BUCHER automation

Themenhandbuch



60881371_03

Dateisystem

Dieses Dokument wurde von der Bucher Automation AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt. Änderungen und technische Weiterentwicklungen an unseren Produkten werden nicht automatisch in einem überarbeiteten Dokument zur Verfügung gestellt.
Die Bucher Automation AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Bucher Automation AG
Thomas-Alva-Edison-Ring 10
71672 Marbach am Neckar, Deutschland
T +49 7141 2550-0
info@bucherautomation.com

Technischer Support
T +49 7141 2550-444
support@bucherautomation.com

Vertrieb
T +49 7141 2550-663
sales@bucherautomation.com

www.bucherautomation.com

Originaldokument

Dokumentenversion	1.40
Ausgabedatum	01.12.2025

Inhaltsverzeichnis

1 Dateisystem ... 6
1.1 Verzeichnisse ... 6
1.1.1 Verzeichnisse bei Steuerungen ... 7
1.1.2 Verzeichnisse bei Bediengeräten ... 7
1.2 Eigenschaften ... 9
1.2.1 Eigenschaften der Flash-Disk ... 10
1.2.2 Eigenschaften der SD-Karte ... 10
1.2.3 Eigenschaften des USB-Sticks ... 11
1.3 Benutzerverwaltung ... 11
1.3.1 Begriffsdefinition ... 12
1.3.2 Benutzer verwalten ... 12
1.3.3 Auslieferungszustand/Vordefinierte Benutzer und Schlüssel ... 14
1.3.4 Schloss anbringen ... 14
1.3.5 Namen einrichten für Schlüssel/Schlösser ... 15
1.3.6 Erweiterte Benutzerverwaltung für JetControl JC-4xx und JC-9xx ... 17
1.4 Belegung der Flash-Disk einsehen ... 21
1.4.1 Belegung der Flash-Disk ... 21
1.5 Formatieren und Prüfen ... 23
1.5.1 Flash-Disk formatieren ... 24
1.5.2 SD-Karte formatieren ... 24
1.5.3 USB-Stick formatieren ... 24
1.5.4 SD-Karte prüfen ... 25
1.5.5 USB-Stick überprüfen ... 25
2 E-Mail-Funktion ... 26
2.1 Konfigurieren der E-Mail-Funktion ... 26
2.1.1 Aufbau der Konfigurationsdatei ... 26
2.1.2 Sektion [SMTP] ... 27
2.1.3 Sektion [POP3] ... 29
2.1.4 Sektion [DEFAULT] ... 31
2.1.5 Beispiele für eine Konfigurationsdatei ... 31
2.2 E-Mail erstellen ... 31
2.2.1 Name der E-Mail-Vorlagendatei ... 32
2.2.2 Struktur der E-Mail-Vorlagendatei ... 32
2.2.3 Einfügen von Echtzeit-Steuerungswerten ... 34
2.3 E-Mail versenden ... 34
2.3.1 STX-Funktion EMailSend() ... 34

2.4 Register ... 35
2.4.1 Registerübersicht ... 35
2.4.2 Registerbeschreibung ... 35
3 FTP-Server ... 39
3.1 Anmeldung ... 39
3.2 FTPS Server ... 40
3.2.1 Benutzer-Verwaltung ... 40
3.2.2 Anmelden als "admin" ... 40
4 HTTP-Server ... 41
4.1 Server Side Includes ... 41
4.1.1 Erster Eintrag in der HTML-Datei ... 42
4.1.2 Einfügen von Echtzeit-Steuerungswerten ... 42
4.1.3 Beispiel einer HTML-Seite ... 46
5 FTP-Client ... 48
5.1 Programmierung ... 48
5.1.1 FTP-Client initialisieren ... 48
5.1.2 Verbindung zum FTP-Server öffnen ... 49
5.1.3 Verbindung schließen ... 50
5.1.4 Datei lesen ... 51
5.1.5 Datei schreiben ... 52
5.1.6 Datei löschen ... 53
5.1.7 Verzeichnis wechseln ... 54
5.1.8 Verzeichnis anlegen ... 55
5.1.9 Verzeichnis löschen ... 57
5.1.10 Aktuelles Verzeichnis ermitteln ... 58
5.2 Register ... 59
5.2.1 Registernummern ... 59
5.2.2 Registerbeschreibung ... 59
6 AutoCopy - Automatisches Kopieren von Steuerungsdaten ... 62
6.1 Funktionsweise ... 63
6.1.1 AutoCopy-Funktion starten ... 63
6.1.2 AutoCopy-Funktion ausführen ... 64
6.1.3 AutoCopy-Funktion beenden ... 67
6.2 autocopy.ini - Aufbau ... 67
6.2.1 Die Sektion [OPTIONS] ... 67
6.2.2 Die Kommandosektionen ... 69
6.2.3 Beispiel für eine Kommandodatei ... 77
6.3 Protokolldatei ... 79

6.3.1 Dateiinhalt ... 79
6.4 Datendateien ... 80
6.4.1 Dateiformat ... 80
7 Anwendungsprogramm speichern und laden ... 82
7.1 Standardablage des Anwendungsprogramms ... 82
7.2 Anwendungsprogramm auf der SD-Karte oder dem USB-Stick speichern ... 82
7.3 Anwendungsprogramm laden ... 84

1 Dateisystem

Dieses Kapitel beschreibt das Dateisystem. Das Dateisystem ermöglicht den Zugriff auf die Dateien der internen Flash-Disk, der SD-Karte und eines USB-Sticks.

Wenn zu lösende Probleme auftreten, ist es hilfreich, das Dateisystem zu verstehen.

Hinweis

Arbeiten Sie vorsichtig im Umgang mit dem Dateisystem, insbesondere mit den Systemdateien. Fehlerhafte Systemdateien können zu einem Gerät führen, das nicht mehr bootet.

Nicht alle Dateien sind lesbar, schreibbar oder löschbar. Dieses Verhalten ist normal. Einige dieser Dateien sind virtuelle Dateien, z. B. Firmware Images, oder geschützte Dateien, z. B. EDS-Dateien.

Aufteilung

Das Dateisystem unterscheidet zwischen folgenden Dateien:

– Vom Betriebssystem verwendete Systemverzeichnisse/-dateien
– Vom Anwender verwendete Dateien

1.1 Verzeichnisse

Systemverzeichnisse

Systemverzeichnisse sind nicht löschbar. Sie sind auch nach dem Formatieren noch vorhanden.

Steuerungen

Verzeichnisnamen durch einen Schrägstrich "/" und nicht durch einen umgekehrten Schrägstrich "\" trennen.

Verzeichniss	Beschreibung
/System	– Systemkonfiguration – Systeminformationen
/SD	– Root-Verzeichnis der SD-Karte
/USBx	– Root-Verzeichnis des USB-Sticks x

Tab. 1: Systemverzeichnisse

Bediengeräte

Verzeichnisnamen durch einen umgekehrten Schrägstrich "\" und nicht durch einen Schrägstrich "/" trennen.

Verzeichnis	Beschreibung
\System	– Systemkonfiguration – Systeminformationen – Begrüßungsbild (Boot-Image) – Screenshot
\SD	– Root-Verzeichnis der SD-Karte
\USB	– Root-Verzeichnis des USB-Sticks
\App	– Speicherbereich für die Applikation
\Data	– Speicherbereich für die Daten

Verzeichnis	Beschreibung
\Windows	– Systemverzeichnis von Windows CE
\	– RAM-Disk

Tab. 2: Bediengeräte

1.1.1 Verzeichnisse bei Steuerungen

Verzeichnis /SD

/SD

Wenn Sie eine SD-Karte in den SD-Karten-Steckplatz stecken, wird das Verzeichnis /SD dynamisch erstellt. Sonst ist dieses Verzeichnis nicht sichtbar.

Verzeichnis /System

/System

Dieses Verzeichnis enthält systemrelevante Dateien, wie den Kernel, die Co-Prozessor-Firmware, Konfigurationen, EDS usw.

Verzeichnis /USBx

/USBx

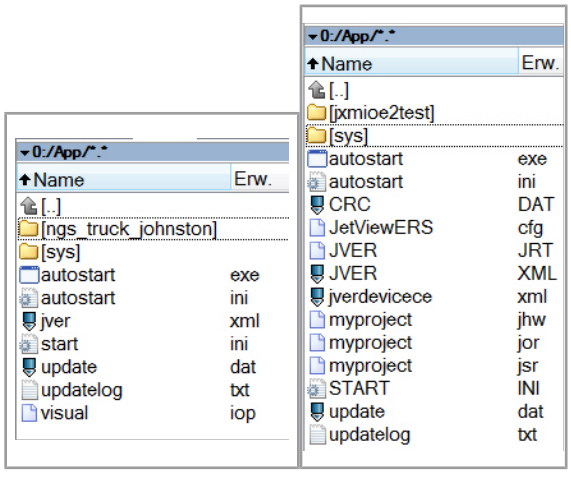
Wenn Sie einen USB-Stick in den USB-Anschluss x stecken, wird das Verzeichnis /USBx dynamisch erstellt. Sonst ist dieses Verzeichnis nicht sichtbar.

1.1.2 Verzeichnisse bei Bediengeräten

Verzeichnis \App

\App

Dieses Verzeichnis enthält die Anwendung und die Visualisierung. In diesem Beispiel ist die STX-Anwendung in den Ordnern ngs_truck_johnston (links) oder jxmioe2test (rechts) gespeichert.



Links basieren die Daten auf der älteren S-Plattform. Die Visualisierung ist in einer .iop-Datei gespeichert, hier ist es die Datei visual.iop.

Rechts basieren die Daten auf der neuen CE-Plattform. Da gibt es keine .iop-Datei. Stattdessen speichert JetViewSoft mehrere Visualisierungsdateien ab.

HINWEIS



Kopieren in falsche Ordner verursacht Verzögerungen beim Hochfahren

► Kopieren Sie alle Anwendungs- und Visualisierungsdateien in den Ordner **App** und **nicht** in den Ordner **Data.** Siehe Verzeichnis \Data.

start.ini

Der Inhalt dieser Textdatei legt fest, welche Anwendung gestartet wird.

\App\sys

Dieses Verzeichnis enthält die Interpreter der STX-Programmiersprache und der Visualisierung. Hier nichts verändern!

autostart.exe

Diese Anwendung lässt ein Betriebssystemupdate zu. Hier nichts verändern!

Außerdem startet die Datei das Gerät mitsamt der Visualisierung.

updatelog.txt

Das ist eine Log-Datei, die während des Betriebssystemupdates geschrieben wird.

Verzeichnis \Data

\Data

Dieses Verzeichnis ist der große Speicher des Bediengeräts. Sie können einige Parameter- oder Konfigurationsdateien hier speichern.

HINWEIS! Auf der Datenpartition können größere Datenmengen abgelegt werden. Um einen zügigen Systemstart sicherzustellen, wird diese Partition ggf. zeitversetzt nach dem Start der STX-Applikation eingebunden. Die Applikation kann nicht auf dieser Partition abgelegt werden.

Verzeichnis \SD

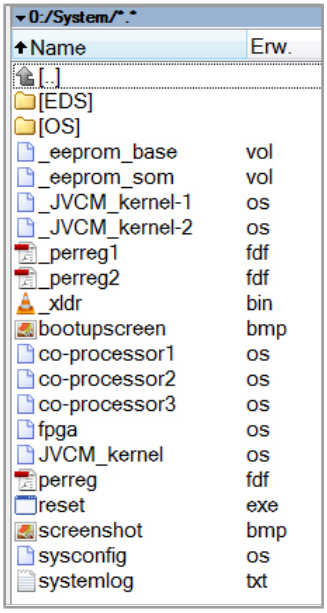
\SD

Wenn Sie eine SD-Karte in den SD-Karten-Steckplatz des Bediengeräts stecken, wird das Verzeichnis \SD dynamisch erstellt. Sonst ist dieses Verzeichnis nicht sichtbar.

Verzeichnis \System

\System

Dieses Verzeichnis enthält systemrelevante Dateien, wie den Kernel, die Co-Prozessor-Firmware, Konfigurationen, EDS usw.



bootupscreen.bmp

Das ist eine 16-Bit-Bmp-Datei (r5, g6, b5), die als Bild direkt nach dem Einschalten sichtbar ist.

Sie können eine eigene Bilddatei erstellen und diese Datei ersetzen.

co-processor1

Diese virtuelle Datei enthält die Firmware eines versteckten Co-Prozessors, der die meisten Interaktionen (Schaltflächen, Summer, Hintergrundlicht usw.) mit dem Anwender steuert.

reset.exe

Wenn Sie diese Datei löschen, führt das Bediengerät sofort danach einen Neustart aus. Diese Funktion können Sie z. B. in Batch-Dateien anwenden, die nach der Abarbeitung einen automatischen Neustart erfordern.

Verzeichnis \USB

\USB

Wenn Sie einen USB-Stick in das Bediengerät stecken, wird das Verzeichnis \USB dynamisch erstellt. Sonst ist dieses Verzeichnis nicht sichtbar.

Verzeichnis \Windows

\Windows

Dieses Verzeichnis enthält Dateien von Windows CE. **Hier nichts verändern!**

1.2 Eigenschaften

Dieses Kapitel beschreibt die Eigenschaften des Dateisystems. Unterschieden wird zwischen Steuerungen und Bediengeräten und dort dann wieder zwischen der internen Flash-Disk, der SD-Karte und des USB-Sticks.

Steuerungen: Allgemeine Eigenschaften

Für die interne Flash-Disk und die SD-Karte gelten folgende Eigenschaften:

- Maximal 8 Dateien gleichzeitig öffnen.
- Verzeichnisnamen durch einen Schrägstrich "/" und nicht durch einen umgekehrten Schrägstrich "\" trennen.
- Wenn die Steuerung eine Datei erstellt, erhält die Datei Datum und Uhrzeit der Steuerung.
- Datum, Uhrzeit und Dateigröße sind nicht bei allen Systemdateien verfügbar.

Bediengeräte: Allgemeine Eigenschaften

Für die interne Flash-Disk, die SD-Karte und den USB-Stick gelten folgende Eigenschaften:

– Maximal 8 Dateien gleichzeitig öffnen.
– Ausschließlich Kleinschreibung für Verzeichnis- und Dateinamen verwenden.
– Wenn das Gerät eine Datei erstellt, erhält die Datei Datum und Uhrzeit des Geräts.
– Datum, Uhrzeit und Dateigröße sind nicht bei allen Systemdateien verfügbar.

1.2.1 Eigenschaften der Flash-Disk

Die Speichergröße ist geräteabhängig.

Steuerungen: Eigenschaften

Die interne Flash-Disk hat folgende weitere Eigenschaften:

– 7 Verzeichnisebenen und 1 Dateiebene sind zulässig.
– Groß- und Kleinschreibung wird unterschieden.
– Verzeichnis- und Dateinamen mit jeweils maximal 63 Zeichen Länge sind möglich.
– Für Verzeichnis- und Dateinamen sind alle Zeichen außer "/" und ".." erlaubt.
– Benutzer-/Zugriffsverwaltung mit maximal 31 Schlössern für maximal 33 Benutzer.

Bediengeräte: Eigenschaften

Die interne Flash-Disk hat folgende weitere Eigenschaften:

– 7 Verzeichnisebenen und 1 Dateiebene sind zulässig.
– Eine Unterscheidung zwischen Groß- und Kleinschreibung gibt es nicht.
– Für die Zeichenlänge von Verzeichnis- und Dateinamen sind gesamt 63 Zeichen zulässig.
– Für Verzeichnis- und Dateinamen sind alle Zeichen außer "\" und ".." erlaubt.
– Die Ordner App und Data liegen auf der Flash-Disk.
– Es gibt keine Benutzer-/Zugriffsverwaltung.

1.2.2 Eigenschaften der SD-Karte

Die Speichergröße ist geräteabhängig.

Steuerungen: Eigenschaften

Die SD-Karte hat folgende weitere Eigenschaften:

– Die SD-Karte muss FAT 16 kompatibel sein.
– Groß- und Kleinschreibung wird nicht unterschieden.
– Verzeichnis- und Dateinamen mit jeweils maximal 63 Zeichen sind möglich.
– In Verzeichnis- und Dateinamen sind nicht erlaubt: "/", "\", ":", "*", "?", """, "<", ">" und "|"
– Die Anzahl der Verzeichnisebenen ist abhängig von der Formatierung.
– Es gibt keine Benutzer-/Zugriffsverwaltung.

Die Bucher Automation AG übernimmt nur für die als Option erhältlichen Karten die Gewähr.

Bediengeräte: Eigenschaften

Die SD-Karte hat folgende weitere Eigenschaften:

– Die SD-Karte muss FAT 16 oder FAT 32 kompatibel sein.

Um Daten auf die SD-Karte zu speichern, beachten Sie Folgendes:

– Für die Zeichenlänge von Verzeichnis- und Dateinamen sind gesamt 260 Zeichen zulässig.
– Eine Unterscheidung zwischen Groß- und Kleinschreibung gibt es nicht.

– In Verzeichnis- und Dateinamen sind nicht erlaubt: "/", "\", ":", "*", "?", """, "<", ">" und "|"

Die Bucher Automation AG übernimmt nur für die als Option erhältlichen SD-Karten die Gewähr.

1.2.3 Eigenschaften des USB-Sticks

Die Speichergröße ist geräteabhängig.

Steuerungen: Eigenschaften

Der USB-Stick hat folgende weitere Eigenschaften:

– Der USB-Stick muss FAT 12, FAT 16 oder FAT 32 kompatibel sein.
– Groß- und Kleinschreibung wird nicht unterschieden.
– Verzeichnis- und Dateinamen mit jeweils maximal 63 Zeichen sind möglich.
– In Verzeichnis- und Dateinamen sind nicht erlaubt: "/", "\", ":", "*", "?", """, "<", ">" und "|"
– Die Anzahl der Verzeichnisebenen ist abhängig von der Formatierung.
– Es gibt keine Benutzer-/Zugriffsverwaltung.

Bediengeräte: Eigenschaften

Der USB-Stick hat folgende weitere Eigenschaften:

– Der USB-Stick muss FAT 12, FAT 16 oder FAT 32 kompatibel sein.
– Für die Zeichenlänge von Verzeichnis- und Dateinamen sind gesamt 260 Zeichen zulässig.
– Eine Unterscheidung zwischen Groß- und Kleinschreibung gibt es nicht.
– In Verzeichnis- und Dateinamen sind nicht erlaubt: "/", "\", ":", "*", "?", """, "<", ">" und "|"
– Es gibt keine Benutzer-/Zugriffsverwaltung.

Die Bucher Automation AG übernimmt nur für die als Option erhältlichen USB-Sticks die Gewähr.

1.3 Benutzerverwaltung

Im Dateisystem für die interne Flash-Disk besteht die Möglichkeit, Zugriffsrechte (Schlösser) für Verzeichnisse zu definieren und Benutzer einzurichten.

Für jeden Standardbenutzer sind die Zugriffsrechte (Schlüssel) einstellbar.

Auf Verzeichnisse und Dateien, für die der Benutzer nicht den benötigten Schlüssel besitzt, ist der Zugriff nicht möglich. Diese Verzeichnisse und Dateien erscheinen bei einer FTP-Verbindung auch nicht.

Voraussetzungen

Für die Benutzerverwaltung sind Administratorrechte erforderlich.

Eigenschaften

Die Benutzerverwaltung hat folgende Eigenschaften:

Eigenschaft	Maximalwert
Anzahl Benutzer	33
Anzahl vordefinierter Benutzer	2 (admin und system)
Länge Benutzername	31 alphanumerische Zeichen
Länge Passwort	31 alphanumerische Zeichen
Anzahl Leseschlüssel	31
Anzahl Schreibschlüssel	31
Anzahl vordefinierter Schlüssel	2

Tab. 3: Eigenschaften

Dateien

Die Einstellungen der Benutzerverwaltung geschieht in drei Dateien im Ordner **System**:

Datei	Funktion
flashdisklock.ini	Schlösser an Verzeichnisse anbringen
keys.ini	Namen für Schlösser/Schlüssel vergeben
users.ini	Benutzer verwalten

Tab. 4: Dateien

Die Dateien sind immer vorhanden. Sie sind nicht löschbar, sondern sind nur änderbar oder überschreibbar.

Einschränkungen

Beachten Sie bitte die folgenden Einschränkungen:

– Die Benutzerverwaltung ist nur für die interne Flash-Disk möglich. Sie ist nicht auf die SD-Karte und den USB-Stick anwendbar.

– Wenn eine Datei der Benutzerverwaltung übertragen wurde, ist ihr Inhalt sofort lesbar. Die Einstellungen werden erst nach dem nächsten Boot-Vorgang gültig.

1.3.1 Begriffsdefinition

Begriff	Beschreibung
Key	Schlüssel für Verzeichnisse, Dateien oder Berechtigungen
admin	Der Benutzer mit dem Namen admin und admin-Rechten
system	Der Benutzer mit dem Namen system und system-Rechten
XCOM	XCOM ist ein Protokoll für die Kommunikation zwischen JetSym und Jet-Control Steuerungen der Bucher Automation.

Tab. 5: Begriffe

1.3.2 Benutzer verwalten

Die Benutzer des Dateisystems verwalten Sie in der Konfigurationsdatei /System/users.ini.

Voraussetzungen

Wenn Sie Namen für die Schlüssel verwenden wollen, muss das Gerät die Namen zuvor kennen. Richten Sie die Namen deshalb zuerst ein, Einrichten von Namen für Schlüssel/Schlösser.

Benutzer verwalten

Um die Benutzer zu verwalten, führen Sie folgende Schritte aus:

1. Stellen Sie eine FTP-Verbindung zum Gerät her. Melden Sie sich hierbei mit Administratorrechten an.
2. Öffnen Sie die Datei /System/users.ini.
3. Ändern Sie die Dateieinträge.
4. Speichern Sie die geänderte Datei wieder auf dem Gerät.
5. Booten Sie das Gerät neu.

⇨ Die geänderten Benutzereinstellungen sind aktiv.

Aufbau der Konfigurationsdatei

Die Konfigurationsdatei ist eine Textdatei, deren Einträge in Sektionen gruppiert sind.

– Für jeden Benutzer wird eine eigene Sektion verwendet.

– In diesen Sektionen werden Werte gesetzt, mit denen das Dateisystem arbeitet.

– Leerzeilen können Sie beliebig einfügen.
– Kommentarzeilen werden mit "!", "#" oder ";" eingeleitet.

Sektionen

Die Namen der Sektionen sind *[USER1]* bis *[USER33]*. Hier legen Sie jeweils Benutzername und Passwort sowie die Lese- und Schreibrechte fest.

Beispiel

```
[USER4]
NAME=TestUser3
PW=testpass
READKEYS=5,openLock2,10,11
WRITEKEYS=openLock2,10,11
SYSKEYS=
```

Eigenschaft	Beschreibung
NAME	
Im Beispiel	TestUser3
Beschreibung	Anmeldename des Benutzers
Gültige Werte	Maximal 31 alphanumerische Zeichen
Bei ungültigem Wert oder nicht vorhandenem Eintrag	Der Benutzer wird nicht angelegt
PW	
Im Beispiel	testpass
Beschreibung	Anmeldepasswort des Benutzers
Gültige Werte	Maximal 31 alphanumerische Zeichen
Bei ungültigem Wert oder nicht vorhandenem Eintrag	Für die Anmeldung des Benutzers ist kein Passwort erforderlich
READKEYS	
Im Beispiel	5,openLock2,10,11
Beschreibung	Schlüssel für Lesezugriffe
Gültige Werte	1 ... 31 (oder entsprechende Namen)
Bei ungültigem Wert oder nicht vorhandenem Eintrag	Der Benutzer erhält keine Leseschlüssel
WRITEKEYS	
Im Beispiel	openLock2,10,11
Beschreibung	Schlüssel für Schreibzugriffe
Gültige Werte	1 ... 31 (oder entsprechende Namen)
Bei ungültigem Wert oder nicht vorhandenem Eintrag	Der Benutzer erhält keine Schreibschlüssel
SYSKEYS	
Beschreibung	Keine Funktion; reserviert für zukünftige Erweiterungen

1.3.3 Auslieferungszustand/Vordefinierte Benutzer und Schlüssel

Im Dateisystem sind zwei Benutzer mit festgelegten Rechten vordefiniert. Diese beiden Benutzer können nicht gelöscht werden. Mit Hilfe der Benutzerverwaltung können Sie für diese Benutzer lediglich ein anderes Passwort festlegen.

Auslieferungszustand

Im Auslieferungszustand der Steuerung hat die Konfigurationsdatei folgenden Inhalt:

```
[USER1]
NAME=admin
PW=admin
READKEYS=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,2
6,27,28,29,30,31
WRITEKEYS=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
26,27,28,29,30,31
SYSKEYS=PlcControl, ProgControl, ProgDebug, ProgProfile, ProgDownload,
OsDownload, VarRead, VarWrite, FileRead, FileWrite
[USER33]
NAME=system
PW=system
READKEYS=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,
27,28,29,30,31
WRITEKEYS=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26
,27,28,29,30,31
SYSKEYS=PlcControl, ProgControl, ProgDebug, ProgProfile, ProgDownload,
OsDownload, VarRead, VarWrite, FileRead, FileWrite
```

Vordefinierte Benutzer

Der Benutzer admin hat alle Schlüssel und kann somit auf alle Verzeichnisse und Dateien lesend und schreibend zugreifen.

Der Benutzer system hat bis auf Schlüssel 1 ebenfalls alle Schlüssel.

Vordefinierte Schlüssel

Von den 31 Schlüsseln haben zwei eine vordefinierte Funktion.

Schloss/Schlüssel	Funktion
1	– Ethernet-Konfiguration – Benutzerverwaltung
2	– Betriebssystemupdate der CPU – Betriebssystemupdate der Peripheriemodule

Tab. 6: Vordefinierte Schlüssel

1.3.4 Schloss anbringen

In der Konfigurationsdatei /System/flashdisklock.ini versehen Sie Verzeichnisse der Flash-Disk mit Schlössern. Nur Benutzer mit dem passenden Schlüssel können Dateien und Unterverzeichnisse in diesen Verzeichnissen lesen oder schreiben (löschen).

Voraussetzungen

Wenn Sie Namen für die Schlösser verwenden wollen, muss das Gerät die Namen zuvor kennen. Richten Sie die Namen deshalb zuerst ein, Namen einrichten für Schlüssel/Schlösser (siehe Seite 22).

Schloss anbringen

Um ein Schloss an ein Verzeichnis anzubringen, führen Sie folgende Schritte aus:

1. Stellen Sie eine FTP-Verbindung zu dem Gerät her; melden Sie sich hierbei mit Administratorrechten an.
2. Öffnen Sie die Datei /System/flashdisklock.ini.
3. Ändern Sie die Dateieinträge.
4. Speichern Sie die geänderte Datei wieder auf dem Gerät.
5. Booten Sie das Gerät neu.

⇨ Das Verzeichnis ist mit einem Schloss versehen.

Aufbau der Konfigurationsdatei

Die Konfigurationsdatei ist eine Textdatei, die eine Sektion enthält.

– In dieser Sektion werden Werte gesetzt, mit denen das Dateisystem arbeitet.
– Jedes Verzeichnis geben Sie mit seiner Schlossnummer auf einer eigenen Zeile an.
– Leerzeilen können Sie beliebig einfügen.
– Kommentarzeilen werden mit "!", "#" oder ";" eingeleitet.

Sektion

Der Name der Sektion ist *[LOCKS].* Hier ordnen Sie die Schlösser den Verzeichnissen nach folgendem Schema zu:

Verzeichnis=Schloss

Beispiel:

```
[LOCKS]
test1=0
test1/sub1=2
test1/sub2=5
test2=userlock2
```

Schlossnummern

Die Schlossnummern haben folgende Eigenschaften:

– Gültige Schlossnummern sind 0 ... 31.
– Schlossnummer 0: Am Verzeichnis ist kein Schloss angebracht. Sie benötigen keine besondere Zugangsberechtigung.
– Sie können Nummern oder zuvor definierte Namen verwenden.

1.3.5 Namen einrichten für Schlüssel/Schlösser

Die Schlüssel/Schlösser sind von 1 ... 31 durchnummeriert. Um einfacher mit Schlüssel/Schloss umzugehen, kann jedem Schlüssel/Schloss ein Name zugeordnet werden. In der Konfigurationsdatei /System/keys.ini richten Sie die Namen ein.

Namen einrichten

Um Namen für Schlüssel/Schlösser einzurichten, führen Sie folgende Schritte aus:

1. Stellen Sie eine FTP-Verbindung zu dem Gerät her; melden Sie sich hierbei mit Administratorrechten an.
2. Öffnen Sie die Datei /System/keys.ini.
3. Ändern Sie die Dateieinträge.
4. Speichern Sie die geänderte Datei wieder auf dem Gerät.
5. Booten Sie das Gerät neu.

⇨ Die Namen stehen jetzt zur Verfügung. Benutzer können die Namen jetzt beim Anbringen von Schlössern und Verwalten von Benutzern verwenden.

Aufbau der Konfigurationsdatei

Die Konfigurationsdatei ist eine Textdatei, die eine Sektion enthält.

– In dieser Sektion werden Werte gesetzt, mit denen das Dateisystem arbeitet.
– Jeder Schlüssel ist mit seinem Namen auf einer eigenen Zeile angegeben.
– Leerzeilen können Sie beliebig einfügen.
– Kommentarzeilen werden mit "!", "#" oder ";" eingeleitet.

Sektion

Der Name der Sektion ist [KEYS]. Hier ordnen Sie die Namen den Schlüsseln/Schlössern nach folgendem Schema zu:

KEYxx=Name

xx: Nummer des Schlüssels (01 ... 31)

Beispiel:

```
[KEYS]
KEY01=Admin
KEY02=System
KEY03=
KEY04=
KEY05=service
...
KEY31=
```

Namen für Schlösser und Schlüssel

Für die Namen gelten folgende Festlegungen:

– Maximal 15 alphanumerische Zeichen
– Schloss und Schlüssel müssen denselben Namen haben

1.3.6 Erweiterte Benutzerverwaltung für JetControl JC-4xx und JC-9xx

Für die Benutzerverwaltung mit erhöhter Sicherheit und Rechten, wurde die bisherige Benutzerverwaltung erweitert.

In diesem Kapitel sind im Wesentlichen nur die Unterschiede zu der zuvor beschriebenen Benutzerverwaltung dokumentiert.

Ab folgenden OS-Versionen gibt es eine erweiterte Benutzerverwaltung für JC-4xx und JC-9xx.

JC-4xx: 1.15.0.08

JC-9xx: 1.23.0.14

Beschreibung der Funktionalität

Allgemein

- Die Benutzer des Dateisystems verwalten Sie in der **Datei /System/users.ini**.
  - Berechtigungen für **XCOM** sind als zusätzliche Keys integriert.
  - Enthält auch das **„Admin-Passwort“ (ADMINPW)**, um die Änderung eines individuellen Passworts zu erzwingen.
  - Die **users.ini** ist für alle Benutzer sichtbar und – je nach Benutzer eingeschränkt - änderbar.
  - Die Änderungen werden erst beim nächsten Booten der Steuerung aktiv!

Rechte der einzelnen User

	User admin	User system	Standard User
Anzeige	Sieht alle Einträge aller Benutzer mit Ausnahme der individuellen Passwörter.	Sieht nur seinen Eintrag.	Sieht nur seinen Eintrag.
Vergabe von Namen und Keys	Kann Namen und Keys vergeben.	nein	nein
Löschung/Änderung des Namens	Kann nicht geändert oder gelöscht werden.	Kann nicht geändert oder gelöscht werden.	Kann durch den admin geändert oder gelöscht werden.
Berechtigungen	Können nicht geändert oder gelöscht werden.	Können nicht geändert oder gelöscht werden.	Können durch den admin geändert oder gelöscht werden.
Admin-Rechte (Schlüssel 1)	ja	nein	nein
Passwortänderung	Kann Admin-Passwort für andere Benutzer vorgeben.	Kann nur individuelles Passwort ändern.	Kann nur individuelles Passwort ändern.

Tab. 7: Gegenüberstellung User-Rechte

Besonderheiten Admin-Passwort

- Wenn vom admin ein **Admin-Passwort** für einen Standarduser eingetragen wurde, muss sich dieser Benutzer mit diesem Admin-Passwort einloggen.
  - Sein individuelles Passwort funktioniert dann nicht mehr.
  - Bei der Abfrage der Berechtigungen (WRITEKEYS, READKEYS, XCOMKEYS) wird 0 zurückgeliefert, bis ein neues individuelles Passwort eingetragen wurde. Bis dahin verfügt er **nicht** über besondere Schlüssel.

  - Er muss ein neues individuelles Passwort eintragen, das bisherige kann nicht nochmals verwendet werden.
  - Beim Eintragen eines individuellen Passworts wird das Admin-Passwort gelöscht.

Passwortvorgaben

Ein Passwort muss allen folgenden Konventionen entsprechen:

- Länge 8-33 Zeichen
- Kleinbuchstaben a-z.
- Großbuchstaben A-Z
- Zahlen 0-9

Wenn nicht alle 4 Bedingungen erfüllt sind, wird das Passwort **nicht** übernommen.

XCOM-Berechtigungen

- Der Name des Items für die XCOM-Berechtigungen lautet: XCOMKEYS
- Die XCOM-Berechtigungen werden nur durch Namen eingegeben, Zahlen sind nicht möglich.
  - Es können 31 Berechtigungen vergeben werden.
  - Die Namen sind im OS fest hinterlegt.
  - Erlaubt sind die Zeichen a-z, A-Z und 0-9.
  - Entspricht den Konventionen der anderen Schlüsselnamen.

HINWEIS! XCOMKEYS werden in aktuellen Betriebssystemen (bis 1.18.0.00 bei JC-4xx und bis 1.26.0.00 bei JC-9xx) noch nicht ausgewertet.

Beispiele für users.ini

Die Datei `users.ini` wird, je nach eingeloggtem Benutzer, unterschiedlich angezeigt.

Login als Benutzer "admin"

Der Benutzer „admin“ sieht alle eingetragenen Benutzer, außer deren Passwörter (Eintrag PW) und kann Änderungen vornehmen.

```
[USER1]
NAME=admin
ADMINPW=
PW=admin
READ-
KEYS=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27
,28,29,30,31
WRITE-
KEYS=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27
,28,29,30,31
XCOMKEYS= PlcControl, ProgControl, ProgDebug, ProgProfile, ProgDownload, Os-
Download, VarRead, VarWrite, FileRead, FileWrite

[USER15]
NAME=Homer
ADMINPW=
READKEYS=15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
WRITEKEYS=11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
XCOMKEYS= ProgDownload, OsDownload, VarRead, FileRead
```

```
[USER33]
NAME=system
ADMINPW=
READKEYS=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,
27,28,29,30,31
WRITEKEYS=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26
,27,28,29,30,31
XCOMKEYS= PlcControl, ProgControl, ProgDebug, ProgProfile, ProgDownload,
OsDownload, VarRead, VarWrite, FileRead, FileWrite
```

Login als Benutzer "Homer"

```
[USER15]
NAME=Homer
ADMINPW=
PW=Simpson99
READKEYS=15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
WRITEKEYS=11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
XCOMKEYS= ProgDownload, OsDownload, VarRead, FileRead
```

Protokollierung

Protokolldatei

Um Zugriffe auf die Benutzerverwaltung nachvollziehbar zu machen, wird für bestimmte Aktionen ein Eintrag in eine Protokolldatei gemacht.

Pfad und Dateiname	Format	Zugriffsrechte
/Drive_C/UserLogIn.log	Textdatei	admin

Tab. 8: Protokolldatei

Die Einträge werden an die bestehende Datei angehängt. Es ist sinnvoll, sie von Zeit zu Zeit zu löschen, um sie nicht zu groß werden zu lassen.

Aktionen

Folgende Aktionen werden protokolliert:

- Logins mit unbekanntem Benutzer-Namen
- Logins mit falschem Passwort
- Änderung der ADMIN-Passworts
- Änderung des Benutzer-Passworts
- Anlegen eines neuen Benutzers
- Löschen eines bestehenden Benutzers

Beispiel

Beispiel für den Inhalt der Log-Datei. Dem Meldungstext werden Datum (Monat/Tag/Jahr) und Tageszeit vorangestellt.

```
09/17/2024 10:54:24    Set ADMIN password for user 'system'.
09/17/2024 10:55:54    Bad authorization of user 'system'
09/17/2024 10:58:59    Set password for user 'system'.
09/17/2024 11:12:40    Unknown user name 'Marge'.
09/17/2024 13:00:33    Add user 'Marge'.
09/17/2024 13:09:31    Remove user 'Marge'.
```

Login sperren

Um zu verhindern, dass ein Benutzer-Passwort durch „Probieren“ ermittelt werden kann, wird ein Benutzer nach 5-maliger Eingabe des falschen Passworts gesperrt. Die Sperrung kann nur durch einen Reboot der Steuerung aufgehoben werden.

STX-Programm-Downloadschutz

Um einen minimalen Schutz vor unberechtigtem Download eines STX-Anwenderprogramms zu ermöglichen, kann wie folgt vorgegangen werden:

1. Öffnen Sie die FTP-Verbindung zur Steuerung als Benutzer „admin“.
2. Laden Sie die Datei „/System/flashdisklock.ini“ aus der Steuerung und setzen Sie mit einem Texteditor das Item „app=0“.
3. Übertragen Sie die Datei auf die Steuerung.
   ⇨ Beim Speichern der geänderten Datei auf der Steuerung werden alle Schlösser vom Verzeichnis „/app“ gelöscht.
4. Downloaden Sie das STX-Anwenderprogramms in JetSym.
5. Laden Sie die Datei „/System/flashdisklock.ini“ aus der Steuerung und setzen Sie mit einem Texteditor das Item „app=1“.
6. Übertragen Sie die Datei auf die Steuerung.
   ⇨ Beim Speichern der geänderten Datei auf der Steuerung wird das „admin“-Schloss am Verzeichnis „/app“ angebracht.

⇨ Jetzt kann nur noch vom Benutzer "admin" auf das Verzeichnis "/app" und die darin enthaltenen Programmdateien zugegriffen werden.

1.4 Belegung der Flash-Disk einsehen

Den Anwendungsbereich der internen Flash-Disk können Sie einsehen.
Einzelheiten zur Belegung des Anwendungsbereichs finden Sie in diesem Kapitel.

1.4.1 Belegung der Flash-Disk

Infodatei

Sie können den Bereich mit den Anwendungsdaten auf der internen Flash-Disk einsehen.
Die Belegung des Bereichs mit den Anwendungsdaten lesen Sie aus der Datei /System/flashdiskinfo.txt.

Beispiel

Das Beispiel zeigt die mögliche Belegung der Flash-Disk einer JetControl 340 (4 MB):

```
Name : flash disk
Date : 25.11.2008
Time : 15:04
Tracks: 64
Track 0: sectors: 128 (used: 81 / blocked: 47 / free: 0)
Track 1: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 2: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 3: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 4: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 5: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 6: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 7: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 8: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 9: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 10: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 11: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 12: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 13: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 14: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 15: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 16: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 17: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 18: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 19: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 20: sectors: 128 (used: 64 / blocked: 64 / free: 0)
Track 21: sectors: 128 (used: 85 / blocked: 43 / free: 0)
Track 22: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 23: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 24: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 25: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 26: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 27: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 28: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 29: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 30: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 31: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 32: sectors: 128 (used: 128 / blocked: 0 / free: 0)
Track 33: sectors: 128 (used: 105 / blocked: 0 / free: 23)
Track 34: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 35: sectors: 128 (used: 0 / blocked: 0 / free: 128)
```

```
Track 36: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 37: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 38: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 39: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 40: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 41: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 42: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 43: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 44: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 45: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 46: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 47: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 48: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 49: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 50: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 51: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 52: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 53: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 54: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 55: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 56: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 57: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 58: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 59: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 60: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 61: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 62: sectors: 128 (used: 0 / blocked: 0 / free: 128)
Track 63: sectors: 128 (used: 0 / blocked: 0 / free: 128)
```

```
Total: sectors: 8192 (used: 4175 / blocked: 154 / free: 3863)
```

```
Used : 2120900 byte
Blocked: 78232 byte
Free : 1962404 byte
Total : 4161536 byte
```

Elemente der Infodatei

Tracks und Sektoren bilden die Verwaltungseinheiten der Flash-Disk. Die Infodatei enthält folgende Elemente:

Element	Beschreibung
`Name`	Fester Name der Flash-Disk
`Date/Time`	Zeitpunkt, wann die Flash-Disk zuletzt formatiert wurde
`Tracks`	Gesamtzahl der Tracks
`Track xx: sectors: 128`	Belegung der Sektoren eines Tracks
`Total: sectors:`	Gesamtstatistik der Sektoren
`Used`	Gesamtzahl der belegten Bytes
`Blocked`	Gesamtzahl der blockierten Bytes

Element	Beschreibung
`Free`	Gesamtzahl der freien Bytes
`Total`	Gesamtgröße der Flash-Disk

Tab. 9: Elemente der Infodatei

Zustände des Sektors

Die kleinste Verwaltungseinheit der Flash-Disk, der Sektor, kann folgende Zustände einnehmen:

Zustand	Bedeutung
`used`	Der Sektor ist durch Daten belegt.
`blocked`	Der Sektor ist nicht mehr belegt, kann aber aus Verwaltungsgründen noch nicht wieder benutzt werden.
`free`	Der Sektor ist nicht belegt und kann benutzt werden.

Tab. 10: Zustände des Sektors

1.5 Formatieren und Prüfen

Dieses Kapitel beschreibt:

– Flash-Disk formatieren
– SD-Karte formatieren
– SD-Karte prüfen
– USB-Stick formatieren
– USB-Stick prüfen

Die interne Flash-Disk braucht nicht durch eine separate Funktion geprüft werden, da sie auf maximale Sicherheit der Verwaltungsstrukturen ausgelegt ist.

ⓘ Info

Diese Funktionen werden nicht von allen Steuerungen unterstützt. Lesen Sie in der Betriebsanleitung Ihrer Steuerung nach, ob diese die o.g. Funktionen unterstützt.

Funktionsweise

Wenn das Gerät bootet, prüft das Betriebssystem den Inhalt des Steuerregisters. Das Steuerregister ist Bestandteil des Dateisystems.

Abhängig vom Wert in diesem Register werden die folgenden Funktionen ausgeführt:

– Flash-Disk formatieren
– SD-Karte formatieren
– USB-Stick formatieren
– SD-Karte überprüfen
– USB-Stick überprüfen

Registernummer

Die Nummer des Steuerregisters ist 202936

1.5.1 Flash-Disk formatieren

In den folgenden Fällen ist es erforderlich, die Flash-Disk neu zu formatieren:

– Wenn Sie eine Betriebssystemversion mit einem geänderten Format der Flash-Disk einspielen
– Wenn Informationen zur Verwaltung der Flash-Disk zerstört wurden

Auswirkungen

– Alle Dateien und Verzeichnisse des Anwenderbereichs werden gelöscht.
– Systemdateien und Systemverzeichnisse werden nicht beeinflusst.

Flash-Disk formatieren

Damit das Gerät die interne Flash-Disk formatiert, führen Sie folgende Schritte aus:

1. Schalten Sie das Gerät ein.
2. Beschreiben Sie das Steuerregister 202936 des Dateisystems mit dem Wert -999720373 (0xc4697a4b).
3. Schalten Sie das Gerät aus.
4. Schalten Sie das Gerät ein.

⇨ Während des Boot-Vorgangs wird die Flash-Disk formatiert und das Steuerregister 202936 auf 0 gesetzt.

1.5.2 SD-Karte formatieren

Im folgenden Fall ist es erforderlich, die SD-Karte neu zu formatieren:

– Wenn Informationen zur Verwaltung der SD-Karte zerstört wurden

Auswirkungen

Alle Dateien und Verzeichnisse auf der SD-Karte werden gelöscht.

SD-Karte formatieren

Damit das Gerät die SD-Karte formatiert, führen Sie folgende Schritte aus:

1. Schalten Sie das Gerät ein.
2. Beschreiben Sie das Steuerregister 202936 des Dateisystems mit dem Wert -748362163 (0xd364e64d).
3. Schalten Sie das Gerät aus.
4. Schalten Sie das Gerät ein.

⇨ Während des Boot-Vorgangs wird die SD-Karte formatiert und das Steuerregister 202936 auf 0 gesetzt.

1.5.3 USB-Stick formatieren

Unter Umständen kann es notwendig sein, den USB-Stick neu zu formatieren. Dies kann dann der Fall sein, wenn Informationen zur Verwaltung des USB-Sticks zerstört wurden.

Auswirkungen

Alle Dateien und Verzeichnisse auf dem USB-Stick werden gelöscht.

USB-Stick formatieren

Führen Sie folgende Schritte aus, damit der USB-Stick formatiert wird:

1. Schalten Sie das Gerät ein.
2. Beschreiben Sie das Steuerregister des Dateisystems mit dem Wert (0x8f3d5185).
3. Schalten Sie das Gerät aus.
4. Schalten Sie das Gerät ein.

⇨ Während der Boot-Phase des Geräts wird der USB-Stick formatiert und das Steuerregister auf 0 gesetzt.

1.5.4 SD-Karte prüfen

Im folgenden Fall ist es erforderlich, die SD-Karte auf Fehler zu prüfen:

– Wenn das Gerät während eines Zugriffs auf die SD-Karte ausgeschaltet wurde.

Auswirkungen

Alle Dateien und Verzeichnisse auf der SD-Karte werden geprüft und Fehler ggf. korrigiert.

Anschließend sind die Verwaltungsstrukturen auf der SD-Karte in einem konsistenten Zustand.

Abhängig von der Größe der SD-Karte und der Anzahl der Verzeichnisse und Dateien verlängert sich die Boot-Phase bis auf mehrere Minuten.

SD-Karte prüfen

Damit das Gerät die SD-Karte prüft, führen Sie folgende Schritte aus:

1. Schalten Sie das Gerät ein.
2. Beschreiben Sie das Steuerregister 202936 des Dateisystems mit dem Wert 748371092 (0x2c9b3c94).
3. Schalten Sie das Gerät aus.
4. Schalten Sie das Gerät ein.

⇨ Während des Boot-Vorgangs prüft das Gerät die SD-Karte. Der Wert im Steuerregister bleibt erhalten, so dass die Prüfung bei jedem Neustart des Geräts durchgeführt wird.

Einschränkungen

Die Funktion repariert die Verwaltungsstrukturen der SD-Karte. Sie können die SD-Karte somit weiter verwenden. Zu beachten ist, dass z. B. das Gerät die Daten einer unvollständig geschriebenen Datei nicht immer restaurieren kann.

1.5.5 USB-Stick überprüfen

Unter Umständen kann es notwendig sein, den USB-Stick auf Fehler zu überprüfen. Dies kann dann der Fall sein, wenn das Gerät ausgeschaltet wurde während auf den USB-Stick zugegriffen wurde.

Auswirkungen

– Alle Dateien und Verzeichnisse auf dem USB-Stick werden überprüft und Fehler gegebenenfalls korrigiert.

  Anschließend sind die Verwaltungsstrukturen auf dem USB-Stick in einem konsistenten Zustand.

– Abhängig von der Größe des USB-Sticks und der Anzahl der Verzeichnisse und Dateien kann die Boot-Phase bis auf mehrere Minuten verlängert werden.

Überprüfung

Führen Sie folgende Schritte aus, damit das Gerät den USB-Stick überprüft:

1. Schalten Sie das Gerät ein.
2. Beschreiben Sie das Steuerregister des Dateisystems mit dem Wert (0x17dbd42a).
3. Schalten Sie das Gerät aus.
4. Schalten Sie das Gerät ein.

⇨ Während der Boot-Phase des Geräts wird der USB-Stick überprüft. Der Wert im Steuerregister bleibt erhalten, so dass die Überprüfung bei jedem Neustart des Geräts durchgeführt wird.

Einschränkungen

Die Funktion repariert nur die Verwaltungsstrukturen des USB-Sticks, damit dieser weiter verwendet werden kann. Zum Beispiel können die Daten einer unvollständig geschriebenen Datei nicht immer restauriert werden.

2 E-Mail-Funktion

Der Anwender erstellt Vorlagendateien für E-Mails, in die die Steuerung beim Versenden ggf. Variablenwerte einfügt. Dies ermöglicht den automatisierten Versand von Protokollen oder anderen produktions- oder wartungsrelevanten Informationen per E-Mail.

Die Steuerung sendet die E-Mails an einen E-Mail-Server, der die Weiterleitung an den Empfänger der E-Mail übernimmt.

Dieses Kapitel beschreibt, wie der Anwender die E-Mail-Funktion konfiguriert und E-Mails sowohl erstellt als auch sendet.

E-Mail-Funktion aktivieren

Für die Aktivierung der E-Mail-Funktion müssen folgende Voraussetzungen erfüllt sein:

– Die Steuerung kann mit bestimmten Optionen bestellt werden. Je nach Option ist die E-Mail-Funktion freigeschaltet oder nicht.
– In der Boot-Phase der Steuerung muss eine gültige E-Mail-Konfigurationsdatei /EMAIL/email.ini vorhanden sein.

Wenn beide Voraussetzungen erfüllt sind, ist das entsprechende Bit im Webstatusregister gesetzt.

Kenntnisse des Programmierers

Im Umgang mit E-Mails sind folgende Kenntnisse vorausgesetzt:

– Da die Konfiguration der E-Mail-Funktion über Dateien geschieht und die E-Mails selbst auf Dateien basieren, sind Kenntnisse des Dateisystems erforderlich.
– Kenntnisse über IP-Netzwerke sind erforderlich.

2.1 Konfigurieren der E-Mail-Funktion

Dieses Kapitel beschreibt die Konfiguration der E-Mail-Funktion, so dass E-Mails im Anwendungsprogramm versendet werden können.

Das Gerät liest die Konfigurationsdaten während der Boot-Phase aus der Datei /EMAIL/email.ini.

Voraussetzungen

Für die Erstellung der Konfigurationsdatei müssen folgende Voraussetzungen erfüllt sein:

– Die IP-Adresse des E-Mail-Servers ist bekannt.
– Wenn die IP-Adresse des E-Mail-Servers nicht bekannt ist, muss die Namensauflösung über einen DNS-Server möglich sein, siehe Verwenden von Namen für IP-Adressen.
– Die Parameter für Anmeldung und Authentifizierung am E-Mail-Server sind bekannt.

Diese Informationen erhalten Sie vom jeweiligen Netzwerkadministrator.

2.1.1 Aufbau der Konfigurationsdatei

Die Konfiguration des E-Mail-Clients geschieht mit den Daten aus der Datei /EMAIL/email.ini. Das Gerät liest die Datei nur während der Boot-Phase.

Aufbau der Konfigurationsdatei

Die Konfigurationsdatei ist eine Textdatei, deren Einträge in Sektionen gruppiert sind.

– In diesen Sektionen trägt der Anwender Werte ein, mit denen der E-Mail-Client arbeitet.
– Leerzeilen können Sie beliebig einfügen.
– Kommentarzeilen werden mit "!", "#" oder ";" eingeleitet.

Sektionen

Die Konfigurationsdatei enthält bis zu drei Sektionen. Die Sektion [SMTP] muss vorhanden sein. Die anderen Sektionen braucht der Anwender nur anlegen, wenn sie benötigt werden:

Sektion	Konfigurationswerte
[SMTP]	– IP-Adresse und Port-Nummer des SMTP-Servers – Anmeldeparameter
[POP3]	– IP-Adresse und Port-Nummer des POP3-Servers – Anmeldeparameter
[DEFAULT]	– Name einer E-Mail-Vorlagendatei, die Standardwerte enthält

2.1.2 Sektion [SMTP]

In dieser Sektion geben Sie die Parameter zur Verbindung mit dem SMTP-Server an.

Beispiel:

```
[SMTP]
IP              = 192.168.40.1
PORT          = 25000
HELO          = JetControl_2
USER          = JetControl0815
PASSWORD          = MyPassWord
SECUREEMAIL      = 1
STARTTLS          = 0
```

Authentifizierung

Bei dieser Art der Authentifizierung führt das Gerät vor dem E-Mail versenden eine Benutzeranmeldung beim SMTP-Server durch. Für die Anmeldung ist USER und PASSWORD erforderlich. Das Gerät unterstützt folgende Authentifizierungsverfahren.

– LOGIN
– PLAIN
– CRAM-MD5

Konfigurationswerte

IP

Eigenschaft	Beschreibung
Im Beispiel	– 192.168.40.1 – IP-Adresse des SMTP-Servers; kann auch als Name angegeben werden
Gültige Werte	– > 1.0.0.0 – < 223.255.255.255
Nicht gültige Werte	– Netzwerkadresse – Broadcast-Adresse
Bei ungültigem Wert oder nicht vorhandenem Eintrag	E-Mail-Funktion ist nicht verfügbar

PORT

Eigenschaft	Beschreibung
Im Beispiel	– 25 000 – Port-Nummer des SMTP-Servers
Gültige Werte	– > 0 – < 65.536
Nicht gültige Werte	– > 65.335
Bei nicht vorhandenem Eintrag	– 25

HELO

Eigenschaft	Beschreibung
Im Beispiel	– JetControl_2 – Name für die Anmeldung beim E-Mail-Server
Gültige Werte	– Zeichenkette mit maximal 63 Zeichen
Bei nicht vorhandenem Eintrag	– Wird beim Senden der E-Mail von der Steuerung der Eintrag von [FROM] verwendet

USER

Eigenschaft	Beschreibung
Im Beispiel	– JetControl0815 – Anmeldename bei SMTP-Authentifizierung; – Wenn dieser Eintrag vorhanden ist, ist auch der Eintrag PASSWORD erforderlich.
Gültige Werte	– Zeichenkette mit maximal 63 Zeichen
Bei nicht vorhandenem Eintrag	– 25

PASSWORD

Eigenschaft	Beschreibung
Im Beispiel	– MyPassWord – Anmeldepasswort bei SMTP-Authentifizierung. – Wenn dieser Eintrag vorhanden ist, ist auch der Eintrag USER erforderlich.
Gültige Werte	– Zeichenkette mit maximal 63 Zeichen
Bei nicht vorhandenem Eintrag	– Keine SMTP-Authentifizierung

SECUREEMAIL

Eigenschaft	Beschreibung
Im Beispiel	– 1
Wert = 1	– Der SMTP-Server muss Verschlüsselung unterstützen bzw. schreibt Verschlüsselung vor. – Wenn dazu STARTTLS = 0 ist, sendet die Steuerung seit dem Verbindungsaufbau verschlüsselt.

Eigenschaft	Beschreibung
	– Wenn dazu STARTTLS = 1 ist, erfolgt der Verbindungsaufbau unverschlüsselt und danach wird die Verschlüsselung ausgehandelt.
Wert = 0	– Die E-Mail wird komplett unverschlüsselt verschickt. – Ein STARTTLS = 1 wird nicht berücksichtigt.
Gültige Werte	0 oder 1
Bei nicht vorhandenem Eintrag	E-Mail wird unverschlüsselt verschickt.

STARTTLS

Eigenschaft	Beschreibung
Im Beispiel	– 0
Wert = 1	– Wenn der SMTP-Server STARTTLS verlangt, muss STARTTLS = 1 sein. – Der SMTP-Server verlangt zuerst einen unverschlüsselten Verbindungsaufbau über einen Klartextport, z. B. 25 oder 587. – Nach dem erfolgreichen Verbindungsaufbau, handeln Server und Client einen Verschlüsselungsalgorithmus aus, z. B. TLS 1.0 oder TLS 1.1. – Der nachfolgende Nachrichteninhalt wird dann verschlüsselt über den gleichen Port verschickt.
Wert = 0	– Wenn der SMTP-Server kein STARTTLS verlangt oder unterstützt. – Die Steuerung sendet seit dem Verbindungsaufbau verschlüsselt.
Gültige Werte	0 oder 1
Bei nicht vorhandenem Eintrag	E-Mail wird unverschlüsselt verschickt.

2.1.3 Sektion [POP3]

In dieser Sektion geben Sie die Parameter zur Verbindung mit dem POP3-Server an.

Nur wenn der E-Mail-Server eine Authentifizierung über POP3-vor-SMTP benötigt, ist diese Sektion erforderlich.

Beispiel:

```
[POP3]
IP              = 192.168.40.1
PORT              = 25100
USER              = JetControl4711
PASSWORD          = Pop3PassWord
```

Authentifizierung

Bei dieser Art der Authentifizierung baut das Gerät zuerst eine Verbindung zum POP3-Server auf. Für die Anmeldung ist USER und PASSWORD erforderlich. Daraufhin schaltet der SMTP-Server den Versand von E-Mails für eine bestimmte Zeit (meist 10 bis 30 min) frei.

Konfigurationswerte

IP

Eigenschaft	Beschreibung
Im Beispiel	– 192.168.40.1 – IP-Adresse des POP3-Servers; kann auch als Name angegeben werden.
Gültige Werte	– > 1.0.0.0 – < 223.255.255.255
Nicht gültige Werte	– Netzwerkadresse – Broadcast-Adresse
Bei ungültigem Wert oder nicht vorhandenem Eintrag	Keine POP3-Anmeldung

PORT

Eigenschaft	Beschreibung
Im Beispiel	– 25.100 – Port-Nummer des POP3-Servers
Gültige Werte	– > 0 – < 65.536
Nicht gültige Werte	– > 65.335
Bei nicht vorhandenem Eintrag	110

USER

Eigenschaft	Beschreibung
Im Beispiel	– JetControl4711 – Anmeldename bei POP3-Anmeldung. – Wenn dieser Eintrag vorhanden ist, ist auch der Eintrag PASSWORD erforderlich.
Gültige Werte	– Zeichenkette mit maximal 63 Zeichen
Bei nicht vorhandenem Eintrag	Keine POP3-Anmeldung

PASSWORD

Eigenschaft	Beschreibung
Im Beispiel	– Pop3PassWord – Anmeldepasswort bei POP3-Anmeldung; – Wenn dieser Eintrag vorhanden ist, ist auch der Eintrag USER erforderlich.
Gültige Werte	– Zeichenkette mit maximal 63 Zeichen
Bei nicht vorhandenem Eintrag	Keine POP3-Anmeldung

2.1.4 Sektion [DEFAULT]

In dieser Sektion geben Sie den Namen einer E-Mail-Vorlagendatei an, die Standardeinstellungen für den E-Mail-Versand enthalten. Wenn die entsprechende Sektion in der jeweiligen E-Mail-Vorlagendatei nicht enthalten ist, setzt das Gerät die hier gemachten Definitionen beim E-Mail-Versand ein.

Beispiel

```
[DEFAULT]
MAILCFG = EmailDefaults.cfg
```

2.1.5 Beispiele für eine Konfigurationsdatei

Hier finden Sie einige Beispiele für die E-Mail-Konfigurationsdatei /EMAIL/email.ini.

Minimalkonfiguration

Wenn keine Authentifizierung benötigt wird und der IP-Port am SMTP-Server dem Standardwert entspricht, beschränkt sich der Inhalt der Konfigurationsdatei auf die IP-Adresse des SMTP-Servers.

```
[SMTP]
IP        = 192.168.40.1
```

Authentifizierung über POP3-Anmeldung

Wenn der E-Mail-Server eine vorherige Anmeldung über POP3 verlangt und eine E-Mail-Vorlagendatei mit Standardeinstellungen definiert ist.

```
[SMTP]
IP        = 192.168.40.1
[POP3]
IP         = 192.168.40.1
USER        = JetControl4711
PASSWORD    = Pop3PassWord

[DEFAULT]
MAILCFG    = EmailDefaults.cfg
```

Authentifizierung über SMTP

Wenn der E-Mail-Server eine verschlüsselte Authentifizierung verlangt.

```
[SMTP]
IP            = 192.168.40.1
USER            = JetControl0815
PASSWORD        = MyPassWord
SECUREEMAIL    = 1
STARTTLS        = 0
```

2.2 E-Mail erstellen

Dieses Kapitel beschreibt die E-Mail-Erstellung. Der Versand dieser E-Mails geschieht dann im Anwenderprogramm.

Für jede E-Mail ist eine E-Mail-Vorlagendatei zu erstellen.

2.2.1 Name der E-Mail-Vorlagendatei

Absoluter Dateiname

Bei Verwendung der STX-Funktion

```
EMailSend()
```

können Dateinamen und Verzeichnis (Pfad) absolut, in den durch das Dateisystem gegebenen Grenzen, beliebig gewählt werden.

2.2.2 Struktur der E-Mail-Vorlagendatei

Eine E-Mail-Vorlagendatei ist eine Textdatei, die in Sektionen strukturiert ist. Aus den Informationen in diesen Sektionen stellt das Gerät beim Versenden die E-Mail zusammen.

Struktur

– Die Sektionen [FROM] und [TO] sind erforderlich. Entweder in der zu versendenden E-Mail selbst oder in der E-Mail-Vorlagendatei mit den Standardeinstellungen.
– Alle Parameter in diesen Sektionen können mit Tags, die Echtzeit-Steuerungswerte enthalten, versehen werden, siehe Einfügen von Echtzeit-Steuerungswerten (siehe Seite 64).

E-Mail-Vorlagendatei

[FROM]

Absender

[TO]

Empfänger

[CC]

Weitere(r) Empfänger

[SUBJECT]

Betreff

[ATTACHMENT]

Vollständiger Pfad- und Dateiname

[MESSAGE]

Inhalt der E-Mail

Sektionen

[FROM]

Parameter	Beschreibung
Beschreibung	Absender der E-Mail
Bemerkung	Fragen Sie den zuständigen IT-Administrator, was hier einzutragen ist.
Länge	63 Zeichen
Beispiel	`[FROM]` `JetControl@bucherautomation.com`

[TO]

Parameter	Beschreibung
Beschreibung	Empfänger der E-Mail
Bemerkung	Mehrere Empfänger werden durch das Semikolon getrennt.

Parameter	Beschreibung
Länge	255 Zeichen
Beispiel	[TO] service@mydomain.com

[CC]

Parameter	Beschreibung
Beschreibung	Weitere(r) Empfänger der E-Mail
Bemerkung	Mehrere Empfänger werden durch das Semikolon getrennt.
Länge	255 Zeichen
Beispiel	[CC] service@mydomain.com;hotline@mydomain.com

[SUBJECT]

Parameter	Beschreibung
Beschreibung	Betreff
Länge	255 Zeichen
Beispiel	[SUBJECT] Fatal Error

[ATTACHMENT]

Parameter	Beschreibung
Beschreibung	Vollständiger Pfad und Name einer anzuhängenden Datei, siehe Beispiel 1.
Bemerkung	Die Datei muss eine Textdatei sein. Ein String aus Pfad und Dateiname lässt sich als Echtzeit-Steuerungswert alternativ auslesen, siehe Beispiel 2. Nur ein Anhang ist möglich.
Länge	511 Zeichen
Beispiel 1	[ATTACHMENT] /logfiles/error_report.log
Beispiel 2	[ATTACHMENT] <JW:DTAG name="R420" type="STRING"/>

[MESSAGE]

Parameter	Beschreibung
Beschreibung	Inhalt der E-Mail
Bemerkung	Nur Text möglich
Länge	65.535 Zeichen
Beispiel	[MESSAGE] Have a nice day ! JetControl.

2.2.3 Einfügen von Echtzeit-Steuerungswerten

Aktuelle Echtzeit-Steuerungswerte werden über Tag-Funktionen in die Parameter der Sektionen integriert. Damit können Zustände von Registern, Textregistern, Eingängen, Ausgängen und Merkern angezeigt werden.

2.3 E-Mail versenden

Dieses Kapitel beschreibt, wie zuvor erstellte E-Mails im Anwendungs-programm versendet werden.

Beim Versenden aus dem Anwendungsprogramm stellt das Gerät aus einer E-Mail-Vorlagendatei eine E-Mail zusammen und fügt ggf. Variablenwerte ein.

Verarbeitung im Anwendungsprogramm

Der E-Mail-Versand kann relativ lange dauern. Deshalb werden, während diese Funktion ausgeführt wird, die anderen Tasks des Anwendungsprogramms bearbeitet.

Nur ein E-Mail-Funktionsaufruf ist gleichzeitig möglich.

Während ein E-Mail-Versand eines Tasks gerade läuft, werden daher die Tasks, die die E-Mail-Funktion aufrufen, solange blockiert, bis der E-Mail-Versand abgeschlossen ist. Es darf in dieser Zeit kein TaskBreak oder TaskRestart auf diesen Task oder ein Programmneustart mit JetSym ausgeführt werden.

Systemfunktion 110*

Seit JetSym 5.0 ist die Systemfunktion 110 veraltet. Verwenden Sie stattdessen die JetSym-STX-Funktion `EMailSend()`.

2.3.1 STX-Funktion EMailSend()

Funktionsdeklaration

```
Function EmailSend(Const Ref FileName: String): Int;
```

Parameter	Beschreibung
`FileName`	Dateiname der E-Mail-Vorlagendatei muss absolut, mit Pfadangabe, angegeben werden.

Tab. 11: Parameter FileName

Rückgabewert

Parameter	Beschreibung
0	Kein Fehler
-1	Interner Betriebssystemfehler
-2	FROM nicht definiert
-3	TO nicht definiert
-4	Keine Verbindung zum E-Mail-Server
-10	E-Mail-Funktion nicht verfügbar
-11/-12	Interner Betriebssystemfehler

Tab. 12: Rückgabewert

JetSym-STX-Programm

Wenn ein Fehler in der Anlage auftritt, soll die Steuerung eine E-Mail verschicken.

```
Var
     nResult: Int;
        bSendError: Bool;
End_Var;

Task tMain Autorun
        ...
        If bSendError Then
             nResult := EmailSend('/ErrorMails/MainError.email');
             bSendError := FALSE;
        End_If;

        ...
End_Task;
```

2.4 Register

Dieses Kapitel beschreibt die Register, über die Sie den Bearbeitungsstatus der E-Mail-Funktion abfragen können.

2.4.1 Registerübersicht

Das Gerät stellt einige Register zur Verfügung, über die Sie den Bearbeitungsstatus der E-Mail-Funktion abfragen können.

Register	Beschreibung
202930	Webstatus
292932	IP-Adresse des SMTP-Servers
292933	IP-Adresse des POP3-Servers
292934	Port-Nummer des SMTP-Servers
292935	Port-Nummer des POP3-Servers
292937	Status der E-Mail-Bearbeitung
292938	ID des Tasks, der gerade eine E-Mail verschickt

Tab. 13: Registerübersicht

2.4.2 Registerbeschreibung

R 202930

Webstatus

Das Webstatusregister zeigt bitkodiert die Funktionen an, die verfügbar sind.

Register	Beschreibung	
Zugriff	Lesen	
Bedeutung des Bits		
Bit 0	**FTP-Server**	
	1 =	verfügbar
Bit 1	**HTTP-Server**	
	1 =	verfügbar
Bit 2	**E-Mail**	
	1 =	verfügbar
	Bit 2 ist nur gesetzt, wenn die Konfigurationsdatei / EMAIL/email.ini vorhanden ist.	
Bit 3	**Datendateifunktion**	
	1 =	verfügbar
Bit 4	**Modbus/TCP**	
	1 =	vorhanden
Bit 5	**Modbus/TCP**	
		verfügbar
Bit 7	**FTP-Client**	
		verfügbar
Wert nach Reset	Je nach bestellter Option	

Tab. 14: Webstatus R 202930

292932

IP-Adresse des SMTP-Servers kann hier gelesen werden, wie sie in der Datei /EMAIL/email.ini angegeben ist.

Eigenschaft	Beschreibung
Zugriff	Lesen
Wert nach Reset	Je nach Konfiguration
Wird wirksam	Wenn R 202930.2 = 1

Tab. 15: IP-Adresse des SMTP-Servers R 292932

R 292933

Die IP-Adresse des POP3-Servers kann hier gelesen werden, wie sie in der Datei /EMAIL/email.ini angegeben ist.

IP-Adresse des POP3-Servers

Eigenschaft	Beschreibung
Zugriff	Lesen
Wert nach Reset	Je nach Konfiguration
Wird wirksam	Wenn R 202930.2 = 1

Tab. 16: IP-Adresse des POP3-Servers R 292933

R 292934

IP-Adresse des POP3-Servers

Die Port-Nummer des SMTP-Servers kann hier gelesen werden, wie sie in der Datei /EMAIL/email.ini angegeben ist.

Eigenschaft	Beschreibung
Zugriff	Lesen
Wert nach Reset	Je nach Konfiguration
Wird wirksam	Wenn R 202930.2 = 1

Tab. 17: IP-Adresse des POP3-Servers R 292934

R 292935

Port-Nummer des POP3-Servers

Die Port-Nummer des POP3-Servers kann hier gelesen werden, wie sie in der Datei /EMAIL/email.ini angegeben ist.

Eigenschaft	Beschreibung
Zugriff	Lesen
Wert nach Reset	Je nach Konfiguration
Wird wirksam	Wenn R 202930.2 = 1

Tab. 18: Port-Nummer des POP3-Servers R 292935

R 292937

Status der E-Mail-Bearbeitung

Die Ausführung des E-Mail-Versands wird hier mitverfolgt.

Eigenschaft	Beschreibung
Zugriff	Lesen
Werte:	0 = Keine E-Mail wird versendet.
	1 = Übergabe der Parameter an den E-Mail-Client des Geräts.
	2 = E-Mail wird zusammengestellt und die Verbindung mit dem Server wird aufgenommen.
	3 = E-Mail wurde an den Server gesendet.

Tab. 19: Status der E-Mail-Bearbeitung R 292937

R 292938

Task-ID (E-Mail)

Die ID der Task, der gerade eine E-Mail verschickt, kann hier gelesen werden.

Eigenschaft	Beschreibung
Zugriff	Lesen
Werte:	0 … 99 = Task-ID
	255 = Kein Task, verschickt gerade eine E-Mail.
Werte nach Reset	255

Tab. 20: Task-ID (E-Mail) R 292938

3 FTP-Server

Der FTP-Server erlaubt den Zugriff auf Verzeichnisse und Dateien mit einem FTP-Client. Die Dateien können auf folgende Medien abgespeichert sein:

– Interne Flash-Disk
– SD-Karte
– USB-Stick

Dieses Kapitel beschreibt den Vorgang der Anmeldung und die vom FTP-Server unterstützten Kommandos.

FTP-Clients

Außer dem Kommandozeilen-FTP-Client, wie er bei vielen PC-Betriebs-systemen mitgeliefert wird, können Sie auch grafische FTP-Tools verwenden.

Anzahl möglicher Verbindungen

Der FTP-Server verwaltet vier gleichzeitig geöffnete FTP-Verbindungen.

Jeder weitere FTP-Client, der sich versucht mit dem FTP-Server zu verbinden, erhält auf seine Anfrage zum Aufbau einer Verbindung keine Antwort.

Unterstützte Kommandos

Der FTP-Server unterstützt die standardisierten Kommandos. Sie finden diese unter:

– Help-Menü des FTP-Servers; mit FTP-Server verbinden und als Kommando help oder help binary eingeben
– Im Internet nach FTP und Kommandos suchen

Wenn Sie sich nicht um die Kommandos kümmern wollen, empfehlen wir Ihnen ein FTP-Programm wie TotalCommander zu verwenden.

Kenntnisse des Programmierers

Dieses Kapitel setzt folgende Kenntnisse voraus:

– Dateisystem
– IP-Netzwerke
– FTP-Kommandos

3.1 Anmeldung

Um auf das Dateisystem über FTP zugreifen zu können, muss sich der FTP-Client bei der Aufnahme der Kommunikation wie folgt anmelden:

– Bei Servername geben Sie die IP-Adresse des Geräts ein.
– Bei Benutzername geben Sie den Benutzernamen ein, z. B. admin.
– Bei Passwort geben Sie das Passwort ein, z. B. admin.

Auslieferungszustand

Im Auslieferungszustand sind folgende Benutzer (User) eingerichtet:

[USER1]

NAME=admin

PW=admin

[USER33]

NAME=system

PW=system

Benutzer verwalten

Über die Benutzerverwaltung [▶ 11] des Dateisystems können Sie das Passwort ändern und weitere Benutzer hinzufügen.

3.2 FTPS Server

HINWEIS! Nur für JC-4xx und JC-9xx ab OS-Versionen JC-4xx: 1.14.0.00 und JC-9xx: 1.22.0.00 verfügbar.

Neben dem FTP-Server läuft auf der Steuerung auch ein FTPS-Server (FTPS = FTP Secure) über den eine verschlüsselte Verbindung zwischen Client und Server hergestellt werden kann.

– Der FTPS-Server verwendet die gleichen TCP/IP-Ports wie der FTP-Server.
– Die Einstellung, ob mit oder ohne Verschlüsselung kommuniziert werden soll, wird im FTP-Client vorgenommen.
– Es wird die Verschlüsselung über TLS/SSL unterstützt.

3.2.1 Benutzer-Verwaltung

Um größtmögliche Kompatibilität mit den bisherigen Betriebssystemen, die noch nicht FTPS unterstützen, zu erhalten, sind alle Benutzer über FTP **und** FTPS erreichbar, mit Ausnahme des Benutzers „admin“.

3.2.2 Anmelden als "admin"

Einloggen mit „admin“ – und damit Zugriff auf die Dateien der User-Verwaltung im Verzeichnis „System“ – ist nur über FTPS möglich.

4 HTTP-Server

Ein Standardbrowser reicht für den Zugriff auf den HTTP-Server aus.

Mit dem Browser können Dateien, die per FTP auf die Steuerung geladen wurden, gelesen und angezeigt werden.

Um Zugriff auf bestimmte Seiten zu erhalten, kann es, je nach Konfiguration des Dateisystems, erforderlich sein, Benutzername und Passwort einzugeben.

Vorgegebene Dateinamen

Die vorgegebenen Dateinamen sind **index.htm** und **index.html**.

Unterstützte Dateitypen

Folgende Dateitypen werden unterstüzt:

– *.htm, *.html, *.shtml
– *.txt, *.ini
– *.gif, *.tif, *.tiff, *.bmp, *.wbmp
– *.jpg, *.jpe, *.jpeg, *.png
– *.xml
– *.js, *.jar, *.java, *.class, *.cab
– *.ocx
– *.pdf, *.zip, *.doc, *.rtf
– *.css
– *.wml, *.wmlc, *.wmls, *.wmlsc
– *.ico, *.svg
– *.map
– *.ttf, *.woff, *.woff2

Kenntnisse des Programmierers

Dieses Kapitel setzt folgende Kenntnisse voraus:

– Dateisystem
– IP-Netzwerke

4.1 Server Side Includes

Der HTTP-Server enthält die Funktionalität Server Side Includes (SSI). Mit Hilfe dieser Funktion sind in einer HTML-Seite aktuelle Echtzeit-Steuerungswerte darstellbar.

Regeln

Wenn Sie aktuelle Echtzeit-Steuerungswerte in die HTML-Seite eintragen wollen, müssen Sie am Beginn dieser HTML-Seite den Tag Name Space angeben.

Im Tag Name Space wird der verwendete Name Space definiert.

Im Bodybereich der HTML-Seite folgen die Data-Tags.

Echtzeit-Steuerungs-Werte aktualisieren

Wenn die HTML-Seite in den Browser geladen wird, ersetzt der HTTP-Server die Data-Tags einmalig durch aktuelle Echtzeit-Steuerungswerte.

Um die Werte zu aktualisieren, muss die HTML-Seite immer wieder neu geladen werden.

Angestoßen wird der Ladevorgang vom Anwender, indem er im Browser die Adresse der Steuerung und den Namen der gewünschten Seite einträgt, z. B. *http://192.168.10.209/Homepage/SSI/ssiTimeAndDate.htm*.

4.1.1 Erster Eintrag in der HTML-Datei

Aufbau

Der Tag Name Space muss der erste Eintrag in der HTML-Datei sein. Er hat den folgenden Aufbau:

```
<NS:DTAG xmlns:NS=http://bucherautomation.com/ssi/jetcontrol/
```

Wobei NS den Name Space darstellt. Der Name Space besteht aus einer Zeichenfolge mit einer maximalen Länge von 63 Zeichen.

Der hier eingeführte Name Space wird bei den folgenden Data Tags wieder verwendet. Die übrigen Teile der Zeile sind festgelegt und müssen genauso angegeben werden.

In den folgenden Beispielen ist der verwendete Name Space JC.

4.1.2 Einfügen von Echtzeit-Steuerungswerten

Aktuelle Echtzeit-Steuerungswerte werden über Tag-Funktionen in die Parameter der Sektionen integriert. Damit können Zustände von Registern, Textregistern, Eingängen, Ausgängen und Merkern angezeigt werden.

Tag-Kennzeichnung

Alle Tags beginnen und enden mit bestimmten Zeichenfolgen. Zwischen diesen Tag-Begrenzern sind die Variablen definiert.

Tag-Kennzeichnung

Parameter	Beschreibung
Beginn eines Tags	<JC:DTAG
Ende eines Tags	/>

Variablendefinition

Die Variablendefinition in einem Tag enthält Attribute, die z. B. die Darstellung des Variablenwerts festlegen.

name

Parameter	Beschreibung
Beschreibung	Variablenname
Bemerkung	Kennbuchstabe gefolgt von der Variablennummer
Beispiel	name="R1000023"

type

Parameter	Beschreibung
Beschreibung	Variablentyp der Darstellung
Beispiel	type="REAL"

format

Parameter	Beschreibung
Beschreibung	Darstellungsformat
Bemerkung	Siehe Formatdefinition
Beispiel	`format="+0####.###"`

factor

Parameter	Beschreibung
Beschreibung	Faktor, mit dem der Echtzeit-Steuerungswert multipliziert wird
Bemerkung	Die Multiplikation geschieht vor der Addition des Offsets
Beispiel	`factor="1.5"`

offset

Parameter	Beschreibung
Beschreibung	Wert, der zum Echtzeit-Steuerungswert addiert wird
Bemerkung	Die Addition geschieht nach der Multiplikation mit dem Faktor
Beispiel	`offset="1000"`

Formatdefinition

Die Darstellung der Variablen können Sie über das Attribut steuern.

– Die Anzahl der Stellen/Zeichen, mit denen eine Variable dargestellt wird, legt das Zeichen "#" fest.
– Durch Voranstellen einer "0" wird bei den Registertypen INT, INTX und REAL die Ausgabe von führenden Nullen festgelegt.
– Durch Voranstellen eines "+" wird bei den Registertypen INT und REAL die Ausgabe eines Vorzeichens festgelegt.
– Durch Voranstellen eines Leerzeichens wird bei den Registertypen INT und REAL die Ausgabe eines Leerzeichens für positive Werte festgelegt.

Register/Textregister

Der Variablenname beginnt mit einem großen "R" gefolgt von der Registernummer.

Folgende Typen sind möglich:

Typ	Beschreibung
INT (Standardtyp)	Ganzzahl dezimal
INTX	Ganzzahl hexadezimal
INTB	Ganzzahl binär
BOOL	Registerinhalt = 0 --> Anzeige: 0 Registerinhalt != 0 --> Anzeige: 1
REAL	Gleitkommazahl dezimal
STRING	Textregister

Tab. 21: Register/Textregister

Beispiel:

```
<JC:DTAG name="R1000250" type="REAL" format="+0####.###" factor="3.25" off-
set="500" />
```

Ergebnis:

Aufgrund dieses Befehls wird der Inhalt von Register 1000250 mit 3,25 multipliziert und zum Produkt 500 addiert. Im Webbrowser erscheint das Ergebnis mit Vorzeichen und mindestens fünf Vorkommastellen. Wenn erforderlich, werden führende Nullen hinzugefügt. Darüber hinaus werden drei Nachkommastellen angefügt.

Merker

Der Variablenname beginnt mit einem großen "F" gefolgt von der Merkernummer.

Folgende Typen sind möglich:

Typ	Beschreibung
BOOL (Standardtyp)	Merker = 0 --> Anzeige: 0 Merker = 1 --> Anzeige: 1
STRING	Merker = 0 --> Anzeige: FALSE Merker = 1 --> Anzeige: TRUE

Beispiel:

```
<JC:DTAG name="F100" type="STRING" format="#" />
```

Ergebnis:

Der Zustand des Merkers 100 wird als String "T" oder "F" angezeigt.

Eingänge

Der Variablenname beginnt mit einem großen "I" gefolgt von der Eingangsnummer.

Folgende Typen sind möglich:

Typ	Beschreibung
BOOL (Standardtyp)	Eingang = 0 --> Anzeige: 0 Eingang = 1 --> Anzeige: 1
STRING	Eingang = 0 --> Anzeige: OFF Eingang = 1 --> Anzeige: ON

Beispiel:

```
<JC:DTAG name="I201200308" type="STRING" />
```

Ergebnis:

Der Zustand des Eingangs 201200308 wird als String "ON" oder "OFF" angezeigt.

Ausgänge

Der Variablenname beginnt mit einem großen "O" gefolgt von der Ausgangsnummer.

Folgende Typen sind möglich:

Typ	Beschreibung
BOOL (Standardtyp)	Ausgang = 0 --> Anzeige: 0 Ausgang = 1 --> Anzeige: 1
STRING	Ausgang = 0 --> Anzeige: OFF Ausgang = 1 --> Anzeige: ON

Beispiel:

```
<JC:DTAG name="O201100308" />
```

Ergebnis:

Der Zustand des Ausgangs 201100308 wird als "1" oder "0" eingefügt.

Zugriff über Zeigerregister

Zugriff über ein Zeigerregister ist möglich durch Einfügen eines großen "P" vor dem Variablennamen. Es wird jeweils der Wert der Variablen angezeigt, deren Nummer dem Inhalt des im Variablennamen spezifizierten Registers entspricht.

Beispiele:

```
<JC:DTAG name="PR1000300" />
```

Ergebnis:
Der Inhalt des Registers, dessen Nummer im Register 1000300 enthalten ist, wird angezeigt.

```
<JC:DTAG name="PF1000300" />
```

Ergebnis:
Der Zustand des Merkers, dessen Nummer im Register 1000300 enthalten ist, wird angezeigt.

```
<JC:DTAG name="PI1000300" />
```

Ergebnis:
Der Zustand des Eingangs, dessen Nummer im Register 1000300 enthalten ist, wird angezeigt.

```
<JC:DTAG name="PO1000300" />
```

Ergebnis:
Der Zustand des Ausgangs, dessen Nummer im Register 1000300 enthalten ist, wird angezeigt.

Zugriff über Zeigerregister und Offset

Zusätzlich zum Wert aus dem Zeigerregister kann noch ein konstanter Wert oder ein weiterer Registerinhalt addiert werden, um die Nummer der anzuzeigenden Variablen zu bilden.

Beispiele:

```
<JC:DTAG name="PR1000300 + 100" />
```

Ergebnis:
Der Inhalt des Registers, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem Wert 100 ergibt, wird angezeigt.

```
<JC:DTAG name="PR1000300 + R1000100" />
```

Ergebnis:
Der Inhalt des Registers, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem Inhalt von Register 1000100 ergibt, wird angezeigt.

```
<JC:DTAG name="PF1000300 + 100" />
```

Ergebnis:
Der Zustand des Merkers, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem 100 ergibt, wird angezeigt.

```
<JC:DTAG name="PF1000300 + R1000100" />
```

Ergebnis:
Der Zustand des Merkers, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem Inhalt von Register 1000100 ergibt, wird angezeigt.

```
<JC:DTAG name="PI1000300 + 100" />
```

Ergebnis:
Der Zustand des Eingangs, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem Wert 100 ergibt, wird angezeigt.

```
<JC:DTAG name="PI1000300 + R1000100" />
```

Ergebnis:
Der Zustand des Eingangs, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem Inhalt von Register 1000100 ergibt, wird angezeigt.

```
<JC:DTAG name="PO1000300 + 100" />
```

Ergebnis:
Der Zustand des Ausgangs, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem Wert 100 ergibt, wird angezeigt.

```
<JC:DTAG name="PO1000300 + R1000100" />
```

Ergebnis:
Der Zustand des Ausgangs, dessen Nummer sich aus der Addition des Inhalts von Register 1000300 und dem Inhalt von Register 1000100 ergibt, wird angezeigt.

4.1.3 Beispiel einer HTML-Seite

Aufgabe

- Binden Sie aktuelle Echtzeit-Steuerungswerte in eine HTML-Seite ein.

Die HTML-Seite soll dann mit Hilfe der Funktionalität *Server Side Includes* mit einem Browser darstellbar sein.

Vorgehen

```
<JC:DTAG xmlns:JC="http://bucherautomation.com/ssi/jetcontrol" />
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgID" content="FrontPage.Editor.Document">
<title>Index</title>
</head>
<body>
Hallo Welt, 
<p>So können aktuelle Steuerungswerte in eine html-Seite eingebunden
werden: </p>
<p>Register 201000 = <JC:DTAG name="R201000" type = INT format="+####" />,
oder Hex: 0x<JC:DTAG name="R201000" type="INTX" format="0###" />,
oder lieber so <JC:DTAG name="R201000" type="BOOL" />, wenn man nur boolsch
abfragt.
Es geht aber auch binär: <JC:DTAG name="R201000" type="INTB"
format=######## />b. </p>
```

```
<p>Strings könnte man auch definieren "<JC:DTAG name="R201000"
type="STRING" />". </p>
<p>So sieht eine Realzahl aus: <JC:DTAG name="R1001500" type="REAL" />
oder so <JC:DTAG name="R1001500" type="REAL" factor="1.3" format="###.##" />
>. </p>
<p>Der Wert eines Merkers wird so dargestellt: <JC:DTAG name="F10" />
oder <JC:DTAG name="F10" type="STRING" />. </p>
<p>Bei den Ein- und Ausgängen erfolgt es analog: <JC:DTAG name="I100000205"
type="BOOL" />
bzw. <JC:DTAG name="I100000205" type="STRING" />. </p>
<p>R201000 = <JC:DTAG name="R201000" type="INT" format="+0##########" />
> </p>
<p>Viele Grüße </p>
<p>Dein JetControl</p>
</body>
</html>
```

Speicherort

- Speichern Sie die HTML-Seite im Dateisystem der Steuerung ab.

5 FTP-Client

Der FTP-Client ermöglicht den Zugriff auf Dateien und Verzeichnisse eines fernen Netzwerkteilnehmers aus dem Anwendungsprogramm heraus. Der FTP-Client kommuniziert dazu 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 Anwendungsprogramm

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

5.1 Programmierung

Der FTP-Client ermöglicht, vom Anwendungsprogramm 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:

1. FTP-Client initialisieren.
2. Verbindungen zu FTP-Servern öffnen.
3. Daten übertragen.
4. Verbindungen schließen.

Einschränkungen

Während die Steuerung gerade eine der Funktionen des FTP-Clients bearbeitet, dürfen die Tasks, die den FTP-Client bedienen, nicht durch `TaskBreak` angehalten oder durch `TaskRestart` neu gestartet werden. Sonst beendet die Steuerung diese Funktion nicht und damit werden alle weiteren Funktionsaufrufe des FTP-Client blockiert.

5.1.1 FTP-Client initialisieren

Bei jedem Anwendungsprogrammstart muss mindestens einmal der FTP-Client initialisiert werden.

Funktionsdeklaration

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

Rückgabewert

Folgender Rückgabewert ist möglich:

0	immer

Verwenden der Funktion

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

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

Funktionsweise

Die Steuerung arbeitet die Funktion in folgenden Stufen ab:

Stufe	Beschreibung
1	Die Steuerung schließt alle geöffneten Verbindungen des FTP-Client.
2	Die Steuerung initialisiert alle betriebssysteminternen Datenstrukturen des FTP-Client.

5.1.2 Verbindung zum FTP-Server öffnen

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

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
ServerAddr	IP-Adresse oder Name	Namensauflösung über DNS-Server
UserName	Benutzername für die Anmeldung	Login
PassWord	Passwort für die Anmeldung	Login

Tab. 22: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem positiven Rückgabewert konnte die Verbindung geöffnet und die Anmeldung erfolgreich durchgeführt werden. Bei einem Rückgabewert von 0 ist ein Fehler aufgetreten und die Verbindung konnte nicht geöffnet werden.

Rückgabewert	Beschreibung
>0	Ein positiver Rückgabewert muss in einer Variablen gesichert werden. Er muss als Handle beim Aufruf der folgenden Funktionen mitgegeben werden: – `Verbindung schließen` – `Datei vom FTP-Server lesen` – `Datei auf dem FTP-Server schreiben`

Rückgabewert	Beschreibung
	– Datei auf dem FTP-Server löschen – Verzeichnis auf dem FTP-Server wechseln – Verzeichnis auf dem FTP-Server anlegen – Verzeichnis auf dem FTP-Server löschen – Aktuelles Verzeichnis auf dem FTP-Server ermitteln
0	

Funktionsweise

Der Task bleibt bei der Programmzeile stehen, bis die Verbindung aufgebaut wurde oder der Timeout des FTP-Clients abgelaufen ist.

Die Funktion läuft in folgenden Stufen ab:

Stufe	Beschreibung	
1	Die Steuerung versucht eine TCP/IP-Verbindung zum FTP-Server aufzubauen.	
2	**Wenn ...**	**... dann ...**
	... der Netzwerkteilnehmer die Verbindung akzeptiert hat,	... weiter bei Stufe 3.
	... die Verbindung nicht aufgebaut werden konnte und der Timeout noch nicht abgelaufen ist,	... weiter bei Stufe 1.
	... ein Fehler aufgetreten ist oder der Timeout abgelaufen ist,	... wird die Funktion beendet und der Wert **0** zurückgeliefert.
3	Die Steuerung meldet sich mit dem Benutzernamen *Administrator* und dem Passwort *AdminPassword* am FTP-Server an.	
4	**Wenn ...**	**... dann ...**
	... der FTP-Server die Anmeldung akzeptiert hat,	... wird die Funktion beendet und ein positiver Wert als Handle zum weiteren Zugriff auf diese Verbindung zurückgeliefert.
	... der FTP-Server die Anmeldung z. B. wegen eines ungültigen Benutzernamens oder falschen Passworts nicht akzeptiert hat,	... wird die Funktion beendet und der Wert **0** zurückgeliefert.

5.1.3 Verbindung schließen

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

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion FtpConnect()

Tab. 23: Beschreibung der Funktionsparameter

Rückgabewert

Rückgabewert	Beschreibung
0	Verbindung geschlossen und gelöscht
-1	Ungültiges Handle
-2, -3	Kommunikationsfehler, z. B. FTP-Server antwortet nicht

5.1.4 Datei lesen

Mit dieser Funktion wird der Inhalt einer Datei eines fernen Netzwerkteilnehmers in eine Datei im lokalen Dateisystem der Steuerung kopiert.

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion `FtpConnect ()`
ServerFile	Dateiname	Name der Datei im Dateisystem des FTP-Servers, die die Steuerung lesen soll
ClientFile	Dateiname	Name der Datei, unter dem die Steuerung die gelesene Datei im lokalen Dateisystem speichern soll

Tab. 24: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem negativen Rückgabewert ist ein Fehler aufgetreten. Bei einem Rückgabewert von 0 hat die Steuerung die Datei gelesen und lokal gespeichert

Rückgabewert	Beschreibung
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

Funktionsweise

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

– Die Steuerung muss zuerst z. B. die Datei **ServerTestFile.txt** lesen und sie in ihrem lokalen Dateisystem z. B. unter dem Namen **LocalTestFile.txt** speichern.

– Ein Fehler ist aufgetreten.

Die Funktion läuft in folgenden Stufen ab:

Stufe	Beschreibung	
1	Die Steuerung schickt zum FTP-Server das Kommando, dass er den Inhalt der Datei **ServerTestFile.txt** übermitteln soll.	
2	Die Steuerung empfängt den Inhalt der Datei **ServerTestFile.txt**.	
3	Die Steuerung schreibt den Inhalt in die Datei **LocalTestFile.txt**.	
4	**Wenn ...**	**... dann ...**
	... keine Fehler aufgetreten sind,	... wurde die Datei erfolgreich kopiert, die Funktion beendet und der Wert **0** zurückgeliefert.
	... Fehler aufgetreten sind,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Dateinamen

– Der Funktionsparameter für die lokale Datei kann auch den Pfad zu dieser Datei enthalten, z. B. `'/Data/TestFiles/LocalTestFile.txt'`.

– Wenn das Dateisystem des entfernten Netzwerkteilnehmers dies unterstützt, kann der Funktionsparameter für die Datei auf dem FTP-Server auch den Pfad zu dieser Datei enthalten. Sonst ist zuvor mit dem Befehl `FtpDirChange()` das Verzeichnis einzustellen.

– Das Dateisystem unterstützt beide Varianten.

5.1.5 Datei schreiben

Mit dieser Funktion wird der Inhalt einer Datei im lokalen Dateisystem in eine Datei eines entfernten Netzwerkteilnehmers kopiert.

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion `FtpConnect ()`
ServerFile	Dateiname	Name der Datei, unter dem der FTP-Server die geschriebene Datei speichern soll
ClientFile	Dateiname	Name der Datei in dem lokalen Dateisystem, deren Inhalt die Steuerung an den FTP-Server senden soll

Tab. 25: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem negativen Rückgabewert ist ein Fehler aufgetreten. Bei einem Rückgabewert von 0 hat die Steuerung die Datei gelesen und auf das Dateisystem des fernen Netzwerkteilnehmers geschrieben.

Rückgabewert	Beschreibung
0	Kein Fehler
-1	Ungültiges Handle

Rückgabewert	Beschreibung
-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 des FTP-Servers, z. B. Datei kann nicht angelegt werden

Funktionsweise

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

– Die Steuerung muss z. B. zuerst die Datei **LocalTestFile.txt** lesen und sie in das Dateisystem des fernen Netzwerkteilnehmers z. B. unter dem Namen **ServerTestFile.txt** schreiben.

– Ein Fehler ist aufgetreten.

Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung	
1	Die Steuerung schickt zum FTP-Server das Kommando, dass er den Inhalt der Datei **ServerTestFile.txt** speichern soll.	
2	Die Steuerung sendet den Inhalt der Datei **LocalTestFile.txt**.	
3	Der FTP-Server schreibt den Inhalt in die Datei **ServerTestFile.txt**.	
4	**Wenn ...**	**... dann ...**
	... keine Fehler aufgetreten sind,	... wurde die Datei erfolgreich kopiert, die Funktion beendet und der Wert **0** zurückgeliefert.
	... Fehler aufgetreten sind,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Dateinamen

– Der Funktionsparameter für die lokale Datei kann auch den Pfad zu dieser Datei enthalten, z. B. '/Data/TestFiles/LocalTestFile.txt'.

– Wenn das Dateisystem des entfernten Netzwerkteilnehmers dies unterstützt, kann der Funktionsparameter für die Datei auf dem FTP-Server auch den Pfad zu dieser Datei enthalten. Sonst ist zuvor mit dem Befehl FtpDirChange() das Verzeichnis einzustellen.

– Das Dateisystem unterstützt beide Varianten.

5.1.6 Datei löschen

Mit dieser Funktion kann eine Datei im Dateisystem eines fernen Netzwerkteilnehmers gelöscht werden.

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion `FtpConnect ()`
ServerFile	Dateiname	Name der Datei, die gelöscht werden soll

Tab. 26: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem negativen Rückgabewert ist ein Fehler aufgetreten. Bei einem Rückgabewert von 0 konnte die Datei im Dateisystem des fernen Netzwerkteilnehmers gelöscht werden.

Rückgabewert	Beschreibung
0	Kein Fehler
-1	Ungültiges Handle
-2	Kommunikationsfehler, z. B. FTP-Server antwortet nicht
-3	Fehlermeldung vom FTP-Server, z. B. Datei nicht vorhanden

Funktionsweise

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

– Der FTP-Server muss zuerst die Datei **ServerTestFile.txt** löschen. Bitte beachten Sie: Der Dateiname dient nur als Beispiel.

– Ein Fehler ist aufgetreten.

Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung	
1	Die Steuerung schickt zum FTP-Server das Kommando, dass er die Datei **ServerTestFile.txt** löschen soll.	
2	Die Steuerung sendet den Inhalt der Datei **LocalTestFile.txt**.	
3	Der FTP-Server löscht die Datei **ServerTestFile.txt**.	
4	**Wenn ...**	**... dann ...**
	... keine Fehler aufgetreten sind,	... ist die Datei erfolgreich gelöscht, die Funktion beendet und der Wert **0** zurückgeliefert.
	... Fehler aufgetreten sind,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Dateinamen

– Der Funktionsparameter für die lokale Datei kann auch den Pfad zu dieser Datei enthalten, z. B. `'/Data/TestFiles/LocalTestFile.txt'`.

– Wenn das Dateisystem des entfernten Netzwerkteilnehmers dies unterstützt, kann der Funktionsparameter für die Datei auf dem FTP-Server auch den Pfad zu dieser Datei enthalten. Sonst ist zuvor mit dem Befehl `FtpDirChange()` das Verzeichnis einzustellen.

– Das Dateisystem unterstützt beide Varianten.

5.1.7 Verzeichnis wechseln

Mit dieser Funktion kann das aktuelle Verzeichnis im Dateisystem eines fernen Netzwerkteilnehmers gewechselt werden.

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion `FtpConnect ()`
ServerFile	Dateiname	Name des Verzeichnisses, in das gewechselt werden soll

Tab. 27: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem negativen Rückgabewert ist ein Fehler aufgetreten. Bei einem Rückgabewert von **0** konnte das Verzeichnis gewechselt werden.

Rückgabewert	Beschreibung
0	Kein Fehler
-1	Ungültiges Handle
-2	Kommunikationsfehler, z. B. FTP-Server antwortet nicht
-3	Fehlermeldung des FTP-Servers, z. B. Verzeichnis nicht vorhanden

Funktionsweise

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

– Der FTP-Server muss zuerst das Verzeichnis wechseln.

– Ein Fehler ist aufgetreten.

Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung	
1	Die Steuerung schickt zum FTP-Server das Kommando, dass er in ein Unterverzeichnis wechseln soll.	
2	Der FTP-Server wechselt das Verzeichnis.	
3	**Wenn ...**	**... dann ...**
	... keine Fehler aufgetreten sind,	... ist das neue Verzeichnis eingestellt, die Funktion beendet und der Wert **0** zurückgeliefert.
	... Fehler aufgetreten sind,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Verzeichnisnamen

– Wenn das Dateisystem des entfernten Netzwerkteilnehmers dies unterstützt, kann der Funktionsparameter für das Verzeichnis auf dem FTP-Server auch den kompletten Pfad - über mehrere Verzeichnisebenen - zu diesem Verzeichnis enthalten.

– Sonst muss zuvor mit dem Befehl FtpDirChange() von einer Verzeichnisebene zur nächsten verzweigt werden, bis das gewünschte Verzeichnis erreicht ist.

– Das Dateisystem unterstützt beide Varianten.

5.1.8 Verzeichnis anlegen

Mit dieser Funktion wird ein neues Verzeichnis im Dateisystem eines fernen Netzwerkteilnehmers angelegt.

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion `FtpConnect ()`
ServerDir	Verzeichnisname	Name des Verzeichnisses, das angelegt werden soll

Tab. 28: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem negativen Rückgabewert ist ein Fehler aufgetreten. Bei einem Rückgabewert von **0** konnte das Verzeichnis im Dateisystem des fernen Netzwerkteilnehmers angelegt werden.

Rückgabewert	Beschreibung
0	Kein Fehler
-1	Ungültiges Handle
-2	Kommunikationsfehler, z. B. FTP-Server antwortet nicht
-3	Fehlermeldung des FTP-Servers, z. B. Verzeichnis bereits vorhanden

Funktionsweise

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

– Der FTP-Server muss zuerst ein Unterverzeichnis anlegen.

– Ein Fehler ist aufgetreten.

Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung	
1	Die Steuerung schickt zum FTP-Server das Kommando, dass er ein Unterverzeichnis anlegen soll.	
2	Der FTP-Server legt das Verzeichnis an.	
3	**Wenn ...**	**... dann ...**
	... keine Fehler aufgetreten sind,	... ist das neue Verzeichnis eingestellt, die Funktion beendet und der Wert **0** zurückgeliefert.
	... Fehler aufgetreten sind,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Verzeichnisnamen

– Wenn das Dateisystem des entfernten Netzwerkteilnehmers dies unterstützt, kann der Funktionsparameter für das Verzeichnis auf dem FTP-Server auch den kompletten Pfad - über mehrere Verzeichnisebenen - zu diesem Verzeichnis enthalten.

– Sonst muss zuvor mit dem Befehl `FtpDirChange()` von einer Verzeichnisebene zur nächsten verzweigt werden, bis das gewünschte Verzeichnis erreicht ist.

– Das Dateisystem unterstützt beide Varianten.

Einschränkungen im Dateisystem einer JetControl

Wenn Sie als Funktionsparameter ein Verzeichnis mit einem Pfad angeben, müssen alle Verzeichnisse bis zu dem, das Sie anlegen wollen, vorhanden sein. Ein rekursives Anlegen von mehreren Verzeichnissen wird nicht unterstützt.

Beispiel:

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

Damit der Ordner **Release** im Verzeichnisbaum */DataFiles/TextFiles* angelegt werden kann, müssen diese vorhanden sein.

5.1.9 Verzeichnis löschen

Mit dieser Funktion wird ein Verzeichnis im Dateisystem eines fernen Netzwerkteilnehmers gelöscht.

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion `FtpConnect ()`
ServerDir	Verzeichnisname	Name des Verzeichnisses, das gelöscht werden soll

Tab. 29: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem negativen Rückgabewert ist ein Fehler aufgetreten. Bei einem Rückgabewert von **0** konnte das Verzeichnis im Dateisystem des fernen Netzwerkteilnehmers gelöscht werden.

Rückgabewert	Beschreibung
0	Kein Fehler
-1	Ungültiges Handle
-2	Kommunikationsfehler, z. B. FTP-Server antwortet nicht
-3	Fehlermeldung des FTP-Servers, z. B. Verzeichnis nicht vorhanden

Funktionsweise

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

– Der FTP-Server muss zuerst das Unterverzeichnis löschen.

– Ein Fehler ist aufgetreten.

Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung	
1	Die Steuerung schickt zum FTP-Server das Kommando, dass er das Unterverzeichnis löschen soll.	
2	Der FTP-Server löscht das Unterverzeichnis.	
3	**Wenn ...**	**... dann ...**
	... keine Fehler aufgetreten sind,	... ist das Verzeichnis gelöscht, die Funktion beendet und der Wert **0** zurückgeliefert.
	... Fehler aufgetreten sind,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

Verzeichnisnamen

– Wenn das Dateisystem des entfernten Netzwerkteilnehmers dies unterstützt, kann der Funktionsparameter für das Verzeichnis auf dem FTP-Server auch den kompletten Pfad - über mehrere Verzeichnisebenen - zu diesem Verzeichnis enthalten.

– Sonst muss zuvor mit dem Befehl `FtpDirChange()` von einer Verzeichnisebene zur nächsten verzweigt werden, bis das gewünschte Verzeichnis erreicht ist.

– Das Dateisystem unterstützt beide Varianten.

5.1.10 Aktuelles Verzeichnis ermitteln

Mit dieser Funktion wird das aktuell eingestellte Verzeichnis im Dateisystem eines fernen Netzwerkteilnehmers ermittelt.

Funktionsdeklaration

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

Funktionsparameter

Parameter	Beschreibung	Bemerkung
FtpConnection	Handle	Rückgabewert der Funktion `FtpConnect ()`
str	Adresse eines Strings	Aktuelles Verzeichnis mit Pfadangabe

Tab. 30: Beschreibung der Funktionsparameter

Rückgabewert

Bei einem negativen Rückgabewert ist ein Fehler aufgetreten. Bei einem Rückgabewert von **0** konnte das aktuell eingestellte Verzeichnis im Dateisystem eines fernen Netzwerkteilnehmers ermittelt werden.

Rückgabewert	Beschreibung
0	Kein Fehler
-1	Ungültiges Handle
-3	Kommunikationsfehler, z. B. FTP-Server antwortet nicht
-4	Fehlermeldung des FTP-Servers
-5	Ungültige Rückmeldung vom Server

Funktionsweise

Der Task bleibt in folgenden Fällen bei diesem Funktionsaufruf stehen:

– Der FTP-Server muss zuerst das aktuell eingestellte Verzeichnis ermitteln.

– Ein Fehler ist aufgetreten.

Die Funktion wird in folgenden Stufen abgearbeitet:

Stufe	Beschreibung	
1	Die Steuerung schickt zum FTP-Server das Kommando, dass er das aktuell eingestellte Verzeichnis ermitteln soll.	
2	Der FTP-Server übermittelt der Steuerung das aktuell eingestellte Verzeichnis mit Pfadangabe.	
3	**Wenn ...**	**... dann ...**
	... keine Fehler aufgetreten sind,	... enthält die Variable den kompletten Pfad des aktuellen Verzeichnisses, die Funktion wird beendet und der Wert **0** wird zurückgeliefert.
	... Fehler aufgetreten sind,	... wird die Funktion beendet und ein negativer Wert zurückgeliefert.

5.2 Register

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

5.2.1 Registernummern

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

Registernummern

Basisregisternummer	Registernummern
320000	320000 ... 320101

Registernummer ermitteln

In diesem Kapitel sind jeweils nur die letzten drei Ziffern der Registernummer angegeben, z. B. MR 002. Addieren Sie zu dieser Modulregisternummer die Basisregisternummer des jeweiligen Geräts, z. B. 320000, um die vollständige Registernummer zu ermitteln.

Registerübersicht

Register	Register
MR 000	Anzahl offener Verbindungen
MR 002	Timeout in Sekunden
MR 003	Port-Nummer des FTP-Servers
MR 004	Index in der Verbindungstabelle
MR 005	Handle der Verbindung
MR 006	IP-Adresse des FTP-Servers
MR 007	Port-Nummer des FTP-Servers
MR 008	IP-Adresse des FTP-Clients
MR 009	Port-Nummer des FTP-Clients
MR 100	Status der FTP-Client-Bearbeitung
MR 101	Task-ID

Tab. 31: Übersicht über die Modulregister des FTP-Clients

5.2.2 Registerbeschreibung

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 zurzeit geöffnet sind.

Modulregistereigenschaften

Werte lesen	0 ... 2.147.483.647	Anzahl Verbindungen

MR 002

Timeout

In diesem Register stellen Sie den Timeout des FTP-Clients beim Zugriff auf den FTP-Server ein.

Modulregistereigenschaften

Werte	0 ... 2.147.483.647	in Sekunden
Wert nach Reset	20	

MR 003

Port-Nummer des FTP-Servers

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

Modulregistereigenschaften

Werte	0 ... 65.535
Wert nach Reset	21

MR 004

Index in der Verbindungstabelle

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

Modulregistereigenschaften

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, können Sie anschließend den Index der Verbindungstabelle aus Modulregister MR 004 und die Verbindungsdaten aus den Modulregistern MR 006 bis MR 009 lesen.

Modulregistereigenschaften

Werte	0 ... 2.147.483.647

MR 006

IP-Adresse des FTP-Servers

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

Modulregistereigenschaften

Zugriff	Lesen
Wird wirksam	wenn MR 004 ≥ 0

MR 007

Port-Nummer des FTP-Servers

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

Modulregistereigenschaften

Zugriff	Lesen
Wird wirksam	wenn MR 004 ≥ 0

MR 008

IP-Adresse des FTP-Clients

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

Modulregistereigenschaften

Zugriff	Lesen
Wird wirksam	wenn MR 004 ≥ 0

MR 009

Port-Nummer des FTP-Clients

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

Modulregistereigenschaften

Zugriff	Lesen
Wird wirksam	wenn MR 004 ≥ 0

MR 100

Status der FTP-Client-Bearbeitung

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

Modulregistereigenschaften

Werte	0	Es wird kein Zugriff ausgeführt
	1	Übergabe der Parameter an den FTP-Client der Steuerung
	2	Der FTP-Client kommuniziert mit dem FTP-Server
	3	Der Zugriff ist abgeschlossen
Zugriff	Lesen	

MR 101

Task-ID

Der Wert in diesem Register zeigt die ID des Tasks an, der gerade eine Funktion des FTP-Clients ausführt.

Modulregistereigenschaften

Werte	0 ... 99	Task-ID
	255	Kein Task führt gerade eine FTP-Funktion aus
Wert nach Reset	255	
Zugriff	Lesen	

6 AutoCopy - Automatisches Kopieren von Steuerungsdaten

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

Funktionen im lokalen Dateisystem

Folgende Funktionen führt die AutoCopy-Funktion aus:

– 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 führt die AutoCopy-Funktion aus:

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

Anwendungsbereiche

Für die AutoCopy-Funktion gibt es hauptsächlich folgende Anwendungsbereiche:

– Wo Fernwartung nicht möglich ist.
– Kein PC vor Ort vorhanden ist.
– Der Anlagenbediener nicht in der Lage ist oder nicht in die Lage versetzt werden soll, Änderungen an der Anlage vorzunehmen.

Folgendes kann mit der AutoCopy-Funktion ausgeführt werden:

– Änderung des Anwendungsprogramms
– Änderung der Anwendungsdaten
– Änderung der Steuerungskonfiguration
– Betriebssystemupdate (Steuerung, Module am Systembus, Netzwerkteilnehmer)

Voraussetzungen

Für das automatische Kopieren von Steuerungsdaten müssen folgende Voraussetzungen erfüllt sein:

– Der Programmierer verfügt über Kenntnisse des Dateisystems.
– Der Programmierer verfügt über Grundkenntnisse der Anwendung von FTP.

config.ini – Beispiel

Das ist ein Beispiel für eine Konfigurationsdatei **config.ini** mit einem Eintrag *AutoCopyIni*.

```
[IP]
Address = 192.168. 10.209
SubnetMask = 255.255.255. 0
DefGateway = 0. 0. 0. 0
DNSServer = 192.168. 10.244

[HOSTNAME]
SuffixType = 0
Name = JetControl350

[PORTS]
JetIPBase = 50000
JVMDebug = 52000

[FILES]
AutoCopyIni = /SD/project_name/autocopy.ini
```

AutoCopyIni – Hinweise

– Die AutoCopy-Funktion hat nur Sinn, wenn die zu kopierenden Daten auf der SD-Karte oder dem USB-Stick abgespeichert sind. Das Stammverzeichnis ist dann zwangsläufig je nach Steuerung */SD/, /USB1/* oder */USB2/.*
  – Die Zuordnung zwischen dem Steckplatz des USB-Sticks und dem Stammverzeichnis ist beschrieben bei der USB-Schnittstelle der jeweiligen Betriebsanleitung.
  – Hat die Steuerung nur einen USB-Steckplatz ist das Stammverzeichnis */USB/.*
– Die Datei **autocopy.ini** kann in einem beliebigen Verzeichnis sein.
– Statt **autocopy.ini** können Sie die Datei beliebig benennen.

Bezeichnung

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

6.1 Funktionsweise

Dieses Kapitel beschreibt das Starten und Ausführen der AutoCopy-Funktion.

6.1.1 AutoCopy-Funktion starten

Die AutoCopy-Funktion kann nur während der Boot-Phase (nach dem Einschalten) der Steuerung ausgeführt werden.

Voraussetzungen

– Sie haben die Kommandodatei erstellt und im entsprechenden Verzeichnis abgelegt.

Wenn der Eintrag AutoCopyIni in der Konfigurationsdatei **config.ini** nicht vorhanden ist, ist der Name der Kommandodatei und das Verzeichnis wie folgt vorgeschrieben:

Parameter	Wert	Beschreibung
Dateiname	autocopy.ini	Alles Kleinbuchstaben
Verzeichnis	/SD/	Stammverzeichnis auf der SD-Karte
Verzeichnis	/USB/ oder /USB1/	Stammverzeichnis auf dem USB-Stick

– Die Datei autocopy.ini kann in einem beliebigen Verzeichnis sein.
– Statt autocopy.ini können Sie die Datei beliebig benennen.

In diesem Fall ist die Bedingung, dass die Konfigurationsdatei **config.ini** den Eintrag *AutoCopyIni* enthält. Dieser Eintrag definiert dann den Verzeichnis- und Dateinamen der Kommandodatei.

AutoCopy-Funktion starten

Um die AutoCopy-Funktion zu starten, führen Sie folgende Schritte aus:

1. Schalten Sie die Steuerung aus.
2. **Wenn** Sie eine **SD-Karte** verwenden, stecken Sie die SD-Karte vollständig in den Steckplatz der SD-Karte.
   Wenn Sie einen **USB-Stick** verwenden, stecken Sie den USB-Stick in die USB-Buchse der Steuerung.
3. Stellen Sie den Schalter in Stellung *LOAD.*
4. Schalten Sie die Steuerung ein.
5. **Wenn** JC-3xx, warten Sie bis die rote LED **D1** an ist, sowie die grüne LED **R** und die gelbe LED **SD** langsam mit ca. 1 Hz blinken.

   Wenn JC-4xx, warten Sie bis die rote LED **D1** an ist, sowie die grüne LED **R** und die gelbe LED  langsam mit ca. 1 Hz blinken.
   Wenn JC-94xMC, warten Sie bis die gelbe LED **D1** an ist und die grüne LED **RUN** langsam mit ca. 1 Hz blinkt.
   Wenn JC-97xMC, warten Sie bis die gelbe LED **GP2** an ist und die grüne LED **GP1** langsam mit ca. 1 Hz blinkt.

   ⇨ Die Steuerung führt die AutoCopy-Funktion aus.
6. **Wenn** JC-3xx, warten Sie bis die rote LED **D1** und die grüne LED **R** langsam mit ca. 1 Hz blinken.
   Wenn JC-4xx, warten Sie bis die rote LED **D1** und die grüne LED **R** langsam mit ca. 1 Hz blinken.
   Wenn JC-94xMC, warten Sie bis die gelbe LED **D1** an ist und die grüne LED **RUN** schnell mit ca. 4 Hz blinkt.
   Wenn JC-97xMC, warten Sie bis die gelbe LED **GP2** an ist und die grüne LED **GP1** schnell mit ca. 4 Hz blinkt.

⇨ Die AutoCopy-Funktion ist beendet.

6.1.2 AutoCopy-Funktion ausführen

Während der Boot-Phase in der Betriebsart AutoCopy führt die Steuerung die Kommandos in der Kommandodatei aus.

Einschränkungen

In der Betriebsart AutoCopy gelten folgende Einschränkungen der Steuerungsfunktion:

– Die Steuerung führt das Anwendungsprogramm nicht aus.
– Die Kommunikation mit der Steuerung ist nicht möglich.
– Nach Beendigung der AutoCopy-Funktion ist ein Neustart der Steuerung erforderlich.

AutoCopy-Funktion ausführen

Stufe	Beschreibung
1	Die Steuerung öffnet die Kommandodatei, die im Eintrag *AutoCopyIni* in der Konfigurationsdatei **/System/config.ini** angegeben ist.
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 ggf. 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 JC-3xx und JC-4xx in der Betriebsart AutoCopy

Stufe	LEDs					Zustand
	R	E	D1	D2	SD/ 	
1	☼ 4 Hz	☼ 4 Hz	☼ 4 Hz	☼ 4 Hz	○ OFF	Reset
2	☼ 1 Hz	○ OFF	○ OFF	● ON	○ OFF	Der Bootloader lädt und prüft das Betriebssystem. Bei Steuerungen ohne Bootloader: Die Steuerung initialisiert das Betriebssystem.
3	☼ 1 Hz	○ OFF	○ OFF	○ OFF	○ OFF	Das Betriebssystem liest den DIP-Schalter auf dem Backplane-Modul und prüft das Vorhandensein des Ethernet-Switches.
4	☼ 1 Hz	● ON	○ OFF	○ OFF	○ OFF	Das Betriebssystem initialisiert die Echtzeituhr, Ethernet-Schnittstelle und das Dateisystem.
5	☼ 1 Hz	● ON	● ON	○ OFF	☼ / ○ OFF	Das Betriebssystem initialisiert die Module am JX3- und JX2-Systembus und die SD-Karte.
6	☼ 1 Hz	○ OFF	● ON	○ OFF	☼	Die Kommandodatei der AutoCopy-Funktion wird bearbeitet.
7a	☼ 1 Hz	○ OFF	☼ 1 Hz	○ OFF	○ OFF	Die AutoCopy-Funktion ist beendet; Fehler sind keine aufgetreten.
7b	☼ 1 Hz	● ON	☼ 1 Hz	○ OFF	○ OFF	Die AutoCopy-Funktion ist beendet; Fehler sind aufgetreten.

LEDs der JC-94xMC in der Betriebsart AutoCopy

Stufe	LEDs			Zustand
	ERR	D1	RUN	
1	● ON	○ OFF	○ OFF	Reset
2	○ OFF	● ON	○ OFF	Initialisierung
3	○ OFF	● ON	☼ 1 Hz	AutoCopy-Funktion wird ausgeführt.
4a	○ OFF	● ON	☼ 4 Hz	Die AutoCopy-Funktion ist beendet; es sind keine Fehler aufgetreten.
4b	● ON	● ON	☼ 4 Hz	Die AutoCopy-Funktion ist beendet; Fehler sind aufgetreten.

LEDs der JC-97xMC in der Betriebsart AutoCopy

Stufe	LEDs			Zustand
	ERR	D1	RUN	
1	● ON	○ OFF	○ OFF	Reset
2	○ OFF	● ON	○ OFF	Initialisierung
3	○ OFF	● ON	☼ 1 Hz	AutoCopy-Funktion wird ausgeführt.
4a	○ OFF	● ON	☼ 4 Hz	Die AutoCopy-Funktion ist beendet; es sind keine Fehler aufgetreten.
4b	● ON	● ON	☼ 4 Hz	Die AutoCopy-Funktion ist beendet; Fehler sind aufgetreten.

6.1.3 AutoCopy-Funktion beenden

Nur ein Neustart der Steuerung beendet die AutoCopy-Funktion.

Voraussetzung

Die Bearbeitung der AutoCopy-Kommandos ist beendet.

AutoCopy-Funktion beenden

Um die AutoCopy-Funktion zu beenden, führen Sie folgende Schritte aus:

1. Schalten Sie die Steuerung aus.
2. Sie können jetzt die SD-Karte oder den USB-Stick entfernen (nicht erforderlich).
3. Stellen Sie den Schalter in Stellung RUN oder STOP.
4. Schalten Sie die Steuerung ein.

⇨ Die Steuerung bootet neu.

6.2 autocopy.ini - Aufbau

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 setzen Sie Werte, mit denen die AutoCopy-Funktion arbeitet.
– Leerzeilen können Sie beliebig einfügen.
– Leiten Sie Kommentarzeilen mit "!", "#" oder ";" ein.

Sektionen

Die Kommandodatei enthält zwei Sektionstypen:

– In der Sektion *[OPTIONS]* legen Sie grundsätzliche Einstellungen fest. Diese Sektion ist nur einmal vorhanden.
– In den Sektionen *[COMMAND_#]* geben Sie die auszuführenden Kommandos an. Die Anzahl der Kommandosektionen ist auf 128 beschränkt.

6.2.1 Die Sektion [OPTIONS]

In der Sektion `[OPTIONS]` legen Sie grundsätzliche Einstellungen fest. Diese Sektion 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 Elementen:

CommandCount

Eigenschaft	Beschreibung
Im Beispiel	14
Funktion	Anzahl der folgenden Kommandosektionen
Gültige Werte	≥ 0
Nicht gültige Werte	< 0
Bei ungültigem Wert oder nicht vorhandenem Eintrag	0

LogFile

Eigenschaft	Beschreibung
Im Beispiel	/SD/autocopy.log
Funktion	Vollständiger Name der Protokolldatei
Gültige Werte	– Alle gültigen Dateinamen – Verzeichnis ist vorhanden
Nicht gültige Werte	– Ungültiger Dateiname – Nicht vorhandenes Verzeichnis
Bei ungültigem Wert oder nicht vorhandenem Eintrag	Die Steuerung erstellt keine Protokolldatei.

LogAppend

Eigenschaft	Beschreibung
Im Beispiel	1
Funktion	Legt fest, ob eine neue Protokolldatei erstellt oder an eine vorhandene angefügt wird.
Gültige Werte	– 0 = ggf. bestehende Datei löschen und neue Datei erstellen – 1 = an bestehende Datei anfügen; wenn keine Datei vorhanden ist, erstellt die Steuerung eine neue Protokolldatei.
Nicht gültige Werte	– < 0 – > 1
Bei ungültigem Wert oder nicht vorhandenem Eintrag	Die Steuerung erstellt die Protokolldatei neu.

6.2.2 Die Kommandosektionen

In diesen Sektionen geben Sie die Befehle an, die die Funktion AutoCopy ausführt.

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 bilden sich aus der Zeichenkette `COMMAND_` gefolgt von einer Zahl. Die Zahl liegt zwischen eins und dem Wert des Eintrags `CommandCount` aus der Sektion `[OPTIONS]`.

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 darf nur ein Kommando enthalten. Somit müssen Sie für jedes Kommando eine eigene Sektion anlegen.

Fehlerbehandlung

Wenn bei der Bearbeitung eines Kommandos ein Fehler auftritt, legt die Steuerung einen entsprechenden Eintrag in der Protokolldatei an. Für jedes Kommando können Sie einstellen, ob die Steuerung den Fehler als `Error` oder als `Warning` in die Protokolldatei einträgt. Nehmen Sie die Einstellung über den optionalen Parameter `ErrorAsWarning` vor.

ErrorAsWarning	Eintrag in die 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'`.

– Wenn das Dateisystem dies unterstützt, kann der Funktionsparameter für die Datei auf dem FTP-Server auch den Pfad zu dieser Datei enthalten. Sonst ist zuvor mit dem Befehl `FtpDirChange()` das Verzeichnis einzustellen.

– Das Dateisystem unterstützt beide Varianten.

Verfügbare Kommandos im lokalen Dateisystem

Command = DirCreate

Eigenschaft	Beschreibung
Funktion	Legt Unterverzeichnis an
Parametername	Path
Parameterwert	Vollständiger Verzeichnisname
Gültige Werte	– Alle gültigen Verzeichnisnamen – Übergeordnete Verzeichnisse sind vorhanden
Nicht gültige Werte	– Ungültiger Verzeichnisname – Nicht vorhandenes übergeordnetes Verzeichnis – Name eines vorhandenen Verzeichnisses
Bei ungültigem Wert	Die Steuerung legt das Verzeichnis nicht an. Sie trägt den Fehler in die Protokolldatei ein.

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

Command = DirRemove

Eigenschaft	Beschreibung
Funktion	Löscht ein Unterverzeichnis
Parametername	Path
Parameterwert	Vollständiger Verzeichnisname
Gültige Werte	– Alle gültigen Verzeichnisnamen – Das Verzeichnis ist leer
Nicht gültige Werte	– Ungültiger Verzeichnisname – Verzeichnis ist nicht leer
Bei ungültigem Wert	Die Steuerung löscht das Verzeichnis nicht. Sie trägt den Fehler in die Protokolldatei ein.

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

Command = FileCopy

Eigenschaft	Beschreibung
Funktion	Kopieren einer Datei
Parametername 1	Source
Parameterwert 1	Vollständiger Name der Quelldatei
Parametername 2	Destination
Parameterwert 2	Vollständiger Name der Zieldatei

Eigenschaft	Beschreibung
Gültige Werte	– Alle gültigen Dateinamen – Das Zielverzeichnis ist vorhanden
Nicht gültige Werte	– Ungültiger Dateiname – Nicht vorhandene Quelldatei – Nicht vorhandenes Zielverzeichnis
Bei ungültigem Wert	Die Steuerung kopiert die Datei nicht. Sie trägt den Fehler in die Protokolldatei ein.

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

Eigenschaft	Beschreibung
Funktion	Datei löschen
Parametername	Path
Parameterwert	Vollständiger Name der Datei
Gültige Werte	Alle gültigen Dateinamen
Nicht gültige Werte	Ungültiger Dateiname
Bei ungültigem Wert	Die Steuerung löscht die Datei nicht. Sie trägt den Fehler in die Protokolldatei ein.

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

Command = DaFileRead

Eigenschaft	Beschreibung
Funktion	Übertragen von Registerwerten und Merkerzuständen von einer Datendatei zur Steuerung
Parametername	DaFile
Parameterwert	Vollständiger Name der Datendatei
Gültige Werte	Alle gültigen Dateinamen für Datendateien
Nicht gültige Werte	– Ungültiger Dateiname – Nicht vorhandene Datendatei
Bei ungültigem Wert	Die Daten werden nicht an die Steuerung übertragen. Sie trägt den Fehler in die Protokolldatei ein.

Eigenschaft	Beschreibung

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

Command = DaFileWrite

Eigenschaft	Beschreibung
Funktion	Abspeichern von Registerwerten und Merkerzuständen in einer Datendatei
Parametername 1	DaFile
Parameterwert 1	Vollständiger Name der Datendatei
Gültige Werte	– Alle gültigen Dateinamen für Datendateien – Das Zielverzeichnis ist vorhanden
Nicht gültige Werte	– Ungültiger Dateiname – Nicht vorhandenes Zielverzeichnis
Bei ungültigem Wert	Die Steuerung legt die Datendatei nicht an. Sie trägt den Fehler in die Protokolldatei ein.
Parametername 2	Append
Parameterwert 2	Legt fest, ob eine neue Datendatei erstellt oder an eine vorhandene angefügt wird
Gültige Werte	– 0 = ggf. bestehende Datendatei löschen und neue Datendatei erstellen – 1 = an bestehende Datendatei anfügen; wenn keine Datendatei vorhanden ist, erstellt die Steuerung eine neue Datendatei
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 Steuerung legt die Datendatei nicht an. Sie trägt den Fehler in die Protokolldatei ein.
Parametername 4	First
Parameterwert 4	Nummer des ersten Registers/Merkers
Gültige Werte	Alle gültigen Nummern aus dem Speicherbereich der jeweiligen Steuerung
Nicht gültige Werte	Ungültige Nummern
Bei ungültigem Wert	Die Steuerung legt die Datendatei nicht an. Sie trägt den Fehler in die Protokolldatei ein.

Eigenschaft	Beschreibung
Parametername 5	Last
Parameterwert 5	Nummer des letzten Registers/Merkers
Gültige Werte	Alle gültigen Nummern aus dem Speicherbereich der jeweiligen Steuerung, die größer oder gleich sind als der Wert bei *First*
Nicht gültige Werte	– Ungültige Nummern – Nummern kleiner als *First*
Bei ungültigem Wert	Die Steuerung speichert nur einen Wert (First) ab.

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

Command = FtpConnect

Eigenschaft	Beschreibung
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

Eigenschaft	Beschreibung
	– Nicht auflösbarer Name
Bei ungültigem Wert	Die Steuerung legt die Datendatei nicht an. Sie trägt den Fehler in die Protokolldatei ein.
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 ungültigem Wert	Die Steuerung stellt die Verbindung nicht her. Sie trägt den Fehler in die Protokolldatei ein.

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

Einschränkung	Nur eine Verbindung zu einem FTP-Server ist möglich. Wenn eine Verbindung zu einem anderen FTP-Server hergestellt wird, schließt die Steuerung zuvor die bestehende Verbindung.

Command = FtpFileRead

Eigenschaft	Beschreibung
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 Dateiname – Nicht vorhandene Quelldatei – Nicht vorhandenes Zielverzeichnis
Bei ungültigem Wert	Die Steuerung kopiert die Datei nicht. Sie trägt den Fehler in die Protokolldatei ein.

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

Command = FtpFileWrite

Eigenschaft	Beschreibung
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 Dateiname – Nicht vorhandene Quelldatei – Nicht vorhandenes Zielverzeichnis
Bei ungültigem Wert	Die Steuerung kopiert die Datei nicht. Sie trägt den Fehler in die Protokolldatei ein.

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

Command = FtpFileRemove

Eigenschaft	Beschreibung
Funktion	Datei im FTP-Server löschen
Parametername	ServerFile
Parameterwert	Vollständiger Dateiname
Gültige Werte	– Alle gültigen Dateinamen
Nicht gültige Werte	– Ungültiger Dateiname
Bei ungültigem Wert	Die Steuerung löscht die Datei nicht. Sie trägt den Fehler in die Protokolldatei ein.

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

Command = FtpDirChange

Eigenschaft	Beschreibung
Funktion	Arbeitsverzeichnis im FTP-Server wechseln
Parametername	ServerDir
Parameterwert	Vollständiger Verzeichnisname
Gültige Werte	– Alle gültigen Verzeichnisnamen
Nicht gültige Werte	– Ungültiger Verzeichnisname

Eigenschaft	Beschreibung
Bei ungültigem Wert	Die Steuerung wechselt nicht das Verzeichnis. Sie trägt den Fehler in die Protokolldatei ein.

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

Command = FtpDirCreate

Eigenschaft	Beschreibung
Funktion	Unterverzeichnis anlegen im FTP-Server
Parametername	ServerDir
Parameterwert	Vollständiger Verzeichnisname
Gültige Werte	– Alle gültigen Verzeichnisnamen – Übergeordnete Verzeichnisse sind vorhanden
Nicht gültige Werte	– Ungültiger Verzeichnisname – Nicht vorhandenes übergeordnetes Verzeichnis – Name eines vorhandenen Verzeichnisses
Bei ungültigem Wert	Die Steuerung legt das Verzeichnis nicht an.Sie trägt den Fehler in die Protokolldatei ein.

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

Einschränkung	Wenn als Funktionsparameter ein Verzeichnis mit einem Pfad angegeben wird, müssen alle Verzeichnisse bis auf das zu Erstellende vorhanden sein. Ein rekursives Anlegen von mehreren Verzeichnissen wird nicht unterstützt.

Command = FtpDirRemove

Eigenschaft	Beschreibung
Funktion	Unterverzeichnis löschen im FTP-Server
Parametername	ServerDir
Parameterwert	Vollständiger Verzeichnisname
Gültige Werte	– Alle gültigen Verzeichnisnamen – Das Verzeichnis ist leer
Nicht gültige Werte	– Ungültiger Verzeichnisname – Das Verzeichnis ist nicht leer
Bei ungültigem Wert	Die Steuerung löscht das Verzeichnis nicht. Sie trägt den Fehler in die Protokolldatei ein.

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

6.2.3 Beispiel für eine Kommandodatei

Aufgabe

Die Steuerung JetControl 340 soll hier als Beispiel dienen. Sie steuert mit verschiedenen JX3-I/O-Modulen eine bestehende Anlage. In dieser Anlage wollen Sie die Funktionen erweitern.

Hierfür sind folgende Änderungen erforderlich:

– Betriebssystemupdate für die Steuerung
– Betriebssystemupdate für ein Analogausgangsmodul
– Neues Anwendungsprogramm
– Neue Werte für einige Register

Lösung

1. Sie kopieren die benötigten Dateien auf eine SD-Karte und erstellen eine Kommandodatei für die Auto-Copy-Funktion.
2. Anschließend schicken Sie diese SD-Karte mit einer kurzen Bedienungsanleitung zum Anlagenbediener, der sie nach dem Update wieder zurückschickt.

Beispielkonfiguration

Dieses Beispiel basiert auf folgender Konfiguration:

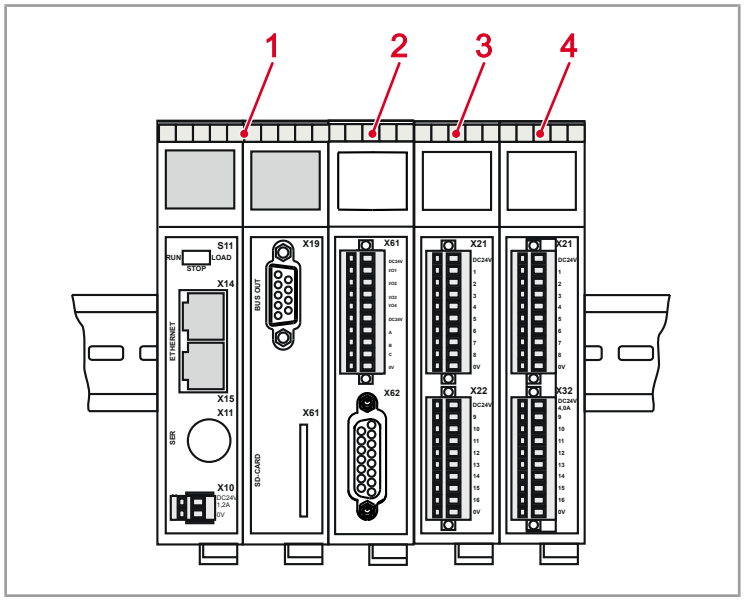


Abb. 1: Beispielkonfiguration

1	JC-340 Steuerung
2	JX3-AO4 Analogausgangsmodul Modulnummer: 2
3	JX3-DI16 Digitales Eingangsmodul
4	JX3-DO16 Digitales Ausgangsmodul

Inhalt der SD-Karte

Die folgende Abbildung zeigt die Verzeichnisstruktur und die Dateien auf der SD-Karte aus der Sicht der Steuerung vor der Ausführung der AutoCopy-Funktion.

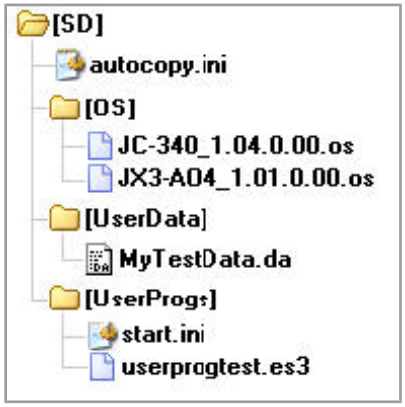


Abb. 2: Beispiel Inhalt der SD-Karte

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
```

6.3 Protokolldatei

Dieses Kapitel beschreibt Aufbau und Inhalt der Protokolldatei, in der das Gerät die Ergebnisse der jeweiligen Kommandos einträgt.

6.3.1 Dateiinhalt

Die Protokolldatei ist eine Textdatei. Durch einen Eintrag in die Kommandodatei bestimmen Sie, ob eine Protokolldatei neu erstellt oder das Gerät die Einträge an eine bestehende Protokolldatei anfügt.

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 Auto-Copy-Funktion - aus drei Elementen:

– Die Kopfzeile enthält Datum und Uhrzeit.

– Der anschließende Block gibt Auskunft über die ausgeführten Kommandos.

– Abschließend erscheint eine kurze Statistik über die Kommandobearbeitung.

Im obigen Beispiel wird beim Versuch, das Verzeichnis /app anzulegen, ein Fehler erzeugt, da das Verzeichnis bereits vorhanden ist. Diesen Fehler trägt das Gerät als Warnung ein. Wenn das Gerät die DA-Datei liest, tritt ebenfalls ein Fehler auf. Diesen Fehler trägt das Gerät in die Protokolldatei ein.

6.4 Datendateien

6.4.1 Dateiformat

Format

Die Datendatei ist wie folgt 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 – Bucher Automation 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
```

7 Anwendungsprogramm speichern und laden

Dieses Kapitel beschreibt folgendes:

– An welchem Ort speichert JetSym das Anwendungsprogramm im Gerät beim Download.
– Von welchem Ort lädt das Betriebssystem des Geräts beim Booten das Anwendungsprogramm.

Der Anwender kann zusätzlich festlegen, welches Programm ausgeführt wird.

Voraussetzungen an den Programmierer

Dieses Kapitel setzt Kenntnisse über das Erzeugen und Übertragen von Anwendungsprogrammen in JetSym und über das Dateisystem des Gerätes voraus.

7.1 Standardablage des Anwendungsprogramms

Beim Download des Anwendungsprogramms von JetSym auf das Gerät wird dieses Programm als Datei auf der internen Flash-Disk abgespeichert. Das Gerät trägt den Pfad- und Dateinamen in der Datei **start.ini** im Ordner **app** ein.

Pfad- und Dateiname

Standardmäßig legt JetSym im Ordner **app** ein Unterverzeichnis mit dem Projektnamen an und speichert dort das Anwendungsprogramm mit der Dateierweiterung *.**esX** ab. Die Pfad- und Dateinamen werden dabei immer in Kleinbuchstaben konvertiert. **X** ist ein hardwareabhängiger Platzhalter, z. B**. es4** für JetControl 400 und **es9** für JetControl 900.

start.ini – Aufbau

Die Datei ist eine reine Textdatei mit einer Sektion mit zwei Einträgen.

Parameter	Beschreibung
[Startup]	Sektionsname
Project	Pfad zur Anwendungsprogrammdatei relativ zum Ordner **app**
Program	Name der Anwendungsprogrammdatei

Beispiel	`[Startup]`
	`Project = test_program`
	`Program = test_program.es9`

Das Anwendungsprogramm wird aus der Datei **test_programm.es9** im Unterverzeichnis *test_program* im Ordner **app** geladen.

7.2 Anwendungsprogramm auf der SD-Karte oder dem USB-Stick speichern

Beim Download des Anwendungsprogramms von JetSym auf das Gerät ist standardmäßig der Speicherort die interne Flash-Disk.

– Wollen Sie, dass der Speicherort die SD-Karte oder der USB-Stick ist, müssen Sie den Speicherort konfigurieren.
– Wollen Sie, dass der Speicherort in einem anderen Verzeichnis der internen Flash-Disk als beim Standard abgespeichert werden soll, gehen Sie analog vor.

Voraussetzungen

Ausschließlich Kleinschreibung für Verzeichnis- und Dateinamen verwenden.

Anwendungsprogramm auf der SD-Karte oder dem USB-Stick speichern

Wenn Sie das Anwendungsprogramm auf der SD-Karte oder dem USB-Stick speichern wollen, konfigurieren Sie das Gerät wie folgt:

1. Erstellen Sie eine Anwendungsprogrammdatei mit JetSym.
2. Legen Sie das gewünschte Verzeichnis auf der SD-Karte oder dem USB-Stick an.
3. Legen Sie die Anwendungsprogrammdatei im gewünschten Verzeichnis ab.
4. Tragen Sie den Pfad zur Anwendungsprogrammdatei und den Programmnamen in die Datei start.ini im Ordner app auf der internen Flash-Disk des Geräts ein.

⇨ Das Gerät lädt das Anwendungsprogramm von der SD-Karte oder dem USB-Stick beim Neustart.

start.ini – Aufbau

Die Datei ist eine reine Textdatei mit einer Sektion mit zwei Einträgen.

Parameter	Beschreibung
[Startup]	Sektionsname
Project	Pfad zur Anwendungsprogrammdatei
Program	Name der Anwendungsprogrammdatei

Steuerung

Beispiel für SD-Karte:

```
[Startup]
Project = /sd/testprogram
Program = test1.esx
```

Beispiel für USB-Stick:

```
[Startup]
Project = /usb/testprogram
Program = test1.esx
```

Bediengerät

Beispiel für SD-Karte:

```
[Startup]
Project = /sd/testprogram
Program = test1.esx
```

Beispiel für USB-Stick:

```
[Startup]
Project = /usb/testprogram
Program = test1.esx
```

Bediengerät JetView

Beispiel für SD-Karte:

```
[Startup]
Project = \..\..\..\Storage Card
Program = test1.esx
```

Beispiel für USB-Stick:

```
[Startup]
Project = \..\..\..\USBMemory
Program = test1.esx
```

Ergebnis

Das Anwendungsprogramm wird aus der Datei **test1.esx** im Unterverzeichnis **testprogram** im Ordner *sd* auf der SD-Karte oder dem Ordner *usb* auf dem USB-Stick geladen.

7.3 Anwendungsprogramm laden

Beim Neustart des Anwendungsprogramms über JetSym oder nach dem Booten des Geräts wird das Anwendungsprogramm über das Dateisystem geladen und ausgeführt.

Der Ladevorgang bei der Steuerung

Dazu muss der Schalter S11 in Stellung *RUN* stehen.

So wird das Anwendungsprogramm vom Betriebssystem der Steuerung geladen:

Stufe	Beschreibung
1	Das Betriebssystem liest die Datei /**app/start.ini** von der internen Flash-Disk.
2	Das Betriebssystem wertet den Eintrag **Project** aus. Er enthält den Pfad zur Anwendungsprogrammdatei.
3	Das Betriebssystem wertet den Eintrag **Program** aus. Er enthält den Programmnamen.
4	Das Betriebssystem lädt das Anwendungsprogramm aus der Datei **<Project>/<Program>**.

Der Ladevorgang beim Bediengerät

So wird das Anwendungsprogramm vom Betriebssystem des Bediengeräts geladen:

Stufe	Beschreibung
1	Das Betriebssystem liest die Datei /**app/start.ini** von der internen Flash-Disk.
2	Das Betriebssystem wertet den Eintrag **Project** aus. Er enthält den Pfad zur Anwendungsprogrammdatei.
3	Das Betriebssystem wertet den Eintrag **Program** aus. Er enthält den Programmnamen.
4	Das Betriebssystem lädt das Anwendungsprogramm aus der Datei **<Project>/<Program>**.

Abbildungsverzeichnis

Abb. 1 Beispielkonfiguration 77
Abb. 2 Beispiel Inhalt der SD-Karte 78

Tabellenverzeichnis

Tab. 1 Systemverzeichnisse 6
Tab. 2 Bediengeräte 6
Tab. 3 Eigenschaften 11
Tab. 4 Dateien 12
Tab. 5 Begriffe 12
Tab. 6 Vordefinierte Schlüssel 14
Tab. 7 Gegenüberstellung User-Rechte 17
Tab. 8 Protokolldatei 19
Tab. 9 Elemente der Infodatei 22
Tab. 10 Zustände des Sektors 23
Tab. 11 Parameter FileName 34
Tab. 12 Rückgabewert 34
Tab. 13 Registerübersicht 35
Tab. 14 Webstatus R 202930 36
Tab. 15 IP-Adresse des SMTP-Servers R 292932 36
Tab. 16 IP-Adresse des POP3-Servers R 292933 36
Tab. 17 IP-Adresse des POP3-Servers R 292934 37
Tab. 18 Port-Nummer des POP3-Servers R 292935 37
Tab. 19 Status der E-Mail-Bearbeitung R 292937 37
Tab. 20 Task-ID (E-Mail) R 292938 38
Tab. 21 Register/Textregister 43
Tab. 22 Beschreibung der Funktionsparameter 49
Tab. 23 Beschreibung der Funktionsparameter 50
Tab. 24 Beschreibung der Funktionsparameter 51
Tab. 25 Beschreibung der Funktionsparameter 52
Tab. 26 Beschreibung der Funktionsparameter 53
Tab. 27 Beschreibung der Funktionsparameter 54
Tab. 28 Beschreibung der Funktionsparameter 55
Tab. 29 Beschreibung der Funktionsparameter 57
Tab. 30 Beschreibung der Funktionsparameter 58
Tab. 31 Übersicht über die Modulregister des FTP-Clients 59

Bucher Automation AG
Thomas-Alva-Edison-Ring 10
71672 Marbach am Neckar, Deutschland
T +49 7141 2550-0
info@bucherautomation.com

www.bucherautomation.com