert als Handle zum weiteren Zugriff auf diese Verbin-dung zurückgeliefert</td></tr>
<tr><td>der FTP-Server die Anmeldung nicht akzeptiert hat (z. B. ungül-tiger User-Name oder falsches Passwort)</td><td>wird die Funktion beendet und der Wert 0 zurückgeliefert</td></tr>
</table>
</td></tr>
</table>

Verwandte Themen:

- **Verbindung schließen** auf Seite 18

Verbindung schließen

Einleitung

Schließen Sie nicht mehr benötigte Verbindungen, da dadurch der Verwaltungsaufwand für die Verbindungen innerhalb der Steuerung reduziert wird.

Funktionsdeklaration

```
Function FtpDisconnect(FtpConnection:Handle):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung

Funktionsergebnis

Folgende Funktionsergebnisse sind möglich:

Funktionsergebnis	
0	Verbindung geschlossen und gelöscht
-1	ungültiges Handle
-2, -3	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)

Verwenden der Funktion

So kann die Funktion aufgerufen werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
    End_Var;

    ....
    Result := FtpDisconnect(FtpHandle);
    ....
End_Task;
```

Verwandte Themen:

- **Verbindung zum FTP-Server öffnen** auf Seite 16

Datei vom FTP-Server lesen

Einleitung

Mit dieser Funktion kann eine Datei von einem FTP-Server gelesen und im lokalen Dateisystem gespeichert werden.

Funktionsdeklaration

```
Function FtpFileRead(FtpConnection:Handle,
                     Const Ref ServerFile: String,
                     Const Ref ClientFile: String):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung
ServerFile	Dateiname	Name der Datei, die vom FTP-Server gelesen werden soll
ClientFile	Dateiname	Name der Datei, unter dem die gelesene Datei gespeichert werden soll

Funktionsergebnis

Bei einem negativen Funktionsergebnis ist ein Fehler aufgetreten. Bei einem Funktionsergebnis von "0" konnte die Datei vom FTP-Server gelesen und lokal gespeichert werden.

Funktionsergebnis	
0	kein Fehler
-1	ungültiges Handle
-2 , -6	Fehler beim Speichern der lokalen Datei
-3, -5, -7, -8	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)
-4	Fehlermeldung vom FTP-Server (z. B. Datei nicht vorhanden)

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
    End_Var;

    ....
    Result := FtpFileRead(FtpHandle,
                          'ServerTestFile.txt',
                          'LocalTestFile.txt');
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis die Datei "ServerTestFile.txt" vom FTP-Server gelesen und im lokalen Dateisystem unter dem Namen "LocalTestFile.txt" gespeichert werden konnte oder ein Fehler aufgetreten ist. Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung
1	Die Steuerung schickt zum FTP-Server das Kommando, dass die Datei "ServerTestFile.txt" gelesen werden soll
2	Die Steuerung empfängt den Inhalt der Datei und schreibt ihn in die Datei "LocalTestFile.txt"
3	<table><tr><th>Wenn ...</th><th>... Dann ...</th></tr><tr><td>keine Fehler aufgetreten sind</td><td>wurde die Datei erfolgreich kopiert, die Funktion beendet und der Wert 0 zurückgeliefert</td></tr><tr><td>Fehler aufgetreten sind</td><td>wird die Funktion beendet und ein negativer Wert zurückgeliefert</td></tr></table>

Dateinamen

- Der Funktionsparameter für die lokale Datei kann auch den Pfad zu dieser Datei enthalten (z. B. `"/Data/TestFiles/LocalTestFile.txt"`).
- Der Funktionsparameter für die Datei auf dem FTP-Server kann auch den Pfad zu dieser Datei enthalten, wenn dies vom Dateisystem unterstützt wird. Ansonsten muss zunächst mit dem Befehl `FtpDirChange(...)` das entsprechende Verzeichnis eingestellt werden.
- Das Dateisystem eines <Produktnamen> unterstützt beide Varianten.

Verwandte Themen:

- **Datei auf dem FTP-Server schreiben** auf Seite 21

Datei auf dem FTP-Server schreiben

Einleitung

Mit dieser Funktion kann eine Datei vom lokalen Dateisystem auf einen FTP-Server übertragen werden.

Funktionsdeklaration

```
Function FtpFileWrite(FtpConnection:Handle,
                      Const Ref ServerFile: String,
                      Const Ref ClientFile: String):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung
ServerFile	Dateiname	Name der Datei, unter welchem die Datei auf dem FTP-Server gespeichert werden soll
ClientFile	Dateiname	Name der Datei, die vom lokalen Dateisystem gelesen werden soll

Funktionsergebnis

Bei einem negativen Funktionsergebnis ist ein Fehler aufgetreten. Bei einem Funktionsergebnis von "0" konnte die Datei gelesen und auf dem FTP-Server gespeichert werden.

Funktionsergebnis	
0	kein Fehler
-1	ungültiges Handle
-2	Fehler beim Lesen der lokalen Datei (z. B. Datei nicht vorhanden)
-3, -5, -8	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)
-4, -7	Fehlermeldung vom FTP-Server (z. B. Datei kann nicht angelegt werden)

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
    End_Var;

    ....
    Result := FtpFileWrite(FtpHandle,
                           'ServerTestFile.txt',
                           'LocalTestFile.txt');
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis die Datei "LocalTestFile.txt" vom lokalen Dateisystem gelesen und im FTP-Server unter dem Namen "ServerTestFile.txt" gespeichert werden konnte oder ein Fehler aufgetreten ist. Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung
1	Die Steuerung schickt zum FTP-Server das Kommando, dass die Datei "ServerTestFile.txt" gespeichert werden soll.
2	Die Steuerung sendet den Inhalt der Datei "LocalTestFile.txt".
3	<table><tr><th>Wenn ...</th><th>... Dann ...</th></tr><tr><td>keine Fehler aufgetreten sind</td><td>wurde die Datei erfolgreich kopiert, die Funktion beendet und der Wert 0 zurückgeliefert</td></tr><tr><td>Fehler aufgetreten sind</td><td>wird die Funktion beendet und ein negativer Wert zurückgeliefert</td></tr></table>

Dateinamen

- Der Funktionsparameter für die lokale Datei kann auch den Pfad zu dieser Datei enthalten (z. B. `"/Data/TestFiles/LocalTestFile.txt"`).
- Der Funktionsparameter für die Datei auf dem FTP-Server kann auch den Pfad zu dieser Datei enthalten, wenn dies vom Dateisystem unterstützt wird. Ansonsten muss zunächst mit dem Befehl `FtpDirChange(...)` das entsprechende Verzeichnis eingestellt werden.
- Das Dateisystem eines <Produktnamen> unterstützt beide Varianten.

Verwandte Themen:

- **Datei vom FTP-Server lesen** auf Seite 19

Datei auf dem FTP-Server löschen

Einleitung

Mit dieser Funktion kann eine Datei auf dem FTP-Server gelöscht werden.

Funktionsdeklaration

```
Function FtpFileRemove(FtpConnection:Handle,
                       Const Ref ServerFile: String):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung
ServerFile	Dateiname	Name der Datei, die gelöscht werden soll

Funktionsergebnis

Bei einem negativen Funktionsergebnis ist ein Fehler aufgetreten. Bei einem Funktionsergebnis von "0" konnte die Datei auf dem FTP-Server gelöscht werden.

Funktionsergebnis	
0	kein Fehler
-1	ungültiges Handle
-2	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)
-3	Fehlermeldung vom FTP-Server (z. B. Datei nicht vorhanden)

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
    End_Var;

    ....
    Result := FtpFileRemove(FtpHandle,
                            'ServerTestFile.txt');
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis die Datei "ServerTestFile.txt" vom FTP-Server gelöscht werden konnte oder ein Fehler aufgetreten ist. Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung
1	Die Steuerung schickt zum FTP-Server das Kommando, dass die Datei "ServerTestFile.txt" gelöscht werden soll.
2	<table><tr><th>Wenn ...</th><th>... Dann ...</th></tr><tr><td>keine Fehler aufgetreten sind</td><td>wurde die Datei gelöscht, die Funktion beendet und der Wert 0 zurückgeliefert</td></tr><tr><td>Fehler aufgetreten sind</td><td>wird die Funktion beendet und ein negativer Wert zurückgeliefert</td></tr></table>

Dateinamen

- Der Funktionsparameter für die lokale Datei kann auch den Pfad zu dieser Datei enthalten (z. B. `"/Data/TestFiles/LocalTestFile.txt"`).
- Der Funktionsparameter für die Datei auf dem FTP-Server kann auch den Pfad zu dieser Datei enthalten, wenn dies vom Dateisystem unterstützt wird. Ansonsten muss zunächst mit dem Befehl `FtpDirChange(...)` das entsprechende Verzeichnis eingestellt werden.
- Das Dateisystem eines <Produktnamen> unterstützt beide Varianten.

Verzeichnis auf dem FTP-Server wechseln

Einleitung

Mit dieser Funktion wird das aktuelle Verzeichnis auf dem FTP-Server gewechselt.

Funktionsdeklaration

```
Function FtpDirChange(FtpConnection:Handle,
                      Const Ref ServerDir: String):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung
ServerDir	Verzeichnisname	Name des Verzeichnisses, in welches gewechselt werden soll

Funktionsergebnis

Bei einem negativen Funktionsergebnis ist ein Fehler aufgetreten. Bei einem Funktionsergebnis von "0" konnte in das Verzeichnis auf dem FTP-Server gewechselt werden.

Funktionsergebnis	
0	kein Fehler
-1	ungültiges Handle
-2	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)
-3	Fehlermeldung vom FTP-Server (z. B. Verzeichnis nicht vorhanden)

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
    End_Var;

    ....
    Result := FtpDirChange(FtpHandle,
                           'DataFiles');
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis im Dateisystem des FTP-Servers in das Unterverzeichnis "DataFiles" gewechselt werden konnte oder ein Fehler aufgetreten ist. Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung
1	Die Steuerung schickt zum FTP-Server das Kommando, dass in das Verzeichnis "DataFiles" gewechselt werden soll.
2	<table><tr><th>Wenn ...</th><th>... Dann ...</th></tr><tr><td>keine Fehler aufgetreten sind</td><td>ist das neue Verzeichnis eingestellt, die Funktion beendet und der Wert 0 zurückgeliefert</td></tr><tr><td>Fehler aufgetreten sind</td><td>wird die Funktion beendet und ein negativer Wert zurückgeliefert</td></tr></table>

Verzeichnisnamen

- Der Funktionsparameter für das Verzeichnis auf dem FTP-Server kann auch den kompletten Pfad - über mehrere Verzeichnisebenen - zu diesem Verzeichnis enthalten, wenn dies vom Dateisystem unterstützt wird.
- Ansonsten muss zunächst mit dem Befehl `FtpDirChange(...)` von einer Verzeichnisebene zur nächsten verzweigt werden, bis das entsprechende Verzeichnis erreicht ist.
- Das Dateisystem eines JetControl unterstützt beide Varianten.

Verwandte Themen:

- **Aktuelles Verzeichnis auf dem FTP-Server ermitteln** auf Seite 31

Verzeichnis auf dem FTP-Server anlegen

Einleitung

Mit dieser Funktion wird ein neues Verzeichnis auf dem FTP-Server angelegt.

Funktionsdeklaration

```
Function FtpDirCreate(FtpConnection:Handle,
                      Const Ref ServerDir: String):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung
ServerDir	Verzeichnisname	Name des Verzeichnisses, das angelegt werden soll

Funktionsergebnis

Bei einem negativen Funktionsergebnis ist ein Fehler aufgetreten. Bei einem Funktionsergebnis von "0" konnte das Verzeichnis auf dem FTP-Server angelegt werden.

Funktionsergebnis	
0	kein Fehler
-1	ungültiges Handle
-2	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)
-3	Fehlermeldung vom FTP-Server (z. B. Verzeichnis bereits vorhanden)

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
    End_Var;

    ....
    Result := FtpDirCreate(FtpHandle,
                           'DataFiles');
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis im Dateisystem des FTP-Servers das Unterverzeichnis "DataFiles" angelegt werden konnte oder ein Fehler aufgetreten ist. Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung
1	Die Steuerung schickt zum FTP-Server das Kommando, dass das Verzeichnis "DataFiles" angelegt werden soll.
2	<table><tr><th>Wenn ...</th><th>... Dann ...</th></tr><tr><td>keine Fehler aufgetreten sind</td><td>ist das neue Verzeichnis angelegt, die Funktion beendet und der Wert 0 wird zurückgeliefert</td></tr><tr><td>Fehler aufgetreten sind</td><td>wird die Funktion beendet und ein negativer Wert zurückgeliefert</td></tr></table>

Verzeichnisnamen

- Der Funktionsparameter für das Verzeichnis auf dem FTP-Server kann auch den kompletten Pfad - über mehrere Verzeichnisebenen - zu diesem Verzeichnis enthalten, wenn dies vom Dateisystem unterstützt wird.
- Ansonsten muss zunächst mit dem Befehl `FtpDirChange(...)` von einer Verzeichnisebene zur nächsten verzweigt werden, bis das entsprechende Verzeichnis erreicht ist.
- Das Dateisystem eines JetControl unterstützt beide Varianten.

Einschränkungen im Dateisystem eines JetControl

Wird als Funktionsparameter ein Verzeichnis mit Pfad angegeben, müssen alle Verzeichnisse bis zu dem zu Erstellenden vorhanden sein. Ein rekursives Anlegen von mehreren Verzeichnissen wird nicht unterstützt.

Beispiel:

```
Result := FtpDirCreate(FtpHandle,
                       '/DataFiles/TextFiles/Release');
```

Damit das Unterverzeichnis "Release" angelegt werden kann, muss das Unterverzeichnis "TextFiles" im Verzeichnis "DataFiles" bereits vorhanden sein.

Verwandte Themen:

- **Verzeichnis auf dem FTP-Server löschen** auf Seite 29

Verzeichnis auf dem FTP-Server löschen

Einleitung

Mit dieser Funktion wird ein Verzeichnis auf dem FTP-Server gelöscht.

Funktionsdeklaration

```
Function FtpDirRemove(FtpConnection:Handle,
                      Const Ref ServerDir: String):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung
ServerDir	Verzeichnisname	Name des Verzeichnisses, welches gelöscht werden soll

Funktionsergebnis

Bei einem negativen Funktionsergebnis ist ein Fehler aufgetreten. Bei einem Funktionsergebnis von "0" konnte das Verzeichnis auf dem FTP-Server gelöscht werden.

Funktionsergebnis	
0	kein Fehler
-1	ungültiges Handle
-2	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)
-3	Fehlermeldung vom FTP-Server (z. B. Verzeichnis nicht vorhanden)

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
    End_Var;

    ....
    Result := FtpDirRemove(FtpHandle,
                           'DataFiles');
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis im Dateisystem des FTP-Servers das Unterverzeichnis "DataFiles" gelöscht werden konnte oder ein Fehler aufgetreten ist. Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung
1	Die Steuerung schickt zum FTP-Server das Kommando, dass das Verzeichnis "DataFiles" gelöscht werden soll
2	<table><tr><th>Wenn ...</th><th>... Dann ...</th></tr><tr><td>keine Fehler aufgetreten sind</td><td>ist das Verzeichnis gelöscht, die Funktion beendet und der Wert 0 wird zurückgeliefert</td></tr><tr><td>Fehler aufgetreten sind</td><td>wird die Funktion beendet und ein negativer Wert zurückgeliefert</td></tr></table>

Verzeichnisnamen

- Der Funktionsparameter für das Verzeichnis auf dem FTP-Server kann auch den kompletten Pfad - über mehrere Verzeichnisebenen - zu diesem Verzeichnis enthalten, wenn dies vom Dateisystem unterstützt wird.
- Ansonsten muss zunächst mit dem Befehl `FtpDirChange(...)` von einer Verzeichnisebene zur nächsten verzweigt werden, bis das entsprechende Verzeichnis erreicht ist.
- Das Dateisystem eines JetControl unterstützt beide Varianten.

Verwandte Themen:

- **Verzeichnis auf dem FTP-Server anlegen** auf Seite 27

Aktuelles Verzeichnis auf dem FTP-Server ermitteln

Einleitung

Mit dieser Funktion wird das aktuell eingestellte Verzeichnis auf dem FTP-Server ermittelt.

Funktionsdeklaration

```
Function FtpDirPrint(FtpConnection:Handle,
                     Ref str: String):Int;
```

Funktionsparameter

Beschreibung der Funktionsparameter:

Parameter	Wert	Bemerkung
FtpConnection	Handle auf die Verbindung	Rückgabewert beim Öffnen der Verbindung
str	Adresse eines Strings	Hier wird das aktuelle Verzeichnis mit Pfadangaben abgelegt

Funktionsergebnis

Bei einem negativen Funktionsergebnis ist ein Fehler aufgetreten. Bei einem Funktionsergebnis von "0" konnte das aktuelle Verzeichnis auf dem FTP-Server ermittelt werden.

Funktionsergebnis	
0	kein Fehler
-1	ungültiges Handle
-3	Kommunikationsfehler (z. B. FTP-Server antwortet nicht)
-4	Fehlermeldung vom FTP-Server
-5	ungültige Rückmeldung vom Server

Verwenden der Funktion

So kann die Funktion verwendet werden und das Funktionsergebnis einer Variablen zur weiteren Auswertung zugewiesen werden:

```
Task FtpClientTask Autorun
    Var
        FtpHandle: Handle;
        Result:    Int;
        Directory: String;
    End_Var;

    ....
    Result := FtpDirPrint(FtpHandle,
                          Directory);
    ....
End_Task;
```

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis im Dateisystem des FTP-Servers das aktuelle Verzeichnis ermittelt werden konnte oder ein Fehler aufgetreten ist. Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung
1	Die Steuerung schickt zum FTP-Server das Kommando, dass das aktuelle Verzeichnis ermittelt werden soll.
2	<table><tr><th>Wenn ...</th><th>... Dann ...</th></tr><tr><td>keine Fehler aufgetreten sind</td><td>enthält die Variable den kompletten Pfad des aktuellen Verzeichnisses, die Funktion wird beendet und der Wert 0 wird zurückgeliefert</td></tr><tr><td>Fehler aufgetreten sind</td><td>wird die Funktion beendet und ein negativer Wert zurückgeliefert</td></tr></table>

Verwandte Themen:

- **Verzeichnis auf dem FTP-Server wechseln** auf Seite 25

2.2.2 Register

Einleitung

Dieses Kapitel beschreibt die Register der Steuerung, in denen Statusinformationen des FTP-Client ausgelesen werden können. Die Register können zu Debug- oder Diagnosezwecken verwendet werden. Weitere Funktionen, wie z. B. Verbindung öffnen oder schließen, können hierüber nicht ausgelöst werden.

Inhalt

Thema	Seite
Registernummern	34
Registerbeschreibung	35

Registernummern

Einleitung

Die Daten jeweils einer Verbindung werden in den Registern eines zusammenhängenden Registerblocks angezeigt. Zwei weitere Register zeigen den Status des aktuell vom Anwenderprogramm ausgeführten Befehls an. Die Basisregisternummer dieser Register ist steuerungsabhängig.

Registernummern

Steuerung	Basisregisternummer	Registernummern
JC-24x		
JM-D203-JC24x		
JC-340, JC-350, JC-360, JC-940MC	320000	320000 ... 350101

Registernummer ermitteln

In diesem Kapitel sind jeweils nur die letzten 3 Ziffern der Registernummer angegeben. Zur Berechnung der tatsächlich verwendeten Registernummer ist die Basisregisternummer der jeweiligen Steuerung hinzuzuaddieren.

Registerübersicht

Übersicht über die Modulregister des FTP-Client:

Register	Beschreibung
MR 000	Anzahl offener Verbindungen
MR 002	Timeout in Sekunden
MR 003	Portnummer des FTP-Servers
MR 004	Index in Verbindungstabelle
MR 005	Handle der Verbindung
MR 006	FTP-Server IP-Adresse
MR 007	FTP-Server Portnummer
MR 008	FTP-Client IP-Adresse
MR 009	FTP-Client Portnummer
MR 100	Anwendertask Zustand
MR 101	Anwendertask ID

Registerbeschreibung

Einleitung

Die geöffneten Verbindungen werden vom Betriebssystem der Steuerung in einer Liste verwaltet. Mit Hilfe der Modulregister MR 004 oder MR 005 werden die Verbindungsdaten einer Verbindung in die Register MR 006 bis MR 009 kopiert.

MR 000

Anzahl geöffneter Verbindungen

Der Wert in diesem Register zeigt an, wie viele Verbindungen zur Zeit geöffnet sind.

Modulregister-Eigenschaften		
Werte lesen	0 ... 2.147.483.647	Anzahl Verbindungen

MR 002

Timeout

In diesem Register wird der Timeout des FTP-Client beim Zugriff auf den FTP-Server eingestellt.

Modulregister-Eigenschaften		
Werte	0 ... 2.147.483.647	in Sekunden
Wert nach Reset	20	

MR 003

FTP-Server Portnummer

In diesem Register wird Nummer des IP-Ports des FTP-Servers eingestellt.

Modulregister-Eigenschaften	
Werte	0 ... 65.535
Wert nach Reset	21

MR 004

Index in Verbindungstabelle

In dieses Register wird der Index der Verbindungstabelle eingetragen. Wenn eine Verbindung an dem eingetragenen Index vorhanden ist, werden anschließend das Handle der Verbindung in Modulregister MR 005 und die Verbindungsdaten in den Modulregistern MR 006 bis MR 009 lesbar.

Modulregister-Eigenschaften	
Werte	0 ... [MR 000] - 1

Wert nach Reset	-1

MR 005

Handle der Verbindung

In dieses Register wird das Handle der Verbindung eingetragen. Wenn eine Verbindung mit diesem Handle vorhanden ist, werden anschließend der Index der Verbindungstabelle in Modulregister MR 004 und die Verbindungsdaten in den Modulregistern MR 006 bis MR 009 lesbar.

Modulregister-Eigenschaften	
Werte	0 ... 2.147.483.647

MR 006

FTP-Server IP-Adresse

Der Wert in diesem Register zeigt die IP-Adresse des FTP-Servers an.

Modulregister-Eigenschaften	
Zugriff	lesen
Wird wirksam	wenn MR 004 >= 0

MR 007

FTP-Server Portnummer

Der Wert in diesem Register zeigt die Portnummer des FTP-Servers an.

Modulregister-Eigenschaften	
Zugriff	lesen
Wird wirksam	wenn MR 004 >= 0

MR 008

FTP-Client IP-Adresse

Der Wert in diesem Register zeigt die IP-Adresse des FTP-Clients an.

Modulregister-Eigenschaften	
Zugriff	lesen
Wird wirksam	wenn MR 004 >= 0

MR 009

FTP-Client Portnummer

Der Wert in diesem Register zeigt die Portnummer des FTP-Clients an.

Modulregister-Eigenschaften		
Zugriff	lesen	
Wird wirksam	wenn MR 004 >= 0	

MR 100

Status der FTP-Client-Bearbeitung

Die Ausführung des FTP-Client Zugriffs kann hier mitverfolgt werden.

Modulregister-Eigenschaften		
Werte	0	Es wird kein Zugriff ausgeführt
	1	Übergabe der Parameter an den FTP-Client der Steuerung
	2	Kommunikation mit dem FTP-Server wird ausgeführt
	3	Der Zugriff wurde abgeschlossen
Zugriff	lesen	

MR 101

Task-ID

Die ID des Tasks, der gerade eine Funktion des FTP-Client ausführt, kann hier gelesen werden.

Modulregister-Eigenschaften		
Werte	0 ... 99	Task-ID
	255	kein Task führt gerade eine FTP-Funktion aus
Wert nach Reset	255	
Zugriff	lesen	

3 Automatisches Kopieren von Steuerungsdaten

Einleitung

Dieses Kapitel beschreibt die AutoCopy-Funktion, die es ermöglicht, Dateien innerhalb der Steuerung und/oder zwischen der Steuerung und einem FTP-Server zu kopieren. Hierzu ist eine Kommandodatei zu erstellen, die zusammen mit den Daten auf der SD-Karte abgelegt ist. Die Kommandodatei wird beim Booten von der Steuerung automatisch abgearbeitet.

Funktionen im lokalen Dateisystem

Folgende Funktionen können ausgeführt werden:

- Register und Merker in eine Datei abspeichern
- Register und Merker aus einer Datei einspielen
- Verzeichnisse anlegen
- Verzeichnisse löschen
- Dateien kopieren
- Dateien löschen

Funktionen im Dateisystem des FTP-Servers

Folgende Funktionen können ausgeführt werden:

- Dateien vom FTP-Server kopieren
- Dateien zum FTP-Server kopieren
- Dateien löschen
- Verzeichnis wechseln
- Verzeichnis anlegen
- Verzeichnis löschen

Anwendungsbereiche

Anwendungsbereiche sind hauptsächlich dort, wo Fernwartung nicht möglich ist, kein PC vor Ort vorhanden ist oder der Anlagenbediener nicht in der Lage ist (oder nicht in die Lage versetzt werden soll) Änderungen an der Anlage vorzunehmen:

- Änderung des Anwenderprogramms
- Änderung der Anwenderdaten
- Änderung der Steuerungskonfiguration
- Betriebssystem-Update (Steuerung, Module am Systembus, Netzwerkteilnehmer)
- Duplizieren eines Steuerungssystems

Voraussetzungen

Folgende Voraussetzungen werden in diesem Kapitel getroffen:

- der Programmierer verfügt über Kenntnisse des Dateisystems der Steuerung JC-340
- der Programmierer verfügt über Grundkenntnisse der Anwendung von FTP
- bei einem JC-340 ist die Option 'SD-Karte' vorhanden

Bezeichnung

Vollständiger Name bezeichnet in dieser Beschreibung den Namen der Datei oder des Verzeichnisses mit komplettem Pfad.

Inhalt

Thema	Seite
Funktionsweise	40
Die Datei 'autocopy.ini'	45
Protokolldatei	58
Datendateien	60

3.1 Funktionsweise

Einleitung

Dieses Kapitel beschreibt wie die AutoCopy-Funktion gestartet wird und wie sie vom JC-340 ausgeführt wird.

Inhalt

Thema	Seite
AutoCopy-Funktion starten	41
AutoCopy-Funktion ausführen	42
Betriebsart AutoCopy beenden	44

AutoCopy-Funktion starten

Einleitung

Die AutoCopy-Funktion kann nur während der Bootphase der Steuerung ausgeführt werden.

Voraussetzungen

Die Kommandodatei wurde erstellt und auf der SD-Karte abgelegt.

	Wert	Bemerkung
Dateiname	autocopy.ini	alles Kleinbuchstaben
Verzeichnis	/SD/	Stammverzeichnis der SD-Karte

AutoCopy-Funktion starten

Führen Sie folgende Schritte aus, um die AutoCopy-Funktion zu starten:

Schritt	Vorgehen
1	Schalten Sie die Steuerung aus
2	Stecken Sie die SD-Karte vollständig in den Steckplatz der SD-Karte
3	Stellen Sie den Schalter in Stellung 'LOAD'
4	Schalten Sie die Steuerung ein

Ergebnis: Die Steuerung bootet in der Betriebsart AutoCopy.

AutoCopy-Funktion ausführen

Einleitung

Während der Bootphase in der Betriebsart AutoCopy führt die Steuerung die Kommandos in der Kommandodatei aus.

Einschränkungen

In der Betriebsart AutoCopy gelten folgende Einschränkungen der Funktion der Steuerung:

- Das Anwenderprogramm wird nicht ausgeführt
- Mit der Steuerung kann nicht kommuniziert werden

AutoCopy-Funktion ausführen

Das Betriebssystem der Steuerung bearbeitet die AutoCopy-Funktion in folgenden Stufen:

Stufe	Beschreibung
1	Die Steuerung lädt die Datei '/SD/autocopy.ini' von der SD-Karte
2	Die Steuerung liest die Werte aus der Sektion [OPTIONS]
3	Die Steuerung liest das Kommando und die Kommandoparameter aus der Sektion [COMMAND_1], bearbeitet es und schreibt das Ergebnis gegebenenfalls in die Protokolldatei
4 .. n	Die Steuerung bearbeitet die weiteren Kommandos in aufsteigender Reihenfolge, bis zur Anzahl aus der Sektion [OPTIONS]
n+1	Die Steuerung ermittelt die Ergebnisstatistik aller Kommandos und schreibt sie in die Protokolldatei

LEDs der Steuerung in der Betriebsart AutoCopy

Die Status-LEDs des Betriebssystems haben in der Bootphase der Steuerung folgende Zustände:

Stufe 1

R	E	D1	D2	SD	Zustand
4Hz	4Hz	4Hz	4Hz	OFF	Reset

Stufe 2

R	E	D1	D2	SD	Zustand
1Hz	OFF	OFF	ON	OFF	der Bootloader lädt und überprüft das Betriebssystem

Stufe 3

R	E	D1	D2	SD	Zustand
1Hz	OFF	OFF	OFF	OFF	das Betriebssystem liest den DIP-Schalter der Backplane und prüft das Vorhandensein des Ethernet-Switches

<table>
<tr><th>Stufe</th><th>Beschreibung</th></tr>
<tr><td>4</td><td>
<table>
<tr><th>R</th><th>E</th><th>D1</th><th>D2</th><th>SD</th><th>Zustand</th></tr>
<tr><td>
</td><td>
</td><td>
</td><td>
</td><td>
</td><td>das Betriebssystem initialisiert die Echtzeituhr und das Dateisystem</td></tr>
</table>
</td></tr>
<tr><td>5</td><td>
<table>
<tr><th>R</th><th>E</th><th>D1</th><th>D2</th><th>SD</th><th>Zustand</th></tr>
<tr><td>
</td><td>
</td><td>
</td><td>
</td><td></td><td>das Betriebssystem initialisiert die Module am JX3- und JX2-Systembus und die SD-Karte</td></tr>
</table>
</td></tr>
<tr><td>6</td><td>
<table>
<tr><th>R</th><th>E</th><th>D1</th><th>D2</th><th>SD</th><th>Zustand</th></tr>
<tr><td>
</td><td>
</td><td>
</td><td>
</td><td></td><td>die Kommandodatei der AutoCopy-Funktion wird bearbeitet</td></tr>
</table>
</td></tr>
<tr><td>7a</td><td>
<table>
<tr><th>R</th><th>E</th><th>D1</th><th>D2</th><th>SD</th><th>Zustand</th></tr>
<tr><td>
</td><td>
</td><td>
</td><td>
</td><td>
</td><td>die AutoCopy-Funktion ist beendet; es sind keine Fehler aufgetreten</td></tr>
</table>
</td></tr>
<tr><td>7b</td><td>
<table>
<tr><th>R</th><th>E</th><th>D1</th><th>D2</th><th>SD</th><th>Zustand</th></tr>
<tr><td>
</td><td>
</td><td>
</td><td>
</td><td>
</td><td>die AutoCopy-Funktion ist beendet; es sind Fehler aufgetreten</td></tr>
</table>
</td></tr>
</table>

Betriebsart AutoCopy beenden

Einleitung

Die Betriebsart AutoCopy kann nur durch Booten der Steuerung verlassen werden.

Betriebsart AutoCopy beenden

Führen Sie, nachdem die AutoCopy-Funktion beendet ist, folgende Schritte aus, um die Betriebsart AutoCopy zu verlassen:

Schritt	Vorgehen
1	Schalten Sie die Steuerung aus
2	Sie können jetzt die SD-Karte entfernen (nicht notwendig)
3	Stellen Sie den Schalter in Stellung 'RUN' oder 'STOP'
4	Schalten Sie die Steuerung ein

Ergebnis: Die Steuerung bootet neu.

3.2 Die Datei 'autocopy.ini'

Einleitung

Dieses Kapitel beschreibt den Aufbau der Datei 'autocopy.ini' und die verfügbaren Kommandos.

Aufbau der Datei

Die Kommandodatei der AutoCopy-Funktion ist eine Textdatei, deren Einträge in Sektionen gruppiert sind.

- In diesen Sektionen werden Werte gesetzt, mit denen die AutoCopy-Funktion arbeitet
- Leerzeilen können beliebig eingefügt werden
- Kommentarzeilen werden mit '!', '#' oder ';' eingeleitet

Sektionen

Die Kommandodatei enthält zwei Sektionstypen:

- In der Sektion [OPTIONS] werden grundsätzliche Einstellungen gemacht; sie ist nur ein Mal vorhanden
- In den Sektionen [COMMAND_#] werden die auszuführenden Kommandos angegeben; die Anzahl der Kommandosektionen ist auf 128 beschränkt

Inhalt

Thema	Seite
Die Sektion [OPTIONS]	46
Die Kommandosektionen	47
Beispiel für eine Kommandodatei	55

Die Sektion [OPTIONS]

Einleitung

In dieser Sektion werden grundsätzliche Einstellungen der AutoCopy-Funktion gemacht. Sie ist nur einmal, vorzugsweise am Beginn der Datei, vorhanden.

Beispiel

```
[OPTIONS]
CommandCount = 14
LogFile      = /SD/autocopy.log
LogAppend    = 1
```

Elemente der Sektion

Die Sektion besteht aus den folgenden Teilen:

CommandCount	
im Beispiel	14
Funktion	Anzahl der nachfolgenden Kommandosektionen
gültige Werte	> = 0
nicht gültige Werte	< 0
bei ungültigem Wert oder nicht vorhandenem Eintrag	0
LogFile	
im Beispiel	/SD/autocopy.log
Funktion	vollständiger Name der Protokolldatei
gültige Werte	■ alle gültigen Dateinamen ■ Verzeichnis ist vorhanden
nicht gültige Werte	■ ungültiger Dateinamen ■ nicht vorhandenes Verzeichnis
bei ungültigem Wert oder nicht vorhandenem Eintrag	Es wird keine Protokolldatei erzeugt.
LogAppend	
im Beispiel	1
Funktion	legt fest, ob eine neue Protokolldatei erstellt oder an eine vorhandene angefügt wird;
gültige Werte	■ 0 = gegebenenfalls bestehende Datei löschen und neue Datei erstellen ■ 1 = an bestehende Datei anfügen; wenn keine Datei vorhanden ist, wird ebenfalls eine neue Protokolldatei erstellt
nicht gültige Werte	■ < 0 ■ > 1
bei ungültigem Wert oder nicht vorhandenem Eintrag	Es wird die Protokolldatei neu erstellt.

Die Kommandosektionen

Einleitung

In diesen Sektionen werden die Befehle angegeben, die von der AutoCopy-Funktion des Gerätes JC-340 ausgeführt werden.

Beispiel

```
[COMMAND_1]
Command        = DirCreate
Path           = /Homepage
ErrorAsWarning = 1

[COMMAND_2]
Command        = FileCopy
Source         = /SD/Index.htm
Destination    = /Homepage/index.htm

[COMMAND_3]
Command        = FtpConnect
ServerAddr     = 192.168.123.45
UserName       = admin
Password       = admin
```

Name der Sektionen

Die Namen der Sektionen werden gebildet durch die Zeichenkette `COMMAND_` gefolgt von einer Zahl, die zwischen 1 und dem Wert des Eintrags `Command-Count` aus der Sektion `[OPTIONS]` liegt.

Bearbeitung der Kommandos

Die AutoCopy-Funktion bearbeitet die Kommandos in der Reihenfolge der Sektionsnamen.

- Beginnend mit dem Kommando unter Sektion `[COMMAND_1]`
- Endend mit dem Kommando unter der Sektion mit dem Wert des Eintrags `CommandCount` aus der Sektion `[OPTIONS]`
- Jede Kommandosektion kann nur ein Kommando enthalten, so dass für jedes Kommando eine eigene Sektion anzulegen ist

Fehlerbehandlung

Tritt bei der Bearbeitung eines Kommandos ein Fehler auf, erfolgt ein entsprechender Eintrag in der Protokolldatei. Für jedes Kommando kann eingestellt werden, ob der Fehler als `Error` oder als `Warning` in der Protokolldatei eingetragen werden soll. Die Einstellung erfolgt über den optionalen Parameter `ErrorAsWarning`:

ErrorAsWarning	Eintrag in Protokolldatei
Parameter nicht vorhanden	Error
ErrorAsWarning = 0	Error
ErrorAsWarning = 1	Warning

Dateinamen

- Der Funktionsparameter für die lokale Datei kann auch den Pfad zu dieser Datei enthalten (z. B. `"/Data/TestFiles/LocalTestFile.txt"`).
- Der Funktionsparameter für die Datei auf dem FTP-Server kann auch den Pfad zu dieser Datei enthalten, wenn dies vom Dateisystem unterstützt wird. Ansonsten muss zunächst mit dem Befehl `FtpDirChange(...)` das entsprechende Verzeichnis eingestellt werden.
- Das Dateisystem eines <Produktnamen> unterstützt beide Varianten.

Verfügbare Kommandos im lokalen Dateisystem

Die folgenden Kommandos für Zugriffe auf das lokale Dateisystem sind verfügbar:

Command = DirCreate	
Funktion	Anlegen eines Unterverzeichnisses
Parametername	Path
Parameterwert	vollständiger Name des Verzeichnisses
gültige Werte	■ alle gültigen Verzeichnisnamen ■ übergeordnete Verzeichnisse sind vorhanden
nicht gültige Werte	■ ungültiger Verzeichnisnamen ■ nicht vorhandenes übergeordnetes Verzeichnis ■ Name eines bereits vorhandenen Verzeichnisses
bei ungültigem Wert	Das Verzeichnis wird nicht angelegt und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_1]`
	`Command = DirCreate`
	`Path = /sub1`
	
	`[COMMAND_2]`
	`Command = DirCreate`
	`Path = /sub1/sub2`

Command = DirRemove	
Funktion	Löschen eines Unterverzeichnisses
Parametername	Path
Parameterwert	vollständiger Name des Verzeichnisses
gültige Werte	■ alle gültigen Verzeichnisnamen ■ das Verzeichnis ist leer
nicht gültige Werte	■ ungültiger Verzeichnisnamen ■ Verzeichnis ist nicht leer
bei ungültigem Wert	Das Verzeichnis wird nicht gelöscht und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_8]`
	`Command = DirRemove`
	`Path = /sub1/sub2`

Command = FileCopy	
Funktion	Kopieren einer Datei
Parametername 1	Source

Parameterwert 1	vollständiger Name der Quelldatei
Parametername 2	Destination
Parameterwert 2	vollständiger Name der Zieldatei
gültige Werte	■ alle gültigen Dateinamen ■ das Zielverzeichnis ist vorhanden
nicht gültige Werte	■ ungültiger Dateinamen ■ nicht vorhandene Quelldatei ■ nicht vorhandenes Zielverzeichnis
bei ungültigem Wert	Die Datei wird nicht kopiert und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_1]`
	`Command = FileCopy`
	`Source = /SD/OS/JC-340_1.04.0.03.os`
	`Destination = /System/OS/op_system.os`
	
	`[COMMAND_2]`
	`Command = FileCopy`
	`Source = /SD/Manual.pdf`
	`Destination = /sub1/Manual.pdf`

Command = FileRemove

Funktion	Löschen einer Datei
Parametername	Path
Parameterwert	vollständiger Name der Datei
gültige Werte	alle gültigen Dateinamen
nicht gültige Werte	ungültiger Dateinamen
bei ungültigem Wert	Die Datei wird nicht gelöscht und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_5]`
	`Command = FileRemove`
	`Path = /sub1/Manual.pdf`

Command = DaFileRead

Funktion	Übertragen von Registerwerten und Merkerzuständen von einer Datendatei zum JC-340
Parametername	DaFile
Parameterwert	vollständiger Name der Datendatei
gültige Werte	alle gültigen Dateinamen für Datendateien
nicht gültige Werte	■ ungültiger Dateinamen ■ nicht vorhandene Datendatei
bei ungültigem Wert	Die Daten werden nicht an das Gerät JC-340 übertragen und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_12]`
	`Command = DaFileRead`
	`DaFile = /SD/Data/MyTestData.da`

Command = DaFileWrite	
Funktion	Abspeichern von Registerwerten und Merkerzuständen in einer Datendatei
Parametername 1	DaFile
Parameterwert 1	vollständiger Name der Datei
gültige Werte	■ alle gültigen Dateinamen für Datendateien ■ das Zielverzeichnis ist vorhanden
nicht gültige Werte	■ ungültiger Dateinamen ■ nicht vorhandenes Zielverzeichnis
bei ungültigem Wert	Die Datei wird nicht angelegt und der Fehler in die Protokolldatei eingetragen
Parametername 2	Append
Parameterwert 2	legt fest, ob eine neue Datendatei erstellt oder an eine vorhandene angefügt wird
gültige Werte	■ 0 = gegebenenfalls bestehende Datei löschen und neue Datei erstellen ■ 1 = an bestehende Datei anfügen; wenn keine Datei vorhanden ist, wird ebenfalls eine neue Datendatei erstellt
nicht gültige Werte	■ < 0 ■ > 1
bei ungültigem Wert	Die Datendatei wird neu erstellt
Parametername 3	Type
Parameterwert 3	legt fest, ob Register oder Merker abgespeichert werden sollen
gültige Werte	■ Register ■ Flag
nicht gültige Werte	Werte ungleich 'Register' oder 'Flag'
bei ungültigem Wert	Die Datei wird nicht angelegt und der Fehler in die Protokolldatei eingetragen
Parametername 4	First
Parameterwert 4	Nummer des ersten Registers / Merkers
gültige Werte	alle gültigen Nummern aus dem Speicherbereich des jeweiligen JC-340
nicht gültige Werte	ungültige Nummern
bei ungültigem Wert	Die Datei wird nicht angelegt und der Fehler in die Protokolldatei eingetragen
Parametername 5	Last
Parameterwert 5	Nummer des letzten Registers / Merkers
gültige Werte	alle gültigen Nummern aus dem Speicherbereich des jeweiligen JC-340, die größer oder gleich sind als der Wert bei 'First'
nicht gültige Werte	■ ungültige Nummern ■ Nummern kleiner als 'First'
bei ungültigem Wert	Es wird nur ein Wert (First) abgespeichert

Beispiel

```
[COMMAND_11]
Command     = DaFileWrite
DaFile      = /SD/MyTestData2.da
Append      = 0
Type        = Register
First       = 1000000
Last        = 1000000

[COMMAND_12]
Command     = DaFileWrite
DaFile      = /SD/MyTestData2.da
Append      = 1
Type        = Flag
First       = 10
Last        = 20

[COMMAND_13]
Command     = DaFileWrite
DaFile      = /SD/MyTestData2.da
Append      = 1
Type        = Register
First       = 1000001
Last        = 1000999
```

Verfügbare Kommandos für FTP-Zugriffe

Die folgenden Kommandos für Zugriffe über das Netzwerk mittels FTP sind verfügbar:

Command = FtpConnect

Funktion	Verbindung zu einem FTP-Server herstellen
Parametername 1	ServerAddr
Parameterwert 1	IP-Adresse oder Name des FTP-Servers
gültige Werte	■ erreichbare IP-Adresse ■ Name, der über DNS auflösbar ist
nicht gültige Werte	■ nicht erreichbare IP-Adresse ■ nicht auflösbarer Name
Parametername 2	UserName
Parameterwert 2	User-Name für die Anmeldung am FTP-Server
Parametername 3	Password
Parameterwert 3	Passwort für die Anmeldung am FTP-Server
bei ungültigen Werten	Verbindung wird nicht hergestellt und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_1]`
	`Command = FtpConnect`
	`ServerAddr = 192.168.123.45`
	`UserName = admin`
	`Password = admin`

Bemerkung	Es kann immer nur eine Verbindung zu einem FTP-Server bestehen. Soll eine Verbindung zu einem anderen FTP-Server hergestellt werden, so schließt das Gerät zuvor die bestehende Verbindung.

Command = FtpFileRead	
Funktion	Datei vom FTP-Server in das lokale Dateisystem kopieren
Parametername 1	ServerFile
Parameterwert 1	vollständiger Name der Quelldatei im FTP-Server
Parametername 2	ClientFile
Parameterwert 2	vollständiger Name der Zieldatei im lokalen Dateisystem
gültige Werte	■ alle gültigen Dateinamen ■ das Zielverzeichnis ist vorhanden
nicht gültige Werte	■ ungültiger Dateinamen ■ nicht vorhandene Quelldatei ■ nicht vorhandenes Zielverzeichnis
bei ungültigem Wert	Die Datei wird nicht kopiert und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_8]`
	`Command = FtpFileRead`
	`ServerFile = /app/cantest/cantest.es3`
	`ClientFile = /SD/cantest3.es`

Command = FtpFileWrite	
Funktion	Datei vom lokalen Dateisystem in das des FTP-Servers kopieren
Parametername 1	ServerFile
Parameterwert 1	vollständiger Name der Zieldatei im FTP-Server
Parametername 2	ClientFile
Parameterwert 2	vollständiger Name der Quelldatei im lokalen Dateisystem
gültige Werte	■ alle gültigen Dateinamen ■ das Zielverzeichnis ist vorhanden
nicht gültige Werte	■ ungültiger Dateinamen ■ nicht vorhandene Quelldatei ■ nicht vorhandenes Zielverzeichnis
bei ungültigem Wert	Die Datei wird nicht kopiert und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_5]`
	`Command = FtpFileWrite`
	`ServerFile = /System/OS/op_system.os`
	`ClientFile = /SD/OS/JC-340_1.09.0.00.os`

Command = FtpFileRemove	
Funktion	Löschen einer Datei im FTP-Server
Parametername	ServerFile

Parameterwert	vollständiger Name der Datei
gültige Werte	alle gültigen Dateinamen
nicht gültige Werte	ungültiger Dateinamen
bei ungültigem Wert	Die Datei wird nicht gelöscht und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_9]`
	`Command = FtpFileRemove`
	`ServerFile = /sub1/Manual.pdf`

Command = FtpDirChange	
Funktion	Arbeitsverzeichnis im FTP-Server wechseln
Parametername	ServerDir
Parameterwert	vollständiger Name des Verzeichnisses
gültige Werte	alle gültigen Verzeichnisnamen
nicht gültige Werte	ungültiger Verzeichnisnamen
bei ungültigem Wert	Wird das Verzeichnis nicht gewechselt und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_12]`
	`Command = FtpDirChange`
	`ServerDir = /Data/MyTestData`

Commando = FtpDirCreate	
Funktion	Anlegen eines Unterverzeichnisses im FTP-Server
Parametername	ServerDir
Parameterwert	vollständiger Name des Verzeichnisses
gültige Werte	■ alle gültigen Verzeichnisnamen ■ übergeordnete Verzeichnisse sind vorhanden
nicht gültige Werte	■ ungültiger Verzeichnisnamen ■ nicht vorhandenes übergeordnetes Verzeichnis ■ Name eines bereits vorhandenen Verzeichnisses
bei ungültigem Wert	Wird das Verzeichnis nicht angelegt und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_6]`
	`Command = FtpDirCreate`
	`ServerDir = /Data/MyTestData`

Einschränkung	Wird als Funktionsparameter ein Verzeichnis mit Pfad angegeben, müssen alle Verzeichnisse bis zu dem zu Erstellenden vorhanden sein. Ein rekursives Anlegen von mehreren Verzeichnissen wird nicht unterstützt.

Commando = FtpDirRemove	
Funktion	Löschen eines Unterverzeichnisses im FTP-Server
Parametername	ServerDir
Parameterwert	vollständiger Name des Verzeichnisses
gültige Werte	■ alle gültigen Verzeichnisnamen ■ das Verzeichnis ist leer

nicht gültige Werte	■ ungültiger Verzeichnisnamen ■ Verzeichnis ist nicht leer
bei ungültigem Wert	Wird das Verzeichnis nicht gelöscht und der Fehler in die Protokolldatei eingetragen

Beispiel	`[COMMAND_8]`
	`Command = FtpDirRemove`
	`ServerDir = /Data/MyTestData`

Beispiel für eine Kommandodatei

Aufgabe

Bei einer bestehenden Anlage, die von einem JetControl 340 mit verschiedenen JX3-Modulen gesteuert wird, soll eine Funktionserweiterung durchgeführt werden. Hierfür müssen folgende Änderungen gemacht werden:

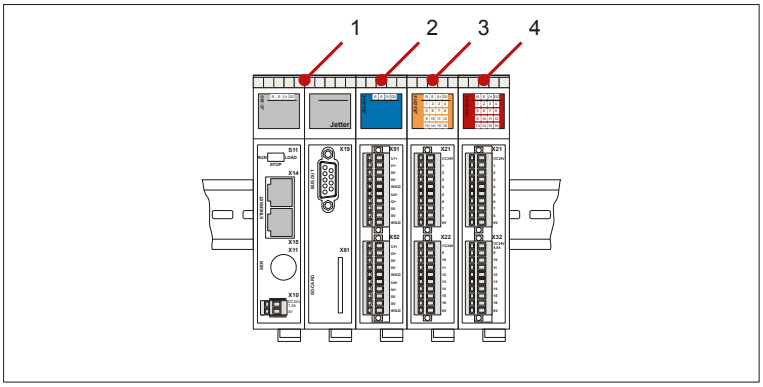
- Betriebssystem-Update für die Steuerung
- Betriebssystem-Update für ein Analogausgangsmodul
- neues Anwenderprogramm
- neue Werte für einige Register

Lösung

Die benötigten Dateien werden auf eine SD-Karte kopiert und eine Kommandodatei für die AutoCopy-Funktion erstellt. Diese SD-Karte wird mit einer kurzen Bedienungsanleitung zum Anlagenbediener geschickt, der sie nach dem Update wieder zurückschickt.

Beispiel-Konfiguration

Diese Beispiel basiert auf folgender Konfiguration:



Nummer	Teil	Funktion
1	JC-340	Steuerung
2	JX3-AO4	Analogausgangsmodul I/O-Modulnummer 2
3	JX3-DI16	Digitales Eingangsmodul
4	JX3-DIO16	Digitales Ausgangsmodul

Inhalt der SD-Karte

Die nachfolgende Grafik zeigt die Verzeichnisstruktur und die Dateien auf der SD-Karte aus Sicht der Steuerung vor Ausführen der AutoCopy-Funktion:



Nach der Ausführung ist zusätzlich die Protokolldatei 'autocopy.log' vorhanden.

Kommandodatei

```
[OPTIONS]
CommandCount = 7
LogFile      = /SD/autocopy.log
LogAppend    = 0

# update operating system of controller
[COMMAND_1]
Command     = FileCopy
Source      = /SD/OS/JC-340_1.04.0.00.os
Destination = /System/OS/op_system.os

# update operating system of JX3-AO4 module
[COMMAND_2]
Command     = FileCopy
Source      = /SD/OS/JX3-AO4_1.01.0.00.os
Destination = /System/JX3-Module02/OS/system.os

# create user program directories
# probably already present - but to be sure ...
[COMMAND_3]
Command        = DirCreate
Path           = /app
ErrorAsWarning = 1

[COMMAND_4]
Command     = DirCreate
Path        = /app/userprogtest

# copy user program start file
[COMMAND_5]
```

```
Command     = FileCopy
Source      = /SD/UserProgs/start.ini
Destination = /app/start.ini

# copy user program
[COMMAND_6]
Command     = FileCopy
Source      = /SD/UserProgs/userprogtest.es3
Destination = /app/userprogtest/userprogtest.es3

# set registers and flags
[COMMAND_7]
Command     = DaFileRead
DaFile      = /SD/UserData/MyTestData.da
```

3.3 Protokolldatei

Einleitung

Dieses Kapitel beschreibt Aufbau und Inhalt der Protokolldatei, in welcher das Ergebnis der jeweiligen Kommandos mitgeschrieben wird.

Inhalt

Thema	Seite
Dateiinhalt	59

Dateiinhalt

Einleitung

Die Protokolldatei ist eine Textdatei. In der Kommandodatei wird festgelegt, ob eine Protokolldatei angelegt werden soll. Und ob diese jeweils neu angelegt wird oder die Einträge an eine bestehende Protokolldatei angehängt werden.

Beispiel

```
JetControl AutoCopy log file 07.11.2008 09:14:09

 1: Ok      - FileCopy   /SD/OS/JC-340_1.04.0.00.os
                         /System/OS/op_system.os (345740 byte)
 2: Ok      - FileCopy   /SD/OS/JX3-AO4_1.01.0.00.os
                         /System/JX3-Module02/OS/system.os
                         (16832 byte)
 3: Warning - DirCreate  /app
 4: Ok      - DirCreate  /app/userprogtest
 5: Ok      - FileCopy   /SD/UserProgs/start.ini
                         /app/start.ini (63 byte)
 6: Ok      - FileCopy   /SD/UserProgs/userprogtest.es3
                         /app/userprogtest/userprogtest.es3
                         (169 byte)
 7: Error   - DaFileRead /SD/UserData/MyTestData.da

Command statistics:
  Total  : 7
  Ok     : 5
  Warning: 1
  Error  : 1
```

Beschreibung

Die Protokolldatei besteht - beim Anfügen an eine bestehende Protokolldatei für jede Ausführung der AutoCopy-Funktion - aus drei Elementen:

- Die Kopfzeile enthält Datum und Uhrzeit.
- Der anschließende Block gibt Auskunft über die ausgeführten Kommandos.
- Abschließend wird eine kurze Statistik über die Kommandobearbeitung angezeigt.

Im obigen Beispiel wird beim Versuch, das Verzeichnis '/app' anzulegen, ein Fehler erzeugt (der als Warnung eingetragen wird), da das Verzeichnis bereits vorhanden ist. Beim Lesen der DA-Datei tritt ebenfalls ein Fehler auf, der als solcher in die Protokolldatei geschrieben wird.

3.4 Datendateien

Einleitung

Dieses Kapitel beschreibt die Datendateien, in denen die Werte von Registern und Merkern abgelegt werden.

Inhalt

Thema	Seite
Dateiformat	61

Dateiformat

Format

Die Datei ist folgendermaßen aufgebaut:

- reine Textdatei
- jeder Eintrag muss in einer eigenen Zeile stehen
- jede Zeile muss mit Wagenrücklauf / Zeilenvorschub (carriage return / line feed) abgeschlossen sein
- Kommentarzeilen werden mit einem Semikolon (;) eingeleitet
- als erster Eintrag in der Datei muss als Kennung 'SD1001' stehen

Datenzeilen

Eine Datenzeile besteht aus den folgenden Teilen:

- Variablenkennung am Beginn der Zeile
- durch Leerzeichen oder Tabulator getrennt folgt die Variablennummer
- durch Leerzeichen oder Tabulator getrennt folgt der Variablenwert

Variablenkennung	Variablentyp
FS	Merker
RS	Ganzzahlregister
QS	Fließkommazahlregister

Beispiel

```
SD1001
; Data File - Jetter AG
;
; Register 1000000 ... 1000005
RS    1000000    12345
RS    1000001    2
RS    1000002    -1062729008
RS    1000003    502
RS    1000004    50
RS    1000005    3
QS    1009000    3.14
;
; Flag 10 ... 13
FS    10    0
FS    11    1
FS    12    1
FS    13    0
```

3.5 Systemkommandoregister

Einleitung

Über das Systemkommandoregister lässt sich das Verhalten des Betriebssystems der Steuerung JC-340 beeinflussen.

Vorgehensweise

Um ein Systemkommando auszulösen, gehen Sie wie folgt vor:

Schritt	Vorgehen
1	Beschreiben Sie das Systempasswortregister mit dem korrekten Wert.
2	Beschreiben Sie das Systemkommandoregister mit dem gewünschten Kommandowert.
3	Überprüfen Sie gegebenenfalls die entsprechenden Bits im Systemstatusregister.
⇨	**Ergebnis:** Die Steuerung setzt das Systempasswortregister und das Systemkommandoregister auf 0.

Inhalt

Thema	Seite
Beschreibung der Systemkommandoregister	63
Beschreibung der Systemkommandos	65

Beschreibung der Systemkommandoregister

Registerübersicht

In dieser Anleitung werden die folgenden Register verwendet:

Register	Beschreibung
R 202960	Systempasswortregister
R 202961	Systemkommandoregister
R 202962	Systemstatusregister

R 202960

Systempasswortregister

In dieses Register ist das Systempasswort 1112502132 (0x424F6F74) einzutragen, damit das Systemkommandoregister beschrieben werden kann. Nach Beschreiben des Systemkommandoregisters wird das Systempasswortregister von der Steuerung auf 0 gesetzt.

Register-Eigenschaften	
Wert	1112502132 (0x424F6F74)

R 202961

Systemkommandoregister

In dieses Register werden die Systemkommandos eingetragen. Beim Beschreiben werden die Kommandos von der Steuerung ausgeführt. Nachdem die Steuerung das Kommando ausgeführt hat, setzt sie das Systemkommandoregisters auf 0.

Kommandos	
102	**Steuerung neu starten**
122	**Warte auf Kommunikation aus**
123	**Warte auf Kommunikation ein**
160	**Task-Wechsel bei I/O-Zugriff aus**
161	**Task-Wechsel bei I/O-Zugriff ein**

Register-Eigenschaften	
Zugriff	Das Systempasswortregister enthält das korrekte Passwort

R 202962

Systemstatusregister

Über das Systemstatusregister können Systemzustände ausgewertet werden.

Bedeutung der Bits		
Bit 0	**Task-Wechsel bei I/O-Zugriff**	
	0 =	Bei einem I/O-Zugriff erfolgt kein Task-Wechsel im Anwenderprogramm
	1 =	Bei einem I/O-Zugriff erfolgt ein Task-Wechsel im Anwenderprogramm
Bit 1	**Warte auf Kommunikation**	
	0 =	Die Steuerung wartet für kurze Zeit auf Kommunikationsanfragen
	1 =	Die Steuerung wartet nicht auf Kommunikationsanfragen

Register-Eigenschaften	
Zugriff	lesen

Beschreibung der Systemkommandos

Systemkommando 102 — **Steuerung neu starten**

Auswirkung:

Die Steuerung führt einen Neustart durch. Die Wirkung entspricht dem Aus-/Einschalten der Stromversorgung.

Anwendung:

Das Kommando kann z. B. verwendet werden, wenn Änderungen an Systemregistern oder Systemdateien vorgenommen wurden, die nur beim Booten der Steuerung aktiv werden.

Systemkommando 122 — **Warte auf Kommunikation Aus**

Auswirkung:

Die Steuerung kommuniziert erst mit externen Kommunikationsteilnehmern, wenn konkrete Anfragen von ihnen bereits vorliegen.

Vorteil:

Das Anwenderprogramm wird schneller abgearbeitet.

Nachteil:

Externe Kommunikationsteilnehmer müssen durchschnittlich etwas länger warten, bis sie von der Steuerung Antwort bekommen.

Systemkommando 123 — **Warte auf Kommunikation Ein**

Auswirkung:

Die Steuerung wartet zyklisch 1 bis 2 Millisekunden und prüft, ob externe Kommunikationsteilnehmer mit ihr kommunizieren möchten.

Vorteil:

Externe Kommunikationsteilnehmer erhalten von der Steuerung durchschnittlich schneller Antwort.

Nachteil:

Die Bearbeitung des Anwenderprogramms verlangsamt sich geringfügig.

Systemkommando 160 — **Task-Wechsel bei I/O-Zugriff Aus**

Auswirkung:

Während eines Zugriffs auf Module am JX2- oder JX3-Systembus werden die anderen Tasks des Anwenderprogramms nicht bearbeitet.

Vorteil:

Die I/O-Zugriffe werden so schnell wie möglich ausgeführt.

Nachteil:

Da manche I/O-Zugriffe deutlich langsamer sind als der Zugriff auf interne Variablen, kann die Reaktionszeit der anderen Tasks ansteigen.

Systemkommando 161

Task-Wechsel bei I/O-Zugriff Ein

Auswirkung:

Während eines Zugriffs auf Module am JX2- oder JX3-Systembus werden die anderen Tasks des Anwenderprogramms bearbeitet.

Vorteil:

Die relativ lange Ausführungszeit mancher I/O-Zugriffe beeinflusst nicht die Reaktionszeit der anderen Tasks.

Nachteil:

Die Ausführungszeit mancher I/O-Zugriffe wird durch die Bearbeitungszeit der anderen Task beeinflusst.

4 Beseitigte Software-Bugs

Einleitung

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

Inhalt

Thema	Seite
Achsbefehle Pos, ActualPos, AxArr	68
Taskrestart bricht einen Delay-Befehl nicht ab	69
Absturz bei fehlender Bibliothek	70
NetCopyVarToReg funktioniert mit Float-Variablen nicht richtig	71
Serielle Schnittstelle wird nach dem Booten nicht richtig initialisiert	72
Speichern von Fließkommazahlregistern in Datendateien funktioniert nicht	73
Eingangsänderung vom JX2-ID8 werden nicht erkannt	74
Schnelle Zustandsänderung von digitalen Eingängen am JX2-Systembus	75

Achsbefehle Pos, ActualPos, AxArr

Fehlerbild

Die Achsbefehle `Pos, ActualPos und AxArr` zur Programmierung der JX2-Systembus-Module JX2-SV1, JX2-SM2 und JX2-SM1D funktionieren nicht.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340/350	< 1.09.0.00
	JC-360	< 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Abhilfe / Workaround

Programmieren Sie die Achsmodule durch Zugriffe auf ihre Register.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340/350	1.09.0.00
	JC-360	1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Achsnummern

Die Achsnummer für JX2-Slave-Module am JX2-Systembus des JC-3xx besteht aus folgenden Elementen:

2	0	0	0	0	0	0	m	a

Element	Bedeutung	Wertebereich
m	Slave-Modulnummer des Moduls	2 ... 9
a	Nummer der Achse auf diesem Modul	1 ... 9

Taskrestart bricht einen Delay-Befehl nicht ab

Fehlerbild

Wenn ein Task, der gerade an einem `Delay()` steht, von einem anderen Task mit `Taskrestart` zurückgesetzt wird, so läuft der zurückgesetzte Task erst nach Ablauf der Delay-Zeit an.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340/350 JC-360	< 1.09.0.00 < 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Abhilfe / Workaround

Eine Abhilfe ist in den betroffenen Versionen / Revisionen nicht möglich.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340/350 JC-360	1.09.0.00 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Absturz bei fehlender Bibliothek

Fehlerbild

Wenn ein Anwenderprogramm mit einer Bibliothek erstellt wurde, diese Bibliothek beim Programmstart aber nicht auf der Steuerung vorhanden ist, so stürzt die Steuerung ab.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340/350 JC-360	< 1.09.0.00 < 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Abhilfe / Workaround

Führen Sie zur Umgehung des Fehlers folgende Schritte aus:

Schritt	Vorgehen
1	Schalten Sie die Steuerung aus.
2	Stellen Sie den Schalter in Stellung STOP.
3	Schalten Sie die Steuerung wieder ein.
4	Übertragen Sie das Anwenderprogramm aus JetSym in die Steuerung.
5	Stellen Sie den Schalter wieder in Stellung RUN und starten Sie die Steuerung neu.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340/350 JC-360	1.09.0.00 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

NetCopyVarToReg funktioniert mit Float-Variablen nicht richtig

Fehlerbild

Wenn von einem Netzwerkteilnehmer mit der Funktion `NetCopyVarToReg()` die Inhalte von Float-Variablen auf Float-Register kopiert werden, so werden in die Register falsche Werte geschrieben.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340/350	< 1.09.0.00
	JC-360	< 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Abhilfe / Workaround

Legen Sie im Programm des Netzwerkteilnehmers, der die Kopieraktion ausführt, die Werte in Float-Register und verwenden Sie dann die Funktion `NetCopyRegToReg()`.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340/350	1.09.0.00
	JC-360	1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Serielle Schnittstelle wird nach dem Booten nicht richtig initialisiert

Fehlerbild

Am Beginn des Anwenderprogramms soll die serielle Schnittstelle durch Beschreiben ihrer Register initialisiert werden. Die geschriebenen Werte werden jedoch von der Steuerung nicht übernommen, so dass die Schnittstelle in der Applikation nicht funktioniert. Das Problem tritt nur nach dem Booten der Steuerung auf. Nach einem Programm-Neustart (z. B. aus JetSym) wird die Schnittstelle korrekt initialisiert.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340/350	< 1.09.0.00
	JC-360	< 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Abhilfe / Workaround

Fügen Sie im Anwenderprogramm eine kurze Wartezeit (z. B. `Delay(T#10 ms)`) ein, bevor erstmalig nach dem Programmstart auf die Register der seriellen Schnittstelle zugegriffen wird.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340/350	1.09.0.00
	JC-360	1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Speichern von Fließkommazahlregistern in Datendateien funktioniert nicht

Fehlerbild

Beim Speichern von Fließkomma-Registern in eine Datendatei werden falsche Werte in die Datei geschrieben. Dies betrifft sowohl die AutoCopy-Funktion als auch die `Systemfunction` 90 und 91 zum Schreiben einer Datendatei im Anwenderprogramm.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340/350 AutoCopy Systemfunction JC-360 AutoCopy Systemfunction	 < 1.09.0.00 = 1.08.0.16 < 1.09.0.00 = 1.08.0.11
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Abhilfe / Workaround

Eine Abhilfe ist in den betroffenen Versionen / Revisionen nicht möglich.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340/350 JC-360	1.09.0.00 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Eingangsänderung vom JX2-ID8 werden nicht erkannt

Fehlerbild

Das Lesen eines digitalen Eingangs eines JX2-ID8 Moduls im Anwenderprogramm liefert nicht immer den tatsächlichen Zustand am Modul. Dies betrifft sowohl die IN-Befehle als auch den Zugriff über die Register der zusammengefassten Eingänge.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340/350	< 1.09.0.00
	JC-360(MC)	< 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	keine CANopen®-Module am JX2-Systembus	

Abhilfe / Workaround

Führen Sie zur Umgehung des Fehlers folgende Schritte aus:

Schritt	Vorgehen
1	Setzen Sie in R 200002077 Bit 6 = TRUE. Dadurch sucht die Steuerung JC-340 nicht nach angeschlossenen CANopen®-Modulen.
2	Starten Sie die Steuerung neu.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340/350	1.09.0.00
	JC-360(MC)	1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Schnelle Zustandsänderung von digitalen Eingängen am JX2-Systembus

Fehlerbild

Ändert sich der Zustand der digitalen Eingänge der Module am JX2-Systembus im Abstand von 1 ... 5 ms, dann kann dies zu einer Fehlermeldung an der Steuerung JC-340 führen.

Im Fehlerfall tritt folgendes Fehlerbild auf:

- Es leuchtet die rote E-LED an der Steuerung JC-340.
- In R 210004 wird Bit 2 = 1 "Fehler JX2-Systembus" gesetzt.
- Das Lesen von Registern 200xxxxxx vom JX2-Systembus führt kurzfristig zu falschen Ergebnissen.

Betroffene Versionen / Revisionen

Der Fehler tritt in folgenden Versionen / Revisionen auf:

Betriebssystem-Version	JC-340	< 1.09.0.00
	JC-350	< 1.09.0.00
	JC-360	< 1.09.0.00
	JC-360MC	< 1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	

Abhilfe / Workaround

Die Abhilfe besteht darin, den Abstand zwischen zwei Zustandsänderungen der digitalen Eingänge am JX2-Systembus zu vergrößern. Dies wird sich aber in der Praxis kaum umsetzen lassen.

Fehlerbehebung

Der Fehler ist ab den folgenden Versionen / Revisionen behoben:

Betriebssystem-Version	JC-340	1.09.0.00
	JC-350	1.09.0.00
	JC-360	1.09.0.00
	JC-360MC	1.09.0.00
Hardware-Revision	nicht relevant	
Konfiguration oder Betriebsart	nicht relevant	