



WAGO-SPEEDWAY 767

ETHERNET, 8 DI, 24 V DC

767-1301

Feldbuskoppler

Version 1.0.0

Vorwort

Es wurden alle erdenklichen Maßnahmen getroffen, um die Richtigkeit und Vollständigkeit der vorliegenden Dokumentation zu gewährleisten. Da sich Fehler trotz aller Sorgfalt nie vollständig vermeiden lassen, sind wir für Hinweise und Anregungen jederzeit dankbar. E-Mail: documentation@wago.com

Service und technischer Support

Weitere Informationen zu diesem und zu anderen Produkten (z. B. Datenblätter) erhalten Sie auf unserer Internetseite www.wago.com.

Lassen sich Störungen mit den in diesem Handbuch beschriebenen Maßnahmen nicht beseitigen, so stehen wir Ihnen gern unter folgendem Kontakt für Fragen zur Verfügung:

AUTOMATION-Support
Tel.: +49 571 887 555
Fax: +49 571 887 8555
E-Mail: support@wago.com

Zusätzliche Unterstützung

Um Sie im Umgang mit WAGO-Produkten zu unterstützen, bietet die Abteilung Seminar und Training entsprechende Seminare an. Informationen dazu erhalten Sie auf unserer Internetseite unter der Telefonnummer +49 571 887-327 oder Sie schreiben eine E-Mail an training@wago.com.

Markennamen

Wir weisen darauf hin, dass die im Handbuch verwendeten Soft- und Hardwarebezeichnungen sowie Markennamen der jeweiligen Firmen im Allgemeinen einem Warenzeichenschutz, Markenschutz oder patentrechtlichen Schutz unterliegen.

Copyright © 2010 by WAGO Kontakttechnik GmbH & Co. KG. Alle Rechte vorbehalten.

Inhaltsverzeichnis

| | |
|---|-----------|
| Vorwort | 2 |
| 1 Sicherheit | 5 |
| 1.1 Hinweise zu dieser Schnellstartanleitung..... | 5 |
| 1.2 Erläuterung der Symbole..... | 6 |
| 1.3 Gültigkeit dieser Schnellstartanleitung | 7 |
| 1.4 Bestimmungsgemäße Verwendung..... | 7 |
| 1.5 Personalqualifikation..... | 7 |
| 1.6 Grundlegende Sicherheitsinformationen..... | 8 |
| 1.7 Sicherheitseinrichtungen | 9 |
| 1.8 Technischer Zustand der 767-Komponenten | 10 |
| 1.9 Hinweise zum Betrieb | 10 |
| 2 Beschreibung | 11 |
| 2.1 Zubehör | 11 |
| 3 Anschluss | 12 |
| 4 EtherNet/IP | 16 |
| 4.1 EDS-Datei | 16 |
| 4.2 Prozessabbild..... | 17 |
| 4.3 Prozessdatenaustausch | 18 |
| 4.3.1 Assembly-Instanzen | 18 |
| 4.3.2 Ermittlung der Datenlänge | 19 |
| 4.3.3 Beispiel mit den Assembly-Instanzen 101 und 104 | 20 |
| 4.4 Diagnose..... | 22 |
| 4.4.1 Diagnose integriert in den Ein- und Ausgangsdaten | 23 |
| 4.4.2 Diagnose über explizite Nachrichten..... | 26 |
| 4.4.3 Diagnose gemischt über Ein- und Ausgangsdaten und explizite Nachrichten..... | 27 |
| 4.5 Objektmodell..... | 28 |
| 4.6 CIP-Klassen..... | 29 |
| 4.6.1 Identity Object (01 _{hex})..... | 31 |
| 4.6.2 Message Router Object (02 _{hex}) | 33 |
| 4.6.3 Assembly Object (04 _{hex}) | 35 |
| 4.6.4 Connection Object (05 _{hex})..... | 41 |
| 4.6.5 Connection Manager Object (06 _{hex})..... | 41 |
| 4.6.6 Port Class Object (F4 _{hex})..... | 42 |
| 4.6.7 TCP/IP Interface Object (F5 _{hex})..... | 44 |
| 4.6.8 Ethernet Link Object (F6 _{hex})..... | 46 |
| 4.7 WAGO-spezifische Klassen..... | 51 |

| | | |
|----------|---|-----------|
| 5 | MODBUS | 52 |
| 5.1 | Prozessabbild..... | 54 |
| 5.2 | Prozessdatenaustausch | 54 |
| 5.3 | Zugriff auf das Prozessabbild über MODBUS-Funktionen..... | 55 |
| 5.3.1 | Registerdienste | 56 |
| 5.3.2 | Bitdienste..... | 58 |
| 5.3.3 | Konfigurationsregister..... | 60 |
| 5.3.4 | Watchdog-Verhalten..... | 61 |
| 5.3.4.1 | Watchdog-Register | 62 |
| 5.3.4.2 | Diagnoseregister | 65 |
| 5.3.4.3 | Konfigurationsregister | 65 |
| 5.3.4.4 | Firmware-Informationsregister..... | 68 |
| 5.3.4.5 | Konstantenregister | 70 |
| 6 | Die Ethernet Schnittstelle..... | 72 |
| 6.1 | IP-Adressvergabe | 72 |
| 6.1.1 | Zuweisen einer festen IP-Adresse mittels DIP-Schalter | 73 |
| 6.1.2 | Zuweisen einer dynamischen IP-Adresse mittels BootP..... | 74 |
| 6.1.3 | Test der Netzwerkverbindung | 77 |
| 6.2 | Benutzerverwaltung | 77 |
| 6.3 | Das Dateisystem..... | 78 |
| 6.3.1 | Zugriff per FTP..... | 78 |
| 6.4 | Das Web-based Management (WBM) | 79 |
| 7 | WAGOframe | 80 |
| 7.1 | Installation..... | 81 |
| 7.2 | Inbetriebnahme des Feldbuskopplers..... | 86 |
| 7.3 | Bedienung..... | 87 |
| 8 | System-Update..... | 92 |
| 8.1 | Hinzufügen des System-Update-DTM..... | 93 |
| 8.2 | Verbindung zum 767-Knoten mittels Update-DTM aufbauen | 95 |
| 8.3 | Aktualisieren der 767-Komponenten | 96 |

1 Sicherheit

1.1 Hinweise zu dieser Schnellstartanleitung

Zur Installation und Inbetriebnahme des Feldbuskopplers sind zusätzlich die Systembeschreibung und das Handbuch 767-1301 zu verwenden. Ausführliche Informationen zum Feldbuskoppler erhalten Sie im Handbuch 767-1301.

WARNUNG Release-Notes beachten!

Beachten Sie, dass im SPEEDWAY-System eine Funktion nur dann **uneingeschränkt** gegeben ist, wenn alle im System eingesetzten Komponenten dem gleichen systemweiten Firmware-Release angehören. Beachten Sie daher unbedingt die entsprechenden Release-Notes zu Ihren verwendeten Produkten.

ACHTUNG Versorgungsauslegung!

Sie benötigen zu dieser Schnellstartanleitung das Handbuch „WAGO-SPEEDWAY 767, Systembeschreibung und Hinweise“, das unter www.wago.com herunterzuladen ist. Dort erhalten Sie unter anderem relevante Hinweise zur Versorgungsauslegung.

1.2 Erläuterung der Symbole

GEFAHR



Warnung vor Personenschäden

Kennzeichnet eine unmittelbare Gefährdung mit hohem Risiko, die Tod oder schwere Körperverletzung zur Folge hat, wenn sie nicht vermieden wird.

GEFAHR



Warnung vor Personenschäden durch elektrischen Strom

Kennzeichnet eine unmittelbare Gefährdung mit hohem Risiko, die Tod oder schwere Körperverletzung zur Folge hat, wenn sie nicht vermieden wird.

WARNUNG



Warnung vor Personenschäden

Kennzeichnet eine mögliche Gefährdung mit mittlerem Risiko, die Tod oder (schwere) Körperverletzung zur Folge haben kann, wenn sie nicht vermieden wird.

VORSICHT



Warnung vor Personenschäden

Kennzeichnet eine mögliche Gefährdung mit geringem Risiko, die leichte oder mittlere Körperverletzung zur Folge haben könnte, wenn sie nicht vermieden wird.

ACHTUNG



Warnung vor Sachschäden

Kennzeichnet eine mögliche Gefährdung, die Sachschaden zur Folge haben könnte, wenn sie nicht vermieden wird.

ESD



Warnung vor Sachschäden durch elektrostatische Aufladung

Kennzeichnet eine mögliche Gefährdung, die Sachschaden zur Folge haben könnte, wenn sie nicht vermieden wird.

Hinweis



Wichtiger Hinweis

Kennzeichnet eine mögliche Fehlfunktion, die aber keinen Sachschaden zur Folge hat, wenn sie nicht vermieden wird.

Information



Hinweis auf weitere Informationen

Kennzeichnet weitere Informationsquellen, die nicht Bestandteil dieser Dokumentation sind, z. B. Internet.

1.3 Gültigkeit dieser Schnellstartanleitung

Diese Schnellstartanleitung ist nur für den Feldbuskoppler 767-1301 (EtherNet/IP) der Serie WAGO-SPEEDWAY 767 gültig.

1.4 Bestimmungsgemäße Verwendung

Der Feldbuskoppler für EtherNet/IP dient zur Aufnahme und zur Verarbeitung digitaler und analoger Feldsignale. Diese können an übergeordnete Steuerungen weitergeleitet werden.

Der Feldbuskoppler darf nicht zur Übertragung von sicherheitsrelevanten Informationen genutzt werden, d. h., Not-Aus-Einrichtungen dürfen an diesem nicht betrieben werden.

Der Feldbuskoppler darf nur in Kombination mit Komponenten der Serie WAGO-SPEEDWAY 767 betrieben werden.

Der Feldbuskoppler ist für ein Arbeitsumfeld entwickelt worden, welches die Schutzklasse IP 67 (NEMA 6, 6P) erfordert.

Der Feldbuskoppler ist um maximal 64 I/O-Module der Serie WAGO-SPEEDWAY 767 erweiterbar.

Andere Anwendungen als die hier beschriebenen sind nicht zulässig.

1.5 Personalqualifikation

Sämtliche Arbeitsschritte, die an dem Feldbuskoppler durchgeführt werden, dürfen nur von Elektrofachkräften mit ausreichenden Kenntnissen im Bereich der Automatisierungstechnik vorgenommen werden. Diese müssen mit den aktuellen Normen und Richtlinien für Feldbuskoppler und Automatisierungsumfeld vertraut sein.

1.6 Grundlegende Sicherheitsinformationen

Dieses Kapitel beinhaltet eine Zusammenfassung der wichtigsten Warnhinweise. Sie wiederholen sich in einzelnen Kapiteln. Sie dienen zum Schutz Ihrer Gesundheit und vor Sachschäden an den 767-Komponenten (Feldbuskoppler und der daran angeschlossenen 767-Komponenten). Lesen und beachten Sie die nachfolgend beschriebenen Sicherheitshinweise, bevor Sie den Feldbuskoppler verwenden.

GEFAHR



Elektrische Spannung!

Betreiben Sie die 767-Komponenten ausschließlich mit 24 V DC PELV- (Protective Extra Low Voltage) oder SELV-Spannungsquellen (Safety Extra Low Voltage). Bei Verwendung von höheren Spannungen können Sie einen elektrischen Schlag bekommen.

VORSICHT



Heiße Anschlussbuchsen!

Auch unter Beachtung des Deratings können während des Betriebs hohe Oberflächentemperaturen an den metallischen Anschlussbuchsen und am Gehäuse auftreten. War die 767-Komponente in Betrieb, lassen Sie diese abkühlen, bevor Sie diese berühren.

ACHTUNG



Höchste Strombelastbarkeit der Versorgungskontakte ist 4 A!

Beachten Sie für jede 767-Komponente die maximale Strombelastbarkeit pro Versorgungslinie (U_{LS} , U_A) sowie die Gesamtstromaufnahme aller 767-Komponenten. Beide dürfen 4 A nicht überschreiten, da eine Erhöhung des Stroms zur Überhitzung der Kontakte und zu Schäden an den 767-Komponenten führt. Angaben zum Strombedarf jeder 767-Komponente finden Sie im dazugehörigen Datenblatt, das Sie unter www.wago.com herunterladen können.

ACHTUNG



Offene Anschlüsse!

Bei nicht durch Schutzkappen verschlossenen Anschlüssen können Flüssigkeiten oder Schmutz in den Feldbuskoppler eindringen und diesen zerstören. Verschließen Sie alle nicht benötigten Anschlüsse mit Schutzkappen, um die Schutzart IP 67 einzuhalten.

- Schalten Sie die Anlage spannungsfrei, an der Sie die 767-Komponenten montieren wollen.
- Halten Sie die Abdeckklappe des DIP-Schalters stets geschlossen.
- Beachten Sie bei der Montage, Inbetriebnahme, Wartung und Störbehebung die für Ihre Anlage zutreffenden Unfallverhütungsvorschriften. Beispielsweise die BGV A 3, "Elektrische Anlagen und Betriebsmittel".
- Die Betriebsanleitungen der verwendeten Komponenten der Serie 767 müssen am Arbeitsplatz bereitliegen.

- Achten Sie auf die exakte Positionierung (Codierung) zwischen Stecker und Buchse.
- Die 767-Komponenten dürfen nicht mit Substanzen in Kontakt kommen, die kriechende und isolierende Eigenschaften besitzen. Andernfalls müssen Sie für die Geräte Zusatzmaßnahmen ergreifen wie den Einbau in ein Gehäuse, das gegen die oben genannten Substanzeigenschaften resistent ist.
- In den 767-Komponenten sind elektronische Komponenten integriert, welche die ESD-Anforderungen gemäß der IEC 61000-6-2 erfüllen. Da unter ungünstigen Umständen im Feld auch höhere Spannungen durch Aufladung auftreten können, ist vor der Durchführung von Arbeiten am 767-System die Entladung zu gewährleisten.
- Achten Sie auf die korrekte Auslegung des Potenzialausgleichs.
- Halten Sie mit sämtlichen Kabeln genügend Abstand zu elektromagnetischen Störquellen ein, um eine hohe Störfestigkeit des 767-Systems gegen elektromagnetische Störstrahlungen zu erzielen. Verwenden Sie an den erforderlichen Stellen ausschließlich geschirmte Kabel. Beachten Sie dazu die entsprechenden Normen für EMV-gerechte Installationen.
- Benutzen Sie für die Weiterleitung der Versorgungsspannung und für den S-BUS ausschließlich die vorkonfektionierten WAGO-Systemkabel. Nur damit werden die angegebenen Kennwerte der technischen Daten erreicht.
- Tauschen Sie defekte oder beschädigte 767-Komponenten (z. B. bei deformierten Anschlüssen) aus, da es andernfalls in betroffenen Feldbusstationen bzw. -knoten zu Funktionsstörungen kommen kann.
- Achten Sie beim Verlegen sämtlicher Kabel darauf, dass Sie diese nicht in Scherbereichen von beweglichen Anlagenteilen verlegen.
- Beachten Sie für jede Tätigkeit die entsprechende Personenqualifikation im Kapitel 1.5.
- Beachten Sie die Bedruckung auf der Vorder- und Rückseite der 767-Komponenten.

1.7 Sicherheitseinrichtungen

Alle Produkte der Serie 767 sind nach der Schutzklasse IP 67 ausgelegt. Unter anderem besteht daraus ein vollständiger Berührungsschutz vor elektrischen Spannungen und Strömen – auch bei Nässe.

1.8 Technischer Zustand der 767-Komponenten

Bei jeglicher Änderung an den 767-Komponenten sowie an der Soft- und Firmware erlöschen ohne schriftliche Genehmigung von WAGO Kontakttechnik GmbH & Co. KG alle Haftungsansprüche.

1.9 Hinweise zum Betrieb

Zur Einbindung der 767-Komponenten in Ihre Maschine oder Anlage sind bei allen Tätigkeiten die jeweils gültigen und anwendbaren Normen, Vorschriften und Richtlinien zu beachten: beispielsweise die BGV A 3, „Elektrische Anlagen und Betriebsmittel“, DIN EN 418, EN60204. Die Not-Aus-Einrichtungen müssen in allen Betriebsarten der Anlage und Maschine wirksam bleiben.

Zum Schutz vor elektromagnetischen Störungen

- schließen Sie Ihre Anlage an Schutz Erde (PE) an und
- stellen Sie sicher, dass die Kabelführung und die Installation der Feldbus-, S-BUS, Versorgungs- und Sensorkabel korrekt sind.

Folgende Maßnahmen zur 24-V-Versorgung müssen vorhanden sein:

- äußerer Blitzschutz an Gebäuden
- innerer Blitzschutz der Versorgungs- und Signalleitungen
- sichere elektrische Trennung der Kleinspannung 24 V DC durch PELV-Spannungsquellen (Protective Extra Low Voltage) oder SELV-Spannungsquellen (Safety Extra Low Voltage)

2 Beschreibung

Diese Schnellstartanleitung beschreibt die wichtigsten Schritte zur Inbetriebnahme des Feldbuskopplers mit einem Digitalausgangsmodule.

2.1 Zubehör

Folgende Soft- und Hardware wurde bei der Erstellung des Dokumentes verwendet:

| Name | Beschreibung | Best.-Nr./Version |
|---|---|-----------------------|
| WAGOframe CD-ROM Ver. 3.0.0 | FDT/DTM-Rahmenapplikation | 759-370 Ver. 1.0.8 |
| WAGO-USB-Treiber | USB-Treiber für die Serie 767 | 759-922 Ver. 1.3.2 |
| WAGO-Service- Interface-DTM | Kommunikations-DTM | 759-371 Ver. 2.1.0 |
| WAGO-DTM für Feldbuskoppler und I/O-Module | Geräte-DTM für die (programmierbaren) Feldbuskoppler und für die I/O-Module der Serie 767 | 759-361 Ver. 2.1.0 |
| System-Update-DTM | DTM zur Firmwareaktualisierung | 759-362 Ver. 1.0.0 |
| Feldbuskoppler für EtherNet/IP | FBK, 8DI, 24 V DC (8xM8) | 767-1301 |
| Digitalausgangsmodule | 8DO, 24V DC 0,5A (8xM8) | 767-4801 |
| USB- Kommunikationskabel (WAGOframe) | M8-Stecker, axial | 756-4101/0042-0030 |
| Ethernet-Kabel (D-codiert), einseitig konfektioniert | M12-Stecker, winklig, 2 m | 756-1202/0060-0200 |
| Ethernet-Stecker | RJ-45, IP 20 | 750-975 |
| S-BUS-Kabel (B- codiert), beidseitig konfektioniert | M12-Buchse, winklig, M12-Stecker, winklig, 0,2 m | 756-1306/0060-0002 |
| S-BUS- Abschlussstecker (B- codiert) | M12-Stecker, axial | 756-9409/0060-0000 |
| Versorgungskabel (A- codiert), beidseitig konfektioniert | M12-Buchse, winklig, M12-Stecker, winklig, 0,2 m | 756-3106/0040-0002 |

Berücksichtigen Sie bei der Bestellung neben der Anzahl an IO-Modulen auch ein zusätzliches Versorgungskabel für die Verbindung zwischen Netzteil und Feldbuskoppler.

3 Anschluss

ACHTUNG Offene Anschlüsse!

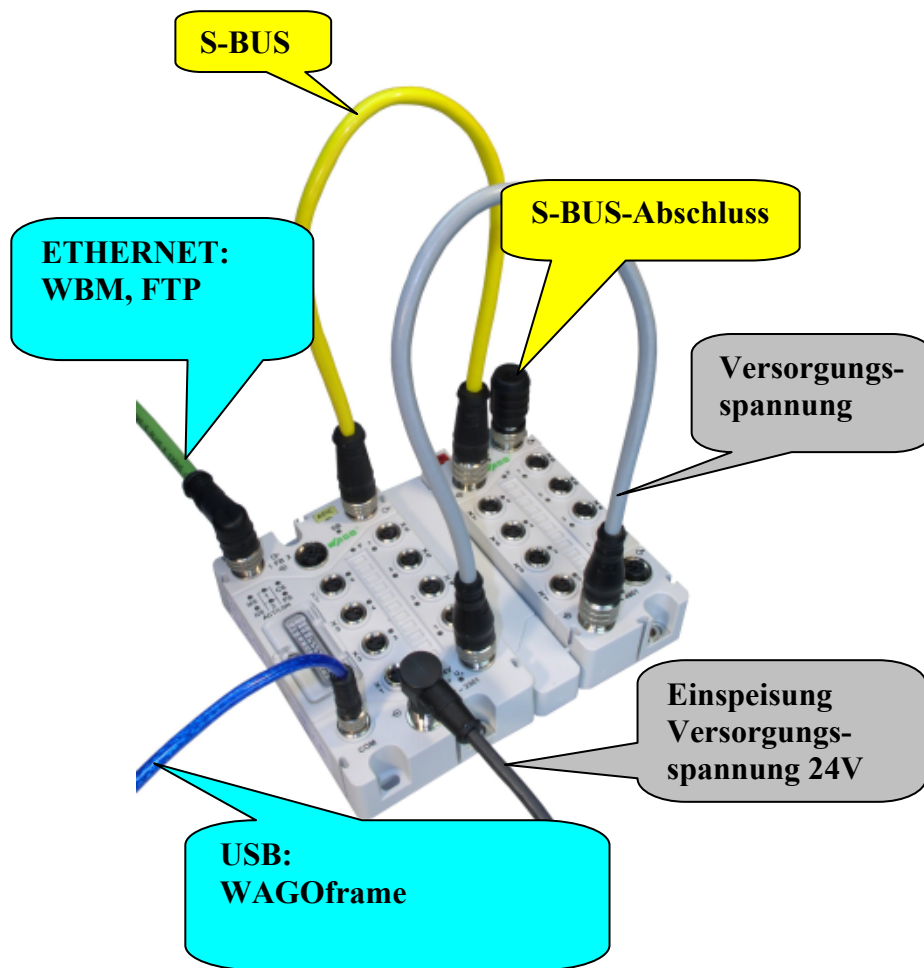


Bei nicht durch Schutzkappen verschlossenen Anschlüssen können Flüssigkeiten oder Schmutz in den Feldbuskoppler eindringen und diesen zerstören. Verschließen Sie alle nicht benötigten Anschlüsse mit Schutzkappen, um die Schutzart IP 67 einzuhalten.

Drehen Sie die Steckverbinder ausschließlich im spannungslosen Zustand mit der Hand fest. Bei Verwendung von mechanischen Hilfsmitteln können Sie die Gewinde überdrehen. In so einem Fall ist der Feldbuskoppler auszutauschen.

Anzugsmomente für die Steckverbinder: 0,6 Nm

Verdrahten Sie die 767-Komponenten entsprechend nachfolgender Darstellung:



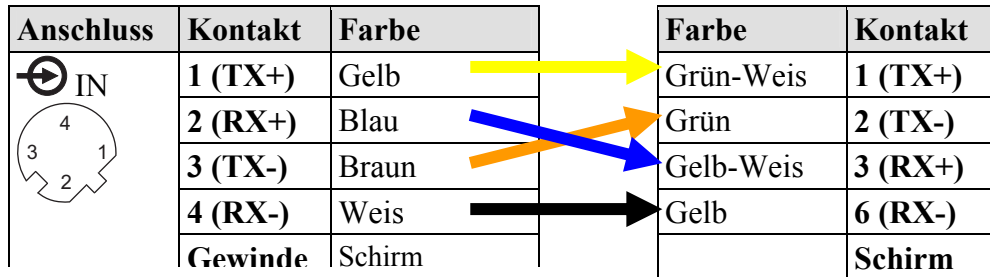
Versorgungsspannung (M12-Stecker, A-codiert, 4-polig):

Die Einspeisung der Versorgungsspannung für die Logik und für die Ausgänge erfolgt über separate Kontakte.

| Anschluss | | Kontakt | Beschreibung |
|-----------|--|---------|------------------|
| | | 1 | 24 V DC U_{LS} |
| | | 2 | 24 V DC U_A |
| | | 3 | 0 V U_{LS} |
| | | 4 | 0 V U_A |

Ethernet:

Um ein Patchkabel zum Betrieb an einem Switch oder Hub herzustellen, verbinden Sie das einseitig konfektionierte Ethernet-Kabel wie folgt:



Um ein Cross-over-Kabel zum Betrieb an einem PC herzustellen, verbinden Sie TX mit RX.

S-BUS:

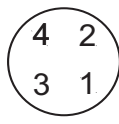
Der S-BUS dient zur internen Kommunikation zwischen dem Feldbuskoppler und den daran angeschlossenen IO-Modulen der Serie 767.

| Anschluss | Kontakt | Beschreibung |
|---|------------------|--------------|
|  | 1 | TD+ |
| | 2 | TD- |
| | 3 | RD- |
| | 4 | RD+ |
| | 5 | 0 V DC |
| | Anschlussgewinde | Geschirmt |

Service-Schnittstelle (USB):

Auf der Service-Schnittstelle stellt der Feldbuskoppler folgender Dienst bereit:

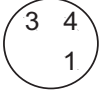
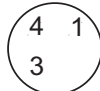
- I/O-Service
Der Dienst I/O-Service wird von der FDT/DTM-Rahmenapplikation WAGOframe verwendet, um den Feldbuskoppler zu parametrieren bzw. konfigurieren.

| Anschluss | Kontakt | Beschreibung |
|---|------------------|--------------|
|  | 1 | + 5 V |
| | 2 | - Data |
| | 3 | + Data |
| | 4 | 0 V DC |
| | Anschlussgewinde | Geschirmt |

Der USB-Speedway-Gerätetreiber ist Bestandteil der Installationsprogramme des Programmes „WAGOframe“. Siehe dazu Kapitel 7.1.

Digitale Eingänge:

Die Sensorleitungen dienen zur Versorgung angeschlossener Sensoren sowie der Übertragung der Sensorsignale. Die nachfolgende Tabelle gibt Ihnen Auskunft zu der Anschlussbelegung der Sensoranschlüsse:

| Anschluss | | Kontakt | Beschreibung |
|---|---|---------|---------------------------------|
| IN | IN | 1 | 24 V DC (aus U_{LS} gespeist) |
|  |  | 3 | 0 V DC |
| X1, X3, X5, X7 | X2, X4, X6, X8 | 4 | Input |

ACHTUNG**Höchste Strombelastbarkeit der Versorgungskontakte ist 4 A!**

Beachten Sie, dass die Sensoren aus der Versorgungslinie U_{LS} gespeist werden. Die Sensorstromaufnahme ist bei der Ermittlung des aktuellen Strombedarfs für die U_{LS} -Versorgungslinie zu berücksichtigen.

ACHTUNG**Sensorstromaufnahme darf 400 mA nicht überschreiten!**

Beachten Sie, dass die Stromaufnahme aller angeschlossenen Sensoren 400 mA nicht überschreiten darf. Die Aufteilung des Stromes auf die vorhandenen Anschlüsse ist dabei beliebig.

4 EtherNet/IP

4.1 EDS-Datei

Die EDS-Datei enthält die Kenndaten des Feldbuskopplers und Angaben zu seinen Kommunikationsfähigkeiten. Die EDS-Datei wird von der jeweiligen Projektierungssoftware eingelesen bzw. installiert.

Sie erhalten die EDS-Datei unter www.wago.com. Hinweise zur Installation der EDS-Datei entnehmen Sie bitte der Dokumentation der von Ihnen genutzten Projektierungssoftware.

4.2 Prozessabbild

Nach Inbetriebnahme des Feldbuskopplers ermittelt dieser automatisch alle angeschlossenen I/O-Module. Der Feldbuskoppler erstellt daraus ein lokales Prozessabbild, unterteilt in einen Ein- und Ausgangsbereich.

Haben Sie für den Feldbuskoppler den Feldbus EtherNet/IP ausgewählt, wird das Prozessabbild wortweise aufgebaut (word-alignment). Die interne Darstellung der Daten, die größer als ein Byte sind, wird im Intel-Format („Little Endian“) vorgenommen. Sind sowohl analoge als auch digitale I/O-Module im Knoten gesteckt, so werden die digitalen Ein- und Ausgangsdaten byteweise zusammengefasst und hinter den analogen Daten im Prozessabbild angehängt.

Die Größe des Prozessabbilds eines Ethernet-Knotens ermittelt sich aus den daran angeschlossenen 767-Komponenten. Das Prozessabbild hat eine Gesamtgröße von 4096 Byte: 2048 Byte Eingangsdaten und 2048 Bytes Ausgangsdaten.

Hinweis



Das Ein- und Ausgangsprozessabbild kann mehr Daten aufnehmen, als ggf. Ihre übergeordnete Steuerung übertragen kann. Sind die Daten größer als das Telegramm der übergeordneten Steuerung, dann wählen Sie mittels der Assembly-Instanzen aus, welche Daten übertragen werden sollen.

Den Zugriff auf das Prozessabbild erhalten Sie über die Assembly-Instanzen oder direkt über die EtherNet/IP-Objekte. Siehe dazu Kapitel 4.6 und 4.7.

4.3 Prozessdatenaustausch

Um einen Prozessdatenaustausch zwischen übergeordneter Steuerung und Feldbuskoppler zu ermöglichen, benötigen Sie die Assembly-Instanzen. Mit deren Hilfe schreiben und lesen Sie die Ein- und Ausgänge der 767-Komponenten.

4.3.1 Assembly-Instanzen

Mittels der Assembly-Instanzen initiieren Sie einen Prozessdatenaustausch. Dazu wählen Sie aus den vorhandenen statischen Assembly-Instanzen (siehe Kapitel 4.6.3) jeweils eine Instanz pro Senderichtung aus.

Für die Übertragung der Prozessdaten von der übergeordneten Steuerung (Originator) zum Feldbuskoppler (Target) stehen Ihnen folgende Assembly-Instanzen zur Verfügung:

- Instanz 101: für analoge und digitale Ausgangsdaten
- Instanz 102: für digitale Ausgangsdaten
- Instanz 103: für analoge Ausgangsdaten
- Instanz 113: für analoge und digitale Ausgangsdaten mit Diagnosebestätigung

Für die Übertragung der Prozessdaten vom Feldbuskoppler zur übergeordneten Steuerung stehen Ihnen folgende Assembly-Instanzen zur Verfügung:

- Instanz 104: für analoge und digitale Eingangsdaten inklusive Statusbyte
- Instanz 105: für digitale Eingangsdaten inklusive Statusbyte
- Instanz 106: für analoge Eingangsdaten inklusive Statusbyte
- Instanz 107: für analoge und digitale Eingangsdaten
- Instanz 108: für digitale Eingangsdaten
- Instanz 109: für analoge Eingangsdaten
- Instanz 112: Für analoge und digitale Eingangsdaten mit Statusbyte, aktueller Fehler und Diagnose

Hinweis



Wählen Sie eine Assembly-Instanz ohne Statusinformationen (107 – 109), so sind Informationen wie Feldbusfehler, anstehende Diagnose und Prozessdatenstatus (gültig/ungültig) nicht verfügbar und können nicht in die Applikation der übergeordneten Steuerung einfließen. Das Statusbyte ist in Klasse 64_{hex}, Attribut 5 beschrieben.

4.3.2 Ermittlung der Datenlänge

Nachdem Sie die benötigte Assembly-Instanz ausgewählt haben, benötigen Sie für die Projektierungssoftware der übergeordneten Steuerung noch die Länge der Daten pro Senderichtung der entsprechenden Instanz. Dazu haben Sie folgende Möglichkeiten:

Auslesen der Längen über die Assembly-Instanzen

Die einfachste Möglichkeit bietet sich durch Auslesen des Instanz-Attributs 4 der jeweiligen Assembly-Instanz 101 bis 113, welche verwendet werden soll. Das Attribut 4 der Assembly-Klassen (ID = 4) gibt die Gesamtzahl der in dieser Richtung übertragenen Datenbytes an. Diese Anzahl beinhaltet die Gesamtgröße der jeweiligen Ein- oder Ausgangsdaten inklusive Statusbyte, Fehler und Diagnose. Die einzelnen Datengrößen werden durch das Auslesen der Assembly-Instanzen nicht ermittelt. Dies ist nur mittels der Instanz-Attribute der Klasse 64_{hex} möglich.

Auslesen der Längen über die Klasse 64_{hex}, Instanz 1

Über die Instanz-Attribute 7 – 10 lesen Sie die Bitlängen der analogen und digitalen I/O-Module aus, die an dem Ethernet-Knoten angeschlossen sind.

Je nachdem, welche Assembly-Instanz Sie für den Austausch der Ein- und Ausgangsdaten wählen, sind noch das Statusbyte sowie drei Byte für den aktuellen Fehler und/oder Diagnose zu addieren. Die Länge der Diagnosedaten setzt sich aus einer festen Anzahl von 10 Bytes und einer Variablen zusammen. Die variable Anzahl ist von den am 767-Knoten angeschlossenem I/O-Modulen abhängig. Diese Anzahl deren Bytes lesen Sie über das Attribut 53 aus.

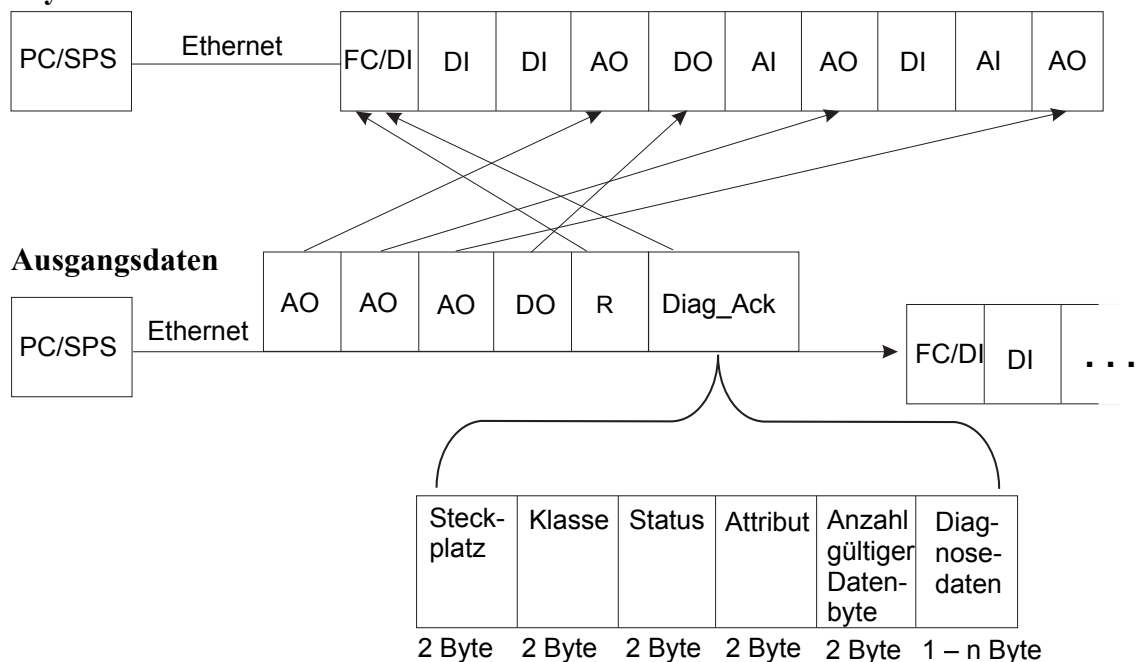
Detaillierte Informationen zu den Instanz-Attributen erhalten Sie in der entsprechenden Tabelle der Klasse 64_{hex}.

4.3.3 Beispiel mit den Assembly-Instanzen 101 und 104

In diesem Beispiel wird die Assembly-Instanz 101 für den Datenaustausch zwischen der übergeordneten Steuerung und dem Feldbuskoppler genutzt, die alle analogen und digitalen Ausgangsdaten beinhaltet. Für den Datenaustausch zwischen dem Feldbuskoppler und der übergeordneten Steuerung wird die Assembly-Instanz 104 genutzt, die außer den analogen und digitalen Eingangsdaten auch das Statusbyte beinhaltet.

Anhand einer Beispielkonfiguration ergibt sich folgender Aufbau der Prozessdaten, die von der übergeordneten Steuerung an den Feldbuskoppler gesandt werden:

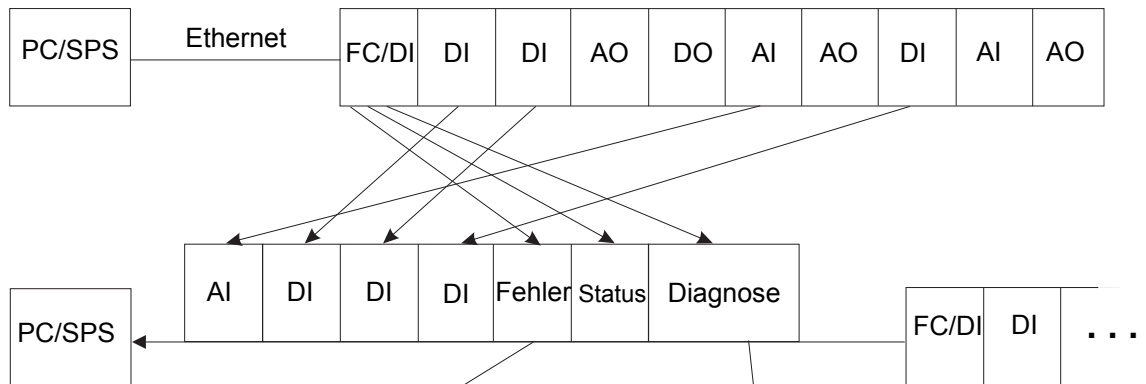
Physikalischer Aufbau



FC/DI: Digitale Eingänge auf dem Feldbuskoppler
 DI: Digitaler Eingang, 8 Bit
 DO: Digitaler Ausgang, 8 Bit
 AI: Analoger Eingang, 8 Byte
 AO: Analoger Ausgang, 8 Byte
 Diag_Ack: Diagnosebestätigung, 10 Byte + x Byte
 R: Reserviert, 4 Byte

Prozessdaten, die vom Feldbuskoppler an die übergeordnete Steuerung gesandt werden:

Physikalischer Aufbau



Eingangsdaten



- DI: Digitaler Eingang, 8 Bit
- DO: Digitaler Ausgang, 8 Bit
- AI: Analoger Eingang, 8 Byte
- AO: Analoger Ausgang, 8 Byte
- Fehler: 3 Byte
- Status: 1 Byte
- Diagnose: 10 Byte + x Byte
- Fehlergruppe: 1 Byte
- Fehlercode: 1 Byte
- Fehlerargument: 1 Byte

4.4 Diagnose

Im Netzwerk werden Ihnen bestimmte Diagnoseinformationen zur Verfügung gestellt. Diese können beispielsweise Fehlermeldungen und Warnungen des Feldbuskopplers oder der angeschlossenen I/O-Module beinhalten. Sie haben mehrere Möglichkeiten, vom Feldbuskoppler Diagnoseinformationen zu bekommen:

- Diagnose integriert in den Ein- und Ausgangsdaten,
- Diagnose über explizite Nachrichten,
- Diagnose gemischt über Ein- und Ausgangsdaten und explizite Nachrichten.

Die Diagnoseinformationen aller angeschlossenen I/O-Module und des Feldbuskopplers werden in Letzterem zur Abholung gespeichert. Mit Speicherung einer Diagnoseinformation wird auch das Bit 3 im Statusbyte gesetzt. Werden dem Feldbuskoppler mehr Diagnoseinformationen gemeldet als abgeholt, kann es zu einem Überlauf des internen Speichers kommen. Tritt dieser Zustand ein, so wird Bit 2 im Statusbyte gesetzt, wodurch ein Verlust von Diagnoseinformationen gemeldet wird. Ab diesem Zeitpunkt werden die jeweils aktuellen Diagnoseinformationen gespeichert und die ältesten Diagnoseinformationen verworfen. Näheres zum Statusbyte finden Sie in den nächsten Kapiteln und unter Attribut 5 der Konfigurationsklasse des Feldbuskopplers.

In den folgenden Kapiteln erfahren Sie, wie der Zugriff auf Diagnoseinformationen über das Protokoll Ethernet/IP stattfindet. Wie bei den Prozessdaten werden auch hier die Daten, die größer als ein Byte sind, im Intel-Format („Little-Endian“) dargestellt.

4.4.1 Diagnose integriert in den Ein- und Ausgangsdaten

Bei dieser Möglichkeit werden die Diagnoseinformationen hinter dem Prozessabbild angehängt. Dieses wird mittels der Assembly-Instanzen 104 bis 106, 112 oder 113 vorgenommen. Die Assembly-Instanzen 104 bis 106 fügen jeweils das Statusbyte an das Prozessabbild. Ist Bit 3 in diesem Statusbyte gesetzt, so liegt eine Diagnoseinformation an. Der Inhalt wird bei diesen Assembly-Instanzen nicht über die Ein- und Ausgangsdaten mitgeteilt, sondern kann von Ihnen z. B. über die expliziten Nachrichten ausgelesen werden. Siehe dazu Kapitel 4.4.2.

Die Assembly-Instanz 112 beinhaltet außer dem Statusbyte auch zusätzliche Diagnoseinformationen, welche die Art und Herkunft der Diagnose beschreiben. Über die Assembly-Instanz 112 teilt der Feldbuskoppler der übergeordneten Steuerung seinen Status und die Diagnoseinformationen mit. Die Assembly-Instanz 113 hat an der Stelle des Statusbytes ein reserviertes Byte, dessen Inhalt nicht relevant ist.

Die Diagnoseinformation ist wie folgt aufgeteilt:

| | | | | | | |
|----------------------|----------------------|------------------|-------------------|--------------------|--|----------------------------------|
| Statusbyte 1 Byte | Steckplatz 2 Byte | Klasse 2 Byte | Instanz 2 Byte | Attribut 2 Byte | Anzahl gültiger Datenbytes 2 Byte | Diagnosedaten 1 – n Bytes |
| | | | | | | |

Beispiel

Im gesamten 767-Knoten liefern die Analogausgangsmodule die maximale Anzahl von 4 Bytes an Diagnosedaten. Diese Anzahl können Sie auch mit einer expliziten Nachricht an Klasse 100, Instanz 1, Attribut 35 auslesen. Das digitale Ausgangsmodul auf Steckplatz 3 meldet einen aktuell aufgetretenen Fehler, der folgende Daten beinhaltet:

| Diagnose- meldung | Statusbyte | Steckplatz 2 Byte | Klasse 2 Byte | Instanz 2 Byte | Attribut 2 Byte | Anzahl gültiger Datenbytes 2 Byte | Diagnose- daten 1 – n Bytes |
|----------------------|------------|----------------------|------------------|-------------------|--------------------|--|---------------------------------------|
| Daten | 0x08 | 0x0003 | 0x0009 | 0x0005 | 0x0081 | 0x0001 | 0x01 0x00 0x00 0x00 |

Statusbyte:

Durch das gesetzte Bit 3 im Statusbyte (0x08) wird eine anstehende Diagnosemeldung angezeigt.

Steckplatz:

Der Steckplatz 0x0003 gibt an, dass das 3. I/O-Modul im 767-Knoten eine Diagnosemeldung angezeigt hat.

Klasse:

0x0009 kennzeichnet die Klasse „Digital Output Point“ als Fehlerklasse, welche den Digitalausgangsmodulen zugeordnet ist.

Instanz:

Die Instanz 0x0005 gibt den Kanal 5 des digitalen Ausgangsmoduls als Fehlerquelle an.

Attribut:

Das Attribut 0x0081 (129) gibt den Fehler „Open Load“ an. Die Beschreibung der Diagnosequelle anhand von Klasse, Instanz und Attribut (CIA) erhalten Sie in den Handbüchern der entsprechenden I/O-Module (Anhang).

Anzahl gültiger Datenbytes:

Die Länge der reinen Diagnosedaten wird durch das I/O-Modul festgelegt, welches die größte Anzahl an Diagnosebytes liefern kann. Wird in einer aktuellen Diagnosemeldung nur ein Teil der Diagnosedaten belegt, so werden diese durch die „Anzahl gültiger Datenbytes“ bestimmt. Beginnend mit dem nächsten Byte ist nur die hier angegebene Anzahl an Bytes gültig.

In dem Beispiel gibt der Wert 0x01 an, dass nur das nächste Byte in den Diagnosedaten einen gültigen Wert enthält. Die folgenden 3 Bytes sind reserviert und dessen Inhalt nicht relevant.

Diagnosedaten:

Das erste Byte der Diagnosedaten enthält eine 1 und kennzeichnet somit eine kommende Diagnose an Kanal 3 des digitalen Ausgangsmoduls, wie z. B. „Open Load“. Der Wert 0 würde eine gehende Diagnose kennzeichnen.

Einige Diagnosemeldungen erfordern eine Bestätigung. Mittels der Assembly-Instanz 113 werden die Diagnosebestätigungen von der übergeordneten Steuerung an den Feldbuskoppler übertragen. Die Dateninhalte entnehmen Sie der folgenden Tabelle:

| Daten vom Feldbuskoppler zur übergeordneten Steuerung | | | |
|--|------------------------|------------------------------------|-------------------------------|
| Prozessabbild | Fehler (3 Byte) | Statusbyte | Diagnosemeldung (10 + x Byte) |
| Daten von der übergeordneten Steuerung zum Feldbuskoppler | | | |
| Prozessabbild | Reserviert (4 Byte) | Diagnosebestätigung (10 + x Bytes) | |

Durch die Reservierung von 4 Byte ist der Anfang der Diagnosemeldung und der Diagnosebestätigung bezogen auf das Prozessabbild gleich (Offset von 4 Bytes).

Hinweis



Stehen mehrere Diagnosemeldungen gleichzeitig an, so werden diese direkt hintereinander übertragen. Die einzelnen Diagnosemeldungen stehen damit nur einem Zyklus, also in einem Ethernet-Telegramm, zur Verfügung. Im nächsten Ethernet-Telegramm würde auch schon die nächste Diagnosemeldung, sofern vorhanden, übertragen.

Information



Zur Bestätigung der Diagnose kopieren Sie die Daten aus der Diagnosemeldung in die Diagnosebestätigung.

4.4.2 Diagnose über explizite Nachrichten

Mittels expliziter Nachrichten können Sie die komplette Diagnose auch über Attribute auslesen. Die einzelnen Dateninhalte der Diagnosemeldung oder -bestätigung können Sie aus den Attributen der Klasse 64_{hex}, Instanz 1 auslesen.

| | | | | | | |
|------------|----------------------|------------------|-------------------|--------------------|--------------------------------------|------------------------------|
| Statusbyte | Steckplatz 2 Byte | Klasse 2 Byte | Instanz 2 Byte | Attribut 2 Byte | Anzahl gültiger Datenbytes 2 Byte | Diagnosedaten 1 – n Bytes |
|------------|----------------------|------------------|-------------------|--------------------|--------------------------------------|------------------------------|

Attribut 5

Attribut 54/55

Attribut 5: Dieses Attribut enthält das Statusbyte.

Attribut 54: Dieses enthält die Diagnosemeldung, deren Inhalt sich aus Steckplatz, Klasse, Instanz, Attribut, Anzahl gültiger Datenbytes und Diagnosedaten zusammensetzt.

Attribut 55: Dieses enthält die Diagnosebestätigung, deren Inhalt sich aus Steckplatz, Klasse, Instanz, Attribut, Anzahl gültiger Datenbytes und Diagnosedaten zusammensetzt.

Die Beschreibungen der einzelnen Inhalte, wie z. B. Steckplatz usw., erhalten Sie im Kapitel 4.4.1.

Über das Attribut 5 ist das Statusbyte auslesbar, indem Bit 3 angibt, ob eine Diagnose ansteht oder nicht. Dieses Bit bleibt so lange gesetzt, wie auch Diagnosemeldungen anstehen. Wird die Diagnose nur über explizite Nachrichten verarbeitet, so sollte das Attribut 5 in bestimmten Zeitabständen ausgelesen werden, um eine Diagnosemeldung zu erkennen. Steht eine Diagnosemeldung an, so kann über Attribut 54 dieser Klasse die Diagnosequelle ausgelesen werden. Ist eine Bestätigung der Diagnose erforderlich, kann der Anwender das Attribut 55 hierfür verwenden.

Der Aufbau der Daten von Attribut 54 und 55 entspricht dem Aufbau, wie er in den Ein- und Ausgangsdaten gesendet wird (siehe oben stehende Abb.). Bitte beachten Sie, dass die Daten im Format „Little-Endian“ gesendet werden und Sie diese auch in diesem Format bestätigen. Steht keine weitere Diagnose mehr zur Verfügung, so wird dieses durch Rücksetzen des Bit 3 von Attribut 5 gekennzeichnet. Weiterhin wird der gesamte Dateninhalt von Attribut 54 auf Null gesetzt.

Hinweis



Eine Diagnosemeldung kann nur einmal über Attribut 54 ausgelesen werden! Bei einem weiteren Lesevorgang wird entweder die nächste anstehende Diagnosemeldung geliefert oder aber keine (Dateninhalt von Attribut 54 ist null).

Information



Zur Bestätigung der Diagnose kopieren Sie die Daten aus der Diagnosemeldung in die Diagnosebestätigung.

4.4.3 Diagnose gemischt über Ein- und Ausgangsdaten und explizite Nachrichten

Es gibt mehrere Möglichkeiten, die Diagnose gemischt über Ein- und Ausgangsdaten und explizite Nachrichten zu empfangen und zu bestätigen. Mit den Assembly-Instanzen 104, 105 oder 106 stehen Ihnen diese Alternativen zur Verfügung.

- **Instanz 104**
Für analoge und digitale Eingangsdaten einschließlich Statusbyte.
- **Instanz 105**
Für digitale Eingangsdaten einschließlich Statusbyte.
- **Instanz 106**
Für analoge Eingangsdaten einschließlich Statusbyte.

In diesen Assembly-Instanzen ist das Statusbyte integriert, welches Auskunft über eine anliegende Diagnose gibt (Bit 3 gesetzt). Die eigentliche Diagnose lesen Sie dann mithilfe der expliziten Nachrichten aus der Klasse 100, Instanz 1, Attribut 54 aus. Mit dem Attribut 55 derselben Klasse lässt sich dann auch die Diagnose bestätigen. Siehe dazu Kapitel 4.4.2.

Es stehen Ihnen weitere Kombinationsmöglichkeiten aus Ein- und Ausgangsdaten und den expliziten Nachrichten zur Verfügung, um die Diagnoseinformationen ins Gesamtkonzept einzubinden.

4.5 Objektmodell

Für die Netzwerkkommunikation verwendet EtherNet/IP ein Objektmodell, in dem alle Funktionen und Daten des Feldbuskopplers beschrieben sind. Jeder Knoten im Netz wird als Sammlung von Objekten dargestellt.

Das Objektmodell enthält Begriffe, die folgendermaßen definiert sind:

- **Objekt (object):**
Ein Objekt ist eine abstrakte Darstellung von einzelnen, zusammengehörigen Bestandteilen innerhalb eines Gerätes. Es ist bestimmt durch seine Daten oder Eigenschaften (Attributes), seine nach außen bereitgestellten Funktionen oder Dienste (Services) und durch sein definiertes Verhalten (Behaviour).
- **Klasse (class):**
Eine Reihe von Objekten, die alle die gleiche Art von Systemkomponenten darstellen. Eine Klasse ist die Verallgemeinerung eines Objektes. Alle Objekte in einer Klasse sind in Bezug auf ihre Form und ihr Verhalten identisch, wobei sie jedoch unterschiedliche Attributwerte umfassen können.
- **Instanz (instance):**
Eine spezifische und tatsächliche (physikalische) Ausprägung eines Objektes. Beispiel: Neuseeland ist eine Instanz der Objektklasse „Land“. Die Benennungen „Objekt“, „Instanz“ und „Objektinstanz“ beziehen sich alle auf eine spezifische Instanz.
- **Variable (attribute):**
Die Beschreibung eines extern sichtbaren Merkmals oder der Funktion eines Objektes. Normalerweise liefern Attribute die Status-Information oder regeln die Funktion eines Objektes. Beispiel: der ASCII-Name eines Objektes und die Wiederholungsfrequenz eines periodischen Objektes.
- **Dienst (service):**
Eine Funktion, die von einem Objekt und/oder einer Objekt-Klasse unterstützt wird. CIP definiert eine Gruppe gemeinsamer Dienste, die auf die Variablen (Attribute) angewendet werden.
- **Verhalten (behaviour):**
Festlegung, wie ein Objekt funktioniert. Die Funktionen resultieren aus unterschiedlichen Ereignissen, die das Objekt ermittelt, wie zum Beispiel der Empfang von Serviceanforderungen, die Erfassung interner Störungen oder der Ablauf von Zeitnehmern.

4.6 CIP-Klassen

Die CIP-Klassen sind in der CIP-Spezifikation der ODVA enthalten. Sie beschreiben, unabhängig von der physikalischen Schnittstelle, z. B. Ethernet, CAN, deren Eigenschaften (Band 1). Die physikalische Schnittstelle wird in einer weiteren Spezifikation beschrieben. Für EtherNet/IP ist das der Band 2, welcher die Adaption des EtherNet/IP an CIP beschreibt.

WAGO nutzt hierbei die Klassen 1, 2, 4 – 6 und auch F4_{hex}, welche in Band 1 („Common Industrial Protocol“) beschrieben sind. Aus dem Band 2 (EtherNet/IP Adaption of CIP) werden die Klassen F5_{hex} und F6_{hex} unterstützt. Die Klassen aus der unten stehenden Tabelle werden in den folgenden Kapiteln näher beschrieben. Die WAGO-spezifischen Klassen werden ab Kapitel 4.7 beschrieben.

Tab. 1: CIP-Klassen

| Klasse | Name | Information |
|-------------------|---------------------------|---------------------|
| 01 _{hex} | Identity Object | Siehe Kapitel 4.6.1 |
| 02 _{hex} | Message Router Object | Siehe Kapitel 4.6.2 |
| 04 _{hex} | Assembly Object | Siehe Kapitel 4.6.3 |
| 05 _{hex} | Connection Object | Siehe Kapitel 4.6.4 |
| 06 _{hex} | Connection Manager Object | Siehe Kapitel 4.6.5 |
| F4 _{hex} | Port Class Object | Siehe Kapitel 4.6.6 |
| F5 _{hex} | TCP/IP Interface Object | Siehe Kapitel 4.6.7 |
| F6 _{hex} | Ethernet Link Object | Siehe Kapitel 4.6.8 |

Tab. 2: Erläuterungen zu den Tabellenköpfen

| Spaltenüberschrift | Beschreibung |
|--------------------|---|
| Attribut ID | Integerwert, der dem entsprechenden Attribut zugeordnet ist |
| Zugriff | <p>Set: Auf das Attribut kann mittels des Dienstes Set_Attribute zugegriffen werden (Schreiben/Verändern des Attribut-Wertes).</p> <p>Hinweis: Unterstützt ein Attribut den Dienst Set_Attribute, so kann dieses auch mit dem Dienst Get_Attribute angesprochen werden.</p> <p>Get: Auf das Attribut kann mittels Get_Attribute Services zugegriffen werden (Lesen des Attribut-Wertes).</p> <p>Get_Attribute_All: Liefert den Inhalt aller Attribute.</p> <p>Set_Attribute_Single: Modifiziert einen Attribut-Wert.</p> <p>Reset: Führt einen Neustart durch. 0: Emuliert einen Neustart. 1: Emuliert einen Neustart und stellt die Werkseinstellungen wieder her.</p> |

Tab. 2: Erläuterungen zu den Tabellenköpfen

| Spaltenüberschrift | Beschreibung |
|---------------------------|---|
| NV | NV (non volatile): Das Attribut wird permanent im Feldbuskoppler gespeichert. V (volatile): Das Attribut wird nicht permanent im Feldbuskoppler gespeichert. |
| Name | Bezeichnung des Attributs. |
| Datentyp | Bezeichnung des CIP-Datentyps des Attributes. |
| Beschreibung | Kurze Beschreibung zu dem Attribute. |
| Defaultwert | Werkseinstellung |

4.6.1 Identity Object (01_{hex})

Die Klasse Identity dient dazu, allgemeine Informationen des Feldbuskopplers bereitzustellen, die diesen eindeutig identifizieren.

Klassen-Attribute

Tab. 3: Klasse „Identity Object“: Klassen-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-------------------------------------|----------|--|-------------|
| 1 | Get | Revision | UINT | Version des Objektes | 1 (0x0001) |
| 2 | Get | Max Instance | UINT | Maximale Instanz | 1 (0x0001) |
| 3 | Get | Max ID Number of Class Attributes | UINT | Maximale Anzahl der Klassen-Attribute (nur mit dem Dienst Get_Attribute_All) | 0 (0x0000) |
| 4 | Get | Max ID Number of Instance Attribute | UINT | Maximale Anzahl der Instanz-Attribute (nur mit dem Dienst Get_Attribute_All) | 0 (0x0000) |

Instanz-Attribute 1

Tab. 4: Klasse „Identity Object“: Instanz-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|----------------|------------|--|------------------|
| 1 | Get | Vendor ID | UINT | Herstelleridentifikation | 40 (0x0028) |
| 2 | Get | Device Type | UINT | Generelle Typbezeichnung des Feldbuskopplers | 12 (0x000C) |
| 3 | Get | Product Code | UINT | Bezeichnung des Feldbuskopplers | 1301 (0x0515) |
| 4 | Get | Revision | STRUCT of: | Version des Identity-Objektes | Firmwareabhängig |
| | | Major Revision | USINT | | |
| | | Minor Revision | USINT | | |

Tab. 4: Klasse „Identity Object“: Instanz-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|---------------|--------------|--------------|------------------------------------|
| 5 | Get | Status | WORD | Siehe * | Vom aktuellen Zustand abhängig |
| 6 | Get | Serial Number | UDINT | Seriennummer | - |
| 7 | Get | Product Name | SHORT_String | Produktname | WAGO FC ETHERNET 8DI 24 V DC |

Übersicht der Dienste

Tab. 5: Klasse „Identity Object“: Dienste

| Servicecode | Service vorhanden | | Servicename | Beschreibung |
|-------------|-------------------|---------|----------------------|---|
| | Klasse | Instanz | | |
| 01 hex | Ja | Ja | Get_Attribute_All | Liefert den Inhalt aller Attribute |
| 05 hex | Nein | Ja | Reset | Führt den Reset-Service aus. Serviceparameter: 0: Emuliert einen Neustart 1: Emuliert einen Neustart und stellt die Werkseinstellungen wieder her |
| 0E hex | Nein | Ja | Get_Attribute_Single | Liefert den Inhalt des entsprechenden Attributes |

* **Bit 0:** Zuweisung zu einem Master; **Bit 1=0** (reserviert); **Bit 2:** Konfiguriert: (= 0: Konfiguration ist unverändert; =1: Konfiguration weicht von Herstellerparametern ab); **Bit 3 = 0** (reserviert); **Bit 4-7:** Extended Device Status: (= 0010: Mindestens eine fehlerhafte I/O-Verbindung, = 0011: Keine I/O-Verbindung hergestellt);

Bit 8-11: nicht genutzt

Bit 12-15 = 0 (reserviert)

4.6.2 Message Router Object (02_{hex})

Das „Message Router Object“ stellt Verbindungspunkte in Form von Klassen oder Instanzen bereit, welche einen Client zum Adressieren von Diensten (Lesen, Schreiben) nutzen kann. Diese Nachrichten können sowohl verbindungsorientiert (connected) als auch verbindungslos (unconnected) vom Client an den Feldbuskoppler gesendet werden.

Klassen-Attribute

Tab. 6: Klasse „Message Router Object“: Klassen-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|--------------------------------------|----------|---------------------------------------|-------------|
| 1 | Get | Revision | UINT | Version des Objektes | 1 (0x0001) |
| 2 | Get | Number of Attributes | UINT | Anzahl der Attribute | 0 (0x0000) |
| 3 | Get | Number of Services | UINT | Anzahl der Dienste | 0 (0x0000) |
| 4 | Get | Max ID Number of Class Attributes | UINT | Maximale Anzahl der Klassen-Attribute | 0 (0x0000) |
| 5 | Get | Max ID Number of Instance Attributes | UINT | Maximale Anzahl der Instanz-Attribute | 0 (0x0000) |

Hinweis



Die Klassen-Attribute sind nur mit dem Dienst Get_Attribute_All erreichbar.

Instanz-Attribute 1

Tab. 7: Klasse „Message Router Object“: Instanz-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------------|------------|--|--|
| 1 | Get | ObjectList | STRUCT of: | - | |
| | | Number | UINT | Anzahl implementierter Klassen | 25 (0x0019) |
| | | Classes | UINT | Implementierte Klassen | 0x0001, 0x0002, 0x0004, 0x0006, 0x00F4, 0x00F5, 0x00F6, 0x0064 –0x0074, 0x0082, 0x00A0, 0x00A1, 0x00A2, 0x00A6, 0x00A7, 0x00AA, 0x00AB, 0x00A3, 0x00A4, 0x00A5, 0x00A8, 0x00A9, 0x00AC, 0x00AD |
| 2 | Get | NumberAvailable | UINT | Maximale Anzahl von unterschiedlichen Verbindungen | 128 (0x0080) |

Übersicht der Dienste

Tab. 8: Klasse „Message Router Object“: Dienste

| Servicecode | Service vorhanden | | Servicename | Beschreibung |
|-------------|-------------------|---------|----------------------|--|
| | Klasse | Instanz | | |
| 01 hex | Ja | Nein | Get_Attribute_All | Liefert den Inhalt aller Attribute |
| 0E hex | Nein | Ja | Get_Attribute_Single | Liefert den Inhalt des entsprechenden Attributes |

4.6.3 Assembly Object (04 hex)

Mit Hilfe der Assembly-Klasse lassen sich mehrere auch verschiedenartige Objekte zusammenfassen. Diese können z. B. Ein- und Ausgangsdaten, Status- und Steuerinformationen oder Diagnoseinformationen sein. WAGO nutzt hier die herstellereigenen Instanzen, um diese Objekte in verschiedenen Anordnungen für Sie bereitzustellen. Hierdurch steht Ihnen ein effizienter Weg zum Austausch von Prozessdaten zur Verfügung. Im Folgenden werden die einzelnen Instanzen mit deren Inhalten und Anordnungen beschrieben.

Klassen-Attribute

Tab. 9 Klasse „Assembly Object“: Klassen-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|--------------|----------|----------------------|--------------|
| 1 | Get | Revision | UINT | Version des Objektes | 2 (0x0002) |
| 2 | Get | Max Instance | UINT | Höchste Instanz | 113 (0x0071) |

Instanz-Attribute 101 (65 hex)

Diese Assembly-Instanz enthält analoge und digitale Ausgangsdaten.

Tab. 10: Klasse „Assembly Object“: Instanz-Attribut 101 (65 hex)

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|---|-------------|
| 3 | Get/Set | Data | ARRAY of BYTE | Es sind nur analoge und digitale Ausgangsdaten im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 102 (66 hex)

Diese Assembly-Instanz enthält nur digitale Ausgangsdaten.

Tab. 11: Klasse „Assembly Object“: Instanz-Attribut 102 (66 hex)

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|---|-------------|
| 3 | Get/Set | Data | ARRAY of BYTE | Es sind nur digitale Ausgangsdaten im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 103 (67 hex)

Diese Assembly-Instanz enthält nur analoge Ausgangsdaten.

Tab. 12: Klasse „Assembly Object“: Instanz-Attribut 103 (67 hex)

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|--|-------------|
| 3 | Get/Set | Data | ARRAY of BYTE | Es sind nur analoge Ausgangsdaten im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 104 (68 hex)

Diese Assembly-Instanz enthält nur analoge und digitale Eingangsdaten und den Status.

Tab. 13: Klasse „Assembly Object“: Instanz-Attribut 104 (68 hex)

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|--|-------------|
| 3 | Get | Data | ARRAY of BYTE | Es sind nur analoge und digitale Eingangsdaten und Status im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 105 (69 hex)

Diese Assembly-Instanz enthält nur digitale Eingangsdaten und den Status.

Tab. 14: Klasse „Assembly Object“: Instanz-Attribut 105 (69 hex)

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|--|-------------|
| 3 | Get | Data | ARRAY of BYTE | Es sind nur digitale Eingangsdaten und Status im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 106 (6A_{hex})

Diese Assembly-Instanz enthält nur analoge Eingangsdaten und den Status.

Tab. 15: Klasse „Assembly Object“: Instanz-Attribute 106 (6A_{hex})

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|---|-------------|
| 3 | Get | Data | ARRAY of BYTE | Es sind nur analoge Eingangsdaten sowie der Status im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 107 (6B_{hex})

Diese Assembly-Instanz enthält analoge und digitale Eingangsdaten.

Tab. 16: Klasse „Assembly Object“: Instanz-Attribute 107 (6B_{hex})

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|---|-------------|
| 3 | Get | Data | ARRAY of BYTE | Es sind nur analoge und digitale Eingangsdaten im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 108 (6C_{hex})

Diese Assembly-Instanz enthält digitale Eingangsdaten.

Tab. 17: Klasse „Assembly Object“: Instanz-Attribute 108 (6C_{hex})

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|---|-------------|
| 3 | Get | Data | ARRAY of BYTE | Es sind nur digitale Eingangsdaten im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 109 (6D_{hex})

Diese Assembly-Instanz enthält analoge Eingangsdaten.

Tab. 18: Klasse „Assembly Object“: Instanz-Attribute 109 (6D_{hex})

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|--|-------------|
| 3 | Get | Data | ARRAY of BYTE | Es sind nur analoge Eingangsdaten im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 112 (70_{hex})

Diese Assembly-Instanz enthält analoge und digitale Eingangsdaten, aktuelle Fehler, Status und Diagnose.

Tab. 19: Klasse „Assembly Object“: Instanz-Attribute 112 (70_{hex})

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|---|-------------|
| 3 | Get | Data | ARRAY of BYTE | Es sind nur analoge und digitale Eingangsdaten, aktuelle Fehler, Status und Diagnose im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | - |

Instanz-Attribute 113 (71_{hex})

Diese Assembly-Instanz enthält analoge und digitale Ausgangsdaten, 4 reservierte Bytes*, Status und Diagnosebestätigung.

Tab. 20: Klasse „Assembly Object“: Instanz-Attribute 113 (71_{hex})

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------|---------------|---|-------------|
| 3 | Get/Set | Data | ARRAY of BYTE | Es sind nur analoge und digitale Ausgangsdaten, 4 reservierte Bytes*, Status und Diagnosebestätigung im Prozessabbild enthalten | - |
| 4 | Get | Data Size | ARRAY of BYTE | Anzahl der Bytes im Prozessabbild | |

Übersicht der Dienste

Tab. 21: Klasse „Assembly Object“: Dienste

| Servicecode | Service vorhanden | | Servicename | Beschreibung |
|-------------|-------------------|---------|----------------------|--|
| | Klasse | Instanz | | |
| 0E hex | Ja | Ja | Get_Attribute_Single | Liefert den Inhalt des entsprechenden Attributes |
| 10 hex | Nein | Ja | Set_Attribute_Single | Modifiziert einen Attribut-Wert |

Das Schreiben des Attributes 3 der Assembly-Instanzen 101 ... 103 und 113 wird von der Software überprüft. Die Überschreitung von Grenzwerten wird festgestellt und, sofern erforderlich, korrigiert. Es wird jedoch keine Schreibanfrage abgelehnt. Das bedeutet, wenn weniger Daten empfangen werden als erwartet, dann werden nur diese Daten geschrieben. Wenn mehr Daten empfangen werden als erwartet, dann werden die empfangenen Daten an der oberen Grenze entfernt. Jedoch wird im Falle von expliziten Nachrichten ein definiertes CIP generiert, obwohl die Daten geschrieben worden sind.

* Diese reservierten Bytes dienen nur dazu, die gleichen Datenbreite der Status- und Diagnoseinformationen, wie der Assembly-Instanz 112, bereitzustellen: 1 Byte Status + 3 Byte Fehler.

Instanz-Attribut 198 (C6_{hex}) „Input Only“

Diese Instanz dient zum Verbindungsaufbau, wenn keine Ausgänge angesprochen werden sollen bzw. wenn Eingänge abgefragt werden, die schon in einer Exclusive-Owner-Verbindung benutzt werden. Die Datenlänge dieser Instanz beträgt immer Null.

Diese Instanz kann nur im „Consumed Path“ benutzt werden (aus Sicht des Slave).

Instanz-Attribut 199 (C7_{hex}) „Listen Only“

Mit dieser Instanz kann eine Verbindung aufgebaut werden, die auf einer vorhandenen Exklusiv-Owner-Verbindung aufsetzt. Dabei hat die neue Verbindung die gleichen Übertragungsparameter wie die Exclusive-Owner-Verbindung. Wird die Exklusiv-Owner-Verbindung abgebaut, wird auch automatisch diese Verbindung abgebaut. Die Datenlänge dieser Instanz beträgt immer Null.

Diese Instanz kann nur im "Consumed Path" (aus Sicht des Slaves) benutzt werden.

4.6.4 Connection Object (05_{hex})

Die Klassen- und Instanz-Attribute dieser Klasse sind nicht sichtbar, da die Verbindungen über den Connection Manager auf- und abgebaut werden.

4.6.5 Connection Manager Object (06_{hex})

Das „Connection Manager Object“ stellt die internen Ressourcen bereit, die für die Ein- und Ausgangsdaten und explizite Nachrichten benötigt werden. Weiterhin ist die Verwaltung dieser Ressource eine Aufgabe des „Connection Manager Object“.

Für jede Verbindung (Ein- und Ausgangsdaten oder explizite) wird eine weitere Instanz der Connection-Klasse erzeugt. Die Verbindungsparameter werden dem Dienst „Forward Open“ entnommen, der für den Aufbau einer Verbindung zuständig ist.

Folgende Dienste werden für die erste Instanz unterstützt:

- Forward_Open
- Unconnected_Send
- Forward_Close

Es sind keine Klassen- und Instanz-Attribute sichtbar.

4.6.6 Port Class Object (F4_{hex})

Das „Port Class Object“ spezifiziert die an dem Feldbuskoppler vorhandenen CIP-Ports. Für jeden CIP-Port gibt es eine Instanz.

Klassen-Attribut

Tab. 22: Klasse „Port Class Object“: Klassen-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|---------------|------------------------------|---|--|
| 1 | Get | Revision | UINT | Version des Objektes | 1 (0x0001) |
| 2 | Get | Max Instance | UINT | Max. Anzahl von Instanzen | 1 (0x0001) |
| 3 | Get | Num Instances | UINT | Anzahl von aktuellen Ports | 1 (0x0001) |
| 8 | Get | Entry Port | UINT | Instanz des Portobjektes, von wo die Anfrage eingetroffen ist | 1 (0x0001) |
| 9 | Get | All Ports | Array of Struct UINT UINT | Array von Instanzattributen 1 und 2 aller Instanzen | 0 (0x0000) 0 (0x0000) 4 (0x0004) 2 (0x0002) |

Instanz-Attribute 1

Tab. 23: Klasse „Port Class Object“: Instanz-Attribute

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|--------------|--------------|---|--|
| 1 | Get | Port Type | UINT | - | 4 (0x0004) |
| 2 | Get | Port Number | UINT | CIP-Portnummer | 2 (0x0002) (EtherNet/IP) |
| 3 | Get | Port Object | UINT | Anzahl von 16 Bit Wörtern im folgenden Pfad | 2 (0x0002) |
| | | | Padded EPATH | Objekt, das diesen Port verwaltet | 0x20 0xF5 0x24 0x01 (entspricht dem TCP/IP-Interface Object) |
| 4 | Get | Port Name | Short String | Portname | “ ” |
| 7 | Get | Node Address | Padded EPATH | Portsegment (IP-Adresse) | Abhängig von der IP-Adresse |

Übersicht der Dienste

Tab. 24: Klasse „Port Class Object“: Dienste

| Servicecode | Service vorhanden | | Servicename | Beschreibung |
|-------------|-------------------|---------|----------------------|--|
| | Klasse | Instanz | | |
| 01 hex | Ja | Ja | Get_Attribute_All | Liefert den Inhalt aller Attribute |
| 0E hex | Ja | Ja | Get_Attribute_Single | Liefert den Inhalt des entsprechenden Attributes |

4.6.7 TCP/IP Interface Object (F5_{hex})

Das „TCP/IP Interface Object“ stellt die Einrichtung zur Konfiguration der TCP/IP-Netzwerk-Schnittstelle eines Feldbuskopplers bereit. Beispiele konfigurierbarer Objekte umfassen die IP-Adresse, Netzwerkmaske und Gateway-Adresse des Feldbuskopplers.

Bei der zugrunde liegenden physikalischen Kommunikationsschnittstelle, die mit dem TCP/IP-Schnittstellen-Objekt verbunden ist, kann es sich um eine beliebige Schnittstelle handeln, die das TCP/IP-Protokoll unterstützt. An einem TCP/IP-Schnittstellen-Objekt kann zum Beispiel eine der folgenden Komponenten angeschlossen werden: eine Ethernet-Schnittstelle 802.3, eine ATM-Schnittstelle oder eine serielle Schnittstelle für Protokolle wie PPP. Das TCP/IP-Schnittstellen-Objekt stellt ein Attribut bereit, welches das linkspezifische Objekt für die angeschlossene physikalische Kommunikationsschnittstelle identifiziert. Das linkspezifische Objekt soll üblicherweise linkspezifische Zähler sowie beliebige linkspezifische Konfigurationsattribute bereitstellen.

Jedes Gerät muss genau eine Instanz des TCP/IP-Schnittstellenobjektes für jede TCP/IP-fähige Kommunikationsschnittstelle unterstützen. Eine Anfrage für den Zugriff auf die 1. Instanz des TCP/IP-Schnittstellen-Objektes muss sich immer auf die Instanz beziehen, die mit der Schnittstelle verbunden ist, über welche die Anfrage eingegangen ist.

Klassen-Attribut

Tab. 25: Klasse „TCP/IP Interface Object“: Klassen-Attribute

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|--------------|----------|--|-------------|
| 1 | Get | Revision | UINT | Version des Objektes | 1 (0x0001) |
| 2 | Get | Max Instance | UINT | Maximale Anzahl von Instanzen | 1 (0x0001) |
| 3 | Get | Num Instance | UINT | Anzahl der aktuell instanziierten Verbindungen | 1 (0x0001) |

Instanz-Attribute 1

Tab. 26: Klasse „TCP/IP Interface Object“: Instanz-Attribute

| Attribut-ID | Zugriff | NV | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|----|--------------------------|----------|--|-----------------|
| 1 | Get | V | Status | DWORD | Interface-Status | - |
| 2 | Get | V | Configuration Capability | DWORD | Interfaceflags für mögliche Konfigurationsarten | 23 (0x00000017) |
| 3 | Get/Set | NV | Configuration Control | DWORD | Legt fest wie der Feldbuskoppler nach dem ersten Neustart zu seiner TCP/IP-Konfiguration kommt | 17 (0x00000011) |

Tab. 26: Klasse „TCP/IP Interface Object“: Instanz-Attribute

| Attribut-ID | Zugriff | NV | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|----|-------------------------|--------------|---|--|
| 4 | Get | V | Physical Link Object | STRUCT of | - | |
| | | | Path | UINT | Anzahl von 16-Bit-Wörtern im folgenden Pfad | 2 (0x0002) |
| | | | | Padded EPATH | Logischer Pfad, der auf das physikalische Link-Objekt zeigt | 0x20 0xF6 0x24 0x03 (entspricht dem Ethernet Link Object) |
| 5 | Set | NV | Interface Configuration | STRUCT of | - | |
| | | | IP Address | UDINT | IP-Adresse | 0 |
| | | | Network Mask | UDINT | Netzwerkmaske | 255 (0xFF) 255 (0xFF) 0 (0x00) 0 (0x00) |
| | | | Gateway Address | UDINT | IP-Adresse des Standard-Gateway | 0 |
| | | | Name Server | UDINT | IP-Adresse des primären Name-Servers | 0 |
| | | | Name Server 2 | UDINT | IP-Adresse des sekundären Name-Servers | 0 |
| | | | Domain Name | STRING | Standard -Domain-Name | „“ |
| 6 | Get/Set | NV | Host Name | STRING | Gerätename | MAC-ID |

Übersicht der Dienste

Tab. 27: Klasse „TCP/IP Interface Object“: Dienste

| Servicecode | Service vorhanden | | Servicename | Beschreibung |
|-------------|-------------------|---------|----------------------|--|
| | Klasse | Instanz | | |
| 01 hex | Ja | Ja | Get_Attribute_All | Liefert den Inhalt aller Attribute |
| 0E hex | Ja | Ja | Get_Attribute_Single | Liefert den Inhalt des entsprechenden Attributes |
| 10 hex | Nein | Ja | Set_Attribute_Single | Modifiziert einen Attribut-Wert |

4.6.8 Ethernet Link Object (F6_{hex})

Das „Ethernet Link Object“ enthält linkspezifische Zähler- und Statusinformationen für eine Kommunikationsschnittstelle vom Typ Ethernet 802.3. Jedes Gerät muss genau eine Instanz des Ethernet-Link-Objektes für jede Kommunikationsschnittstelle vom Typ Ethernet IEEE 802.3 unterstützen. Für die Geräte kann auch eine Ethernet-Link-Objektinstanz für eine interne Schnittstelle verwendet werden, wie zum Beispiel ein interner Port mit integriertem Switch.

Klassen-Attribut

Tab. 28: Klasse „Ethernet Link Object“: Klassen-Attribut

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|---------------|----------|--|-----------------|
| 1 | Get | Revision | UINT | Version des Objektes | 3 (0x0003) |
| 2 | Get | Max Instance | UDINT | Max. Anzahl von Instanzen | 3 (00000003) |
| 3 | Get | Num Instances | UDINT | Anzahl der aktuell instanziierten Verbindungen | 3 (00000003) |

Instanz-Attribut 1

Tab. 29: Klasse „Ethernet Link Object“: Instanz-Attribut 1

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|------------------------|------------------|--|--|
| 1 | Get | Interface Speed | UDINT | Übertragungsgeschwindigkeit | 10 (0x0A) oder 100 (0x64) |
| 2 | Get | Interface Flags | DWORD | Interface Konfigurations-/ Statusinformationen Bit 0: Link-Status Bit 1: Halb-/Vollduplex Bit 2 – 4: Erkennungsstatus Bit 5: Manuelle Einstellungen erfordern Reset Bit 6: Lokaler Hardwarefehler Bit 7 – 31: Reserviert | Wert ist von der Ethernet-Verbindung abhängig. |
| 3 | Get | Physical Address | ARRAY of 6 UINTs | MAC-Adresse | MAC-ID des Feldbuskopplers |
| 6 | Get/Set | Interface Control | STRUCT of: | Konfiguration der physikalischen Schnittstelle | |
| | | Control Bits | WORD | Interface-Kontrollbits Bit 0: Automatische Erkennung Bit 1: Vorgabe Duplex-Modus Bit 2 – 15: Reserviert | 0x0001 |
| | | Forced Interface Speed | UINT | Für die Schnittstelle vorgegebene Geschwindigkeit | 0 (0x0000) |

Tab. 29: Klasse „Ethernet Link Object“: Instanz-Attribut 1

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------------|--------------|--|-------------|
| 7 | Get | Interface Type | USINT | Schnittstellentyp: Wert 0: Unbekannt Wert 1: Interne Schnittstelle, zum Beispiel bei einem integrierten Switch. Wert 2: Twistet-Pair (z. B. 100Base-TX). Wert 3: Glasfaser (z. B. 100Base-FX). Wert 4 – 256: Reserviert | 2 (0x02) |
| 8 | Get | Interface State | USINT | Schnittstellenstatus: Wert 0: Unbekannt Wert 1: Schnittstelle aktiv und zum Senden/Empfangen bereit. Wert 2: Schnittstelle deaktiviert. Wert 3: Schnittstelle testet. Wert 4 – 256: Reserviert | - |
| 9 | Get/Set | Admin State | USINT | Verwaltungsstatus: Wert 0: Reserviert Wert 1: Schnittstelle aktivieren. Wert 2: Schnittstelle deaktivieren. Ist diese die einzige CIP-Schnittstelle, so wird eine Anforderung zum Deaktivieren mit einem Fehler quittiert (Fehlercode 0x09) Wert 3 – 256: Reserviert | 1 (0x01) |
| 10 | Get | Interface Lable | SHORT_STRING | Lesbare Identifikation | „Port 1“ |

Instanz-Attribut 2

Tab. 30: Klasse „Ethernet Link Object“: Instanz-Attribut 2

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|------------------------|---------------------|--|--|
| 1 | Get | Interface Speed | UDINT | Übertragungsgeschwindigkeit | 10 (0x0A) oder 100 (0x64) |
| 2 | Get | Interface Flags | DWORD | Interface Konfigurations-/ Statusinformationen Bit 0: Link-Status Bit 1: Halb-/Vollduplex Bit 2 – 4: Erkennungsstatus Bit 5: Manuelle Einstellungen erfordern Reset Bit 6: Lokaler Hardwarefehler Bit 7 – 31: Reserviert | Wert ist von der Ethernet- Verbindung abhängig. |
| 3 | Get | Physical Address | ARRAY of 6 UINTs | MAC-Adresse | MAC-ID des Feldbuskoppl ers |
| 6 | Get/Set | Interface Control | STRUCT of: | Konfiguration der physikalischen Schnittstelle | - |
| | | Control Bits | WORD | Interface-Kontrollbits Bit 0: Automatische Erkennung Bit 1: Vorgabe Duplex-Modus Bit 2 – 15: Reserviert | 1 (0x0001) |
| | | Forced Interface Speed | UINT | Für die Schnittstelle vorgegebene Geschwindigkeit | 0 (0x0000) |
| 7 | Get | Interface Type | USINT | Schnittstellentyp: Wert 0: Unbekannt Wert 1: Interne Schnittstelle, zum Beispiel bei einem integrierten Switch. Wert 2: Twistet-Pair (z. B. 100Base-TX). Wert 3: Glasfaser (z. B. 100Base-FX). Wert 4 – 256: Reserviert | 2 (0x02) |
| 8 | Get | Interface State | USINT | Schnittstellenstatus: Wert 0: Unbekannt Wert 1: Schnittstelle aktiv und zum Senden/Empfangen bereit. Wert 2: Schnittstelle deaktiviert. Wert 3: Schnittstelle getestet. Wert 4 – 256: Reserviert | - |

Tab. 30: Klasse „Ethernet Link Object“: Instanz-Attribut 2

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------------|--------------|--|-------------|
| 9 | Get/Set | Admin State | USINT | Verwaltungsstatus: Wert 0: Reserviert Wert 1: Schnittstelle aktivieren. Wert 2: Schnittstelle deaktivieren. Ist diese die einzige CIP-Schnittstelle, so wird eine Anforderung zum Deaktivieren mit einem Fehler quittiert (Fehlercode 0x09) Wert 3 – 256: Reserviert | 1 (0x01) |
| 10 | Get | Interface Lable | SHORT_STRING | Lesbare Identifikation | „Port 2“ |

Instanz-Attribut 3

Tab. 31: Klasse „Ethernet Link Object“: Instanz-Attribut 3

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|------------------------|------------------|---|--|
| 1 | Get | Interface Speed | UDINT | Übertragungsgeschwindigkeit | 100 (0x64) |
| 2 | Get | Interface Flags | DWORD | Interface Konfigurations-/ Statusinformationen Bit 0: Link-Status Bit 1: Halb-/Vollduplex Bit 2 – 4: Erkennungsstatus Bit 5: Manuelle Ein-stellungen erfordern Reset Bit 6: Lokaler Hardwarefehler Bit 7 – 31: Reserviert | Wert ist von der Ethernet-Verbindung abhängig. |
| 3 | Get | Physical Address | ARRAY of 6 UINTs | MAC-Adresse | MAC-ID des Feldbuskopplers |
| 6 | Get | Interface Control | STRUCT of: | Konfiguration der physikalischen Schnittstelle | |
| | | Control Bits | WORD | Interface-Kontrollbits Bit 0: Automatische Erkennung Bit 1: Vorgabe Duplex-Modus Bit 2 – 15: Reserviert | 0x0002 |
| | | Forced Interface Speed | UINT | Für die Schnittstelle vorgegebene Geschwindigkeit | 100 (0x0064) |
| 7 | Get | Interface Type | USINT | Schnittstellentyp: Wert 0: Unbekannt Wert 1: Interne Schnittstelle, zum Beispiel bei einem integrierten Switch. Wert 2: Twistet-Pair (z. B. 100Base-TX). Wert 3: Glasfaser (z. B. 100Base-FX). Wert 4 – 256: Reserviert | 1 (0x01) |

Tab. 31: Klasse „Ethernet Link Object“: Instanz-Attribut 3

| Attribut-ID | Zugriff | Name | Datentyp | Beschreibung | Defaultwert |
|-------------|---------|-----------------|--------------|---|-------------------|
| 8 | Get | Interface State | USINT | Schnittstellenstatus: Wert 0: Unbekannt Wert 1: Schnittstelle aktiv und zum Senden/Empfangen bereit. Wert 2: Schnittstelle deaktiviert. Wert 3: Schnittstelle getestet. Wert 4 – 256: Reserviert | - |
| 9 | Get | Admin State | USINT | Verwaltungsstatus: Wert 0: Reserviert Wert 1: Schnittstelle aktivieren. Wert 2: Schnittstelle deaktivieren. Ist dieses die einzige CIP-Schnittstelle, so wird eine Anforderung zum Deaktivieren mit einem Fehler quittiert (Fehlercode 0x09) Wert 3 – 256: Reserviert | 1 (0x01) |
| 10 | Get | Interface Lable | SHORT_STRING | Lesbare Identifikation | „Internal Port 3“ |

Übersicht der Dienste

Tab. 32: Klasse „Ethernet Link Object“: Dienste

| Servicecode | Service vorhanden | | Servicename | Beschreibung |
|-------------|-------------------|---------|----------------------|--|
| | Klasse | Instanz | | |
| 01 hex | Ja | Ja | Get_Attribute_All | Liefert den Inhalt aller Attribute |
| 0E hex | Ja | Ja | Get_Attribute_Single | Liefert den Inhalt des entsprechenden Attributes |
| 10hex | Nein | Ja | Set_Attribute_Single | Modifiziert einen Attribut-Wert |

4.7 WAGO-spezifische Klassen

Diese herstellerspezifische Ergänzung enthält Informationen zu den 767-Komponenten, die nicht in den CIP-Klassen enthalten sind. Die WAGO-spezifischen Klassen sind im Anhang des Handbuchs 767-1301 aufgeführt.

Tab. 33: WAGO-spezifische Klassen

| Name | Klasse | Information |
|----------------------------------|-------------------|------------------------------|
| Coupler configuration Object | 64 _{hex} | Siehe Handbuch zum 767-1301. |
| Node information class | 82 _{hex} | |
| Discrete Input Point | 65 _{hex} | |
| Discrete Input Point Extended 1 | 69 _{hex} | |
| Discrete Input Point Extended 2 | 6D _{hex} | |
| Discrete Input Point Extended 3 | 71 _{hex} | |
| Discrete Output Point | 66 _{hex} | |
| Discrete Output Point Extended 1 | 6A _{hex} | |
| Discrete Output Point Extended 2 | 6E _{hex} | |
| Discrete Output Point Extended 3 | 72 _{hex} | |
| Analog Input Point | 67 _{hex} | |
| Analog Input Point Extended 1 | 6B _{hex} | |
| Analog Input Point Extended 2 | 6F _{hex} | |
| Analog Input Point Extended 3 | 73 _{hex} | |
| Analog Output Point | 68 _{hex} | |
| Analog Output Point Extended 1 | 6C _{hex} | |
| Analog Output Point Extended 2 | 70 _{hex} | |
| Analog Output Point Extended 3 | 74 _{hex} | |

5 MODBUS

Das modulare Konzept der Serie 767 ermöglicht es, bis zu 64 externe I/O-Module an den Feldbuskoppler anzuhängen. Dieser variable Knotenaufbau sowie die große Anzahl verschiedener I/O-Module verhindern jedoch eine statische Zuordnung von Ein- und Ausgangsdaten auf feste MODBUS-Adressen. Einzige Ausnahme sind die „digitalen“ MODBUS-Dienste. Bei ihnen ist die MODBUS-Adresse identisch mit der Kanalnummer, d. h., den 47ten digitalen Eingang findet man immer an MODBUS-Adresse „46“.

Durch das Hinzufügen oder Entfernen von I/O-Modulen verändert sich der Aufbau der Prozessabbilder und damit auch die MODBUS-Adressen der einzelnen Kanäle der I/O-Module.

Die MODBUS-Kommunikation wird mit Hilfe von Dienstaufrufen durchgeführt. Dazu sendet der MODBUS-Master (Client) ein Request-Telegramm an Port 502 des MODBUS-Slaves (Server). Der MODBUS-Slave liefert das Ergebnis des Dienstaufwurfes in einem Response-Telegramm an den MODBUS-Master zurück.

Die wesentlichsten Elemente eines MODBUS-Telegramms sind:

| Begriff | Beschreibung |
|-------------------|---|
| FunctionCode (FC) | Dienstkennung: Lese- oder Schreib-Operation auf Bits oder Worte |
| Address | Startadresse der Operation |
| Count | Dienstabhängig die Anzahl der Bits oder Worte |
| [Data] | Prozessdaten |

Die Dienstkennung bzw. der „FunctionCode“ (FC) bestimmt zunächst, ob es sich um eine Lese- oder Schreib-Operation handelt. Zusätzlich bestimmt sie den Grunddatentyp, auf den die Operation angewendet werden soll. Damit ist auch die Bedeutung der Parameter „Address“ und „Count“ abhängig vom Funktionscode. So kann „address :=3“ für ein Bit oder ein Wort im Ein- oder Ausgangsprozessabbild stehen.

Das MODBUS-Protokoll basiert im Wesentlichen auf den folgenden Grunddatentypen:

| Datentyp | Länge | Beschreibung |
|------------------|--------|-----------------------|
| Discrete Inputs | 1 Bit | Digitale Eingänge |
| Coils | 1 Bit | Digitale Ausgänge |
| Input Register | 16 Bit | Analoge Eingangsdaten |
| Holding Register | 16 Bit | Analoge Ausgangsdaten |

Für jeden Grunddatentyp sind ein oder mehrere „FunctionCodes“ definiert.

Obwohl digitale und analoge Prozessdaten des Feldbuskopplers und I/O-Module in einem Prozessabbild zusammengefasst sind, erreichen Sie mit den „digitalen“ MODBUS-Diensten an der Adresse 0 immer den ersten digitalen Ausgang bzw. Eingang. Über die Wort-Dienste erreichen Sie den ersten analogen Ausgang bzw. Eingang.

Hinweis



Der Feldbuskoppler unterstützt über MODBUS/TCP gleichzeitig bis zu 15 Socket-Verbindungen. Bei MODBUS/UDP sind es bis zu 5 Telegramme, die zwischengespeichert werden können.

5.1 Prozessabbild

Nach Inbetriebnahme des Feldbuskopplers ermittelt dieser automatisch alle angeschlossenen I/O-Module. Der Feldbuskoppler erstellt daraus ein lokales Prozessabbild, unterteilt in einen Ein- und Ausgangsbereich. In denen werden erst die Daten der analogen und anschließend die Daten der digitalen I/O-Module abgelegt.

Die Größe des Prozessabbilds ermittelt sich aus den daran angeschlossenen 767-Komponenten.

5.2 Prozessdatenaustausch

Der Datenaustausch zwischen MODBUS-TCP-Master und den I/O-Modulen wird über die im Feldbuskoppler implementierten MODBUS-Funktionen durch bit- oder wortweises Lesen und Schreiben erreicht. Im Feldbuskoppler gibt es 4 verschiedene Typen von Prozessdaten:

- Eingangsworte
- Ausgangsworte
- Eingangsbits
- Ausgangsbits

5.3 Zugriff auf das Prozessabbild über MODBUS-Funktionen

Die folgende Tabelle beschreibt die Zugriffsarten, mit denen Sie auf logische Adressbereiche zugreifen.

Tabelle 34: MODBUS-Funktionscodes (FC)

| FC | Name | Beschreibung |
|------|-------------------------------|--|
| FC1 | Read coils | Rücklesen mehrerer digitaler Ausgangswerte |
| FC2 | Read inputs discrete | Lesen mehrerer digitaler Ein- und Ausgangswerte |
| FC3 | Read holding registers | Lesen mehrerer analoger Ein- und Ausgangswerte |
| FC4 | Read input registers | Lesen mehrerer analoger Ein- und Ausgangswerte |
| FC5 | Write coil | Schreiben eines einzelnen digitalen Ausgangswerts |
| FC6 | Write single register | Schreiben eines einzelnen analogen Ausgangswerts |
| FC15 | Force multiple coils | Schreiben mehrerer digitaler Ausgangswerte |
| FC16 | Write multiple registers | Schreiben mehrerer analoger Ausgangswerte |
| FC22 | Mask Write Register | Schreiben einzelner Bits in einem Register mithilfe einer UND- bzw. ODER-Maske |
| FC23 | Read/write multiple registers | Schreib- und Leseoperation auf analoge Ein- und Ausgangswerte |

Hinweis



Bei der Adressierung auf das verwendete Zahlensystem achten!

Die aufgeführten Beispiele verwenden als Zahlenformat das Hexadezimalsystem (Bsp.: 0x000). Die Adressierung beginnt mit 0. Je nach Software und Steuerung kann das Format und der Beginn der Adressierung variieren. Alle Adressen sind in diesem Fall dementsprechend umzurechnen.

5.3.1 Registerdienste

Mit den Registerdiensten ermitteln und/oder verändern Sie die Zustände von Analogein- und ausgangsmodule mittels der Funktionscodes FC3, FC4 und FC6, FC16, FC22 und FC23.

Tabelle 35: Lesen von Analogeingangsklemmen mittels FC3, FC4, FC23

| MODBUS-Adresse | | Beschreibung |
|-----------------|---------------|---|
| HEX | DEZ | |
| 0x0000 – 0x00FF | 0 – 255 | Lesen analoger oder digitaler Eingangswerte (Teil 1). Physikalischer Adressraum der Eingangsdaten von 256 Wörtern. |
| 0x0100 – 0x01FF | 256 – 511 | Ungültiger Bereich („Illegal data address“) |
| 0x0200 – 0x02FF | 512 – 767 | Zurücklesen analoger oder digitaler Ausgangswerte (Teil 1). Physikalischer Größe des Adressraums: 256 Wörter. |
| 0x0300 – 0x0FFF | 768 – 4095 | Ungültiger Bereich („Illegal data address“) |
| 0x1000 – 0x2FFF | 4096 – 12287 | Konfigurationsregister. Siehe dazu Kapitel 5.3.3. |
| 0x3000 – 0x5FFF | 12288 – 24575 | Ungültiger Bereich („Illegal data address“) |
| 0x6000 – 0x62FB | 24576 – 25339 | Lesen der Eingangsdaten (Teil 2). Zusätzlicher Adressraum von 764 Wörtern. |
| 0x62FC – 0x6FFF | 25340 – 28671 | Ungültiger Bereich („Illegal data address“) |
| 0x7000 – 0x72FB | 28672 – 29435 | Lesen der Ausgangsdaten (Teil 2). Zusätzlicher Adressraum von 764 Wörtern. |

Tabelle 36: Schreiben von Analogausgangsklemmen mittels FC6, FC16, FC22, FC23

| MODBUS-Adresse | | Beschreibung |
|-----------------|-----------------|---|
| HEX | DEZ | |
| 0x0000 – 0x00FF | 0 – 255 | Schreiben analoger oder digitaler Ausgangswerte (Teil 1). Physikalischer Adressraum der Ausgangsdaten von 256 Wörtern. |
| 0x0100 – 0x01FF | 256 – 511 | Ungültiger Bereich („Illegal data address“) |
| 0x0200 – 0x02FF | 512 – 767 | Schreiben analoger oder digitaler Ausgangswerte (Teil 1). Physikalische Größe des Adressraums: 256 Wörter. Hinweis: Hierüber schreiben Sie dieselben Ausgangswerte des Adressbereichs 0x0000 – 0x00FF. |
| 0x0300 – 0x0FFF | 768 – 4095 | Ungültiger Bereich („Illegal data address“) |
| 0x1000 – 0x2FFF | 4096 – 12287 | Konfigurationsregister. Siehe dazu Kapitel 5.3.3. |
| 0x3000 – 0x5FFF | 12288 – 24575 | Ungültiger Bereich („Illegal data address“) |
| 0x6000 – 0x62FB | 24576 – 25339 | Schreiben der Ausgangsdaten (Teil 2). Zusätzlicher Adressraum von 764 Wörtern. |
| 0x62FC – 0x6FFF | 25340 – 28671 | Ungültiger Bereich („Illegal data address“) |
| 0x7000 – 0x72FB | (28672 – 29435) | Schreiben der Ausgangsdaten (Teil 2). Zusätzlicher Adressraum von 764 Wörtern. Hinweis: Hierüber schreiben Sie dieselben Ausgangswerte des Adressbereichs 0x6000 – 0x62FC. |
| 0x72FC – 0x7FFF | 29436 – 32767 | Ungültiger Bereich („Illegal data address“) |

5.3.2 Bitdienste

Mit den digitalen Bitdiensten ermitteln und/oder verändern Sie die Zustände von Digitalein- und ausgangsmodule mittels der Funktionscodes FC1, FC2 und FC5, FC15.

Tabelle 37: Lesen von Digitaleingangsklemmen mittels FC1, FC2

| MODBUS-Adresse | | Beschreibung |
|-----------------|---------------|---|
| HEX | DEZ | |
| 0x0000 – 0x01FF | 0 – 511 | Lesen der ersten 512 Eingangswerte |
| 0x0200 – 0x03FF | 512 – 1023 | Lesen der ersten 512 Ausgangswerte |
| 0x0400 – 0x7FFF | 1024 – 32767 | Ungültiger Bereich („Illegal data address“) |
| 0x8000 – 0x85F7 | 32768 – 34295 | Lesen der Eingangswerte im Bereich von 513 bis 2039 |
| 0x85F8 – 0x8FFF | 34296 – 36863 | Ungültiger Bereich („Illegal data address“) |
| 0x9000 – 0x95F7 | 36864 – 38391 | Lesen der Ausgangswerte im Bereich von 513 bis 2039 |
| 0xA000 – 0xFFFF | 40960 – 65535 | Ungültiger Bereich („Illegal data address“) |

Tabelle 38: Schreiben von Digitalausgangsklemmen mittels FC5, FC15

| MODBUS-Adresse | | Beschreibung |
|-----------------|---------------|---|
| HEX | DEZ | |
| 0x0000 – 0x01FF | 0 – 511 | Schreiben der ersten 512 Ausgangswerte. |
| 0x0200 – 0x03FF | 512 – 1023 | Schreiben der ersten 512 Ausgangswerte. |
| 0x0400 – 0x7FFF | 1024 – 32767 | Ungültiger Bereich („Illegal data address“) |
| 0x8000 – 0x85F7 | 32768 – 34295 | Schreiben der Ausgangswerte im Bereich von 513 bis 2039. |
| 0x85F8 – 0x8FFF | 34296 – 36863 | Ungültiger Bereich („Illegal data address“) |
| 0x9000 – 0x95F7 | 36864 – 38391 | Schreiben der Ausgangswerte im Bereich von 513 bis 2039. |
| 0x95F8 – 0xFFFF | 38392 – 65535 | Ungültiger Bereich („Illegal data address“) |

5.3.3 Konfigurationsregister

Mittels der Konfigurationsregister können Sie den Feldbuskoppler konfigurieren und Informationen über diesen auslesen. Auf die Konfigurationsregister greifen Sie mittels der MODBUS-Funktionen FC4 „Read Input Registers“ und FC6 „Write Single Register“ auf die Register.

Tabelle 39: Konfigurationsregister

| Adresse | Zugriff | Länge (Wort) | Beschreibung |
|---------|---------|--------------|--|
| 0x1000 | R/W | 1 | Watchdog-Time lesen/schreiben |
| 0x1001 | R/W | 1 | Watchdog-Codiermaske 1-16 |
| 0x1002 | R/W | 1 | Watchdog-Codiermaske 17-32 |
| 0x1003 | R/W | 1 | Watchdog-Trigger |
| 0x1004 | R | 1 | Minimale Triggerzeit |
| 0x1005 | R/W | 1 | Watchdog stoppen (Schreibsequenz 0xAAAA, 0x5555) |
| 0x1006 | R | 1 | Watchdog-Status |
| 0x1007 | R/W | 1 | Restart des Watchdog (Schreibsequenz 0x1) |
| 0x1008 | RW | 1 | Watchdog stoppen (Schreibsequenz 0x55AA oder 0xAA55) |
| 0x1009 | R/W | 1 | MODBUS- und http-Verbindung beenden bei einem Watchdog-Timeout |
| 0x100A | R/W | 1 | Watchdog-Konfiguration |
| 0x100B | W | 1 | Watchdog-Parameter speichern |
| 0x1020 | R | 1 – 2 | Fehlergruppe mittels LED anzeigen |
| 0x1021 | R | 1 | Fehlerargument und Fehlercode mittels LED anzeigen |
| 0x1022 | R | 1 – 4 | Anzahl analoger Ausgangsdaten im Prozessabbild (in Bits) |
| 0x1023 | R | 1 – 3 | Anzahl analoger Eingangsdaten im Prozessabbild (in Bits) |
| 0x1024 | R | 1 – 2 | Anzahl digitaler Ausgangsdaten im Prozessabbild (in Bits) |
| 0x1025 | R | 1 | Anzahl digitaler Eingangsdaten im Prozessabbild (in Bits) |
| 0x1028 | R/W | 1 | Bootkonfiguration |
| 0x1029 | R | 9 | MODBUS-TCP-Statistik |
| 0x102A | R | 1 | Anzahl der TCP-Verbindungen |
| 0x1030 | R/W | 1 | Konfiguration MODBUS/TCP-Timeout |
| 0x1031 | R | 1 | Lesen der MAC-ID des Feldbuskopplers |
| 0x2000 | R | 1 | Konstante 0x0000 |
| 0x2001 | R | 1 | Konstante 0xFFFF |
| 0x2002 | R | 1 | Konstante 0x1234 |
| 0x2003 | R | 1 | Konstante 0xAAAA |
| 0x2004 | R | 1 | Konstante 0x5555 |
| 0x2005 | R | 1 | Konstante 0x7FFF |
| 0x2006 | R | 1 | Konstante 0x8000 |
| 0x2007 | R | 1 | Konstante 0x3FFF |
| 0x2008 | R | 1 | Konstante 0x4000 |
| 0x2010 | R | 1 | Firmware-Index |
| 0x2011 | R | 1 | Serienbezeichnung (767) |
| 0x2012 | R | 1 | Feldbustyp (1301) |
| 0x2013 | R | 1 | Firmware-Versionen (Major Revision) |
| 0x2014 | R | 1 | Firmware-Versionen (Minor Revision) |
| 0x2020 | R | 16 | Kurzbeschreibung des Feldbuskopplers |
| 0x2021 | R | 8 | Kompilierzeit der Firmware |
| 0x2022 | R | 8 | Compile-Datum der Firmware |
| 0x2023 | R | 32 | Version des Firmware-Loaders |

Tabelle 39: Konfigurationsregister

| Adresse | Zugriff | Länge (Wort) | Beschreibung |
|---------|---------|--------------|---|
| 0x2030 | R | 65 | Beschreibung der angeschlossenen I/O-Module (0 – 64) |
| 0x2040 | W | 1 | Software-Reset (Schreibsequenz 0x55AA oder 0xAA55) |
| 0x2041 | W | 1 | Formatieren des User-Laufwerks |
| 0x2042 | W | 1 | HTML-Seiten aus der Firmware extrahieren |
| 0x2043 | W | 1 | Werkseinstellungen |
| 0x2099 | R/W | 1 | MODBUS-Versionswechsel (Kompilitätsmodus zum Release 1) |

5.3.4 Watchdog-Verhalten

Zur Überwachung der TCP-Verbindung wird von der übergeordneten Steuerung eine Zeitfunktion (Time-out) in dem Feldbuskoppler zyklisch angestoßen. Bei fehlerfreier Kommunikation kann diese Zeit ihren Endwert nicht erreichen, weil sie zuvor immer wieder neu gestartet wird. Falls diese Zeit abgelaufen sein sollte, liegt ein Feldbusausfall vor. In diesem Fall antwortet der Feldbuskoppler allen folgenden MODBUS „TCP/IP Requests“ mit dem Exceptioncode 0x0004 (Slave Device Failure). Im Feldbuskoppler sind gesonderte Register für die Ansteuerung und für die Statusabfrage des Watchdog durch die übergeordnete Steuerung vorhanden.

Nach dem Einschalten der Versorgungsspannung ist der Watchdog noch nicht aktiviert. Zunächst ist der Time-out-Wert festzulegen (Register 0x1000). Der Watchdog kann dadurch aktiviert werden, indem im Register (0x1001) ein von 0 abweichender Wert geschrieben wird.

5.3.4.1 Watchdog-Register

Tabelle 40: Registeradresse 0x1000

| Registeradresse 0x1000 (4096 _{dez}) | |
|---|--|
| Wert | Watchdog time |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Dieses Register speichert den Wert für die Zeitüberschreitung (Time-out). Damit der Watchdog gestartet werden kann, muss der Vorgabewert auf einen Wert ungleich Null geändert werden. Die Zeit wird in Vielfachen von 100 ms gesetzt, 0x0009 bedeutet also eine Time-out-Zeit von 0.9 s. Dieser Wert kann bei laufendem Watchdog nicht geändert werden. Es gibt keinen Code, durch den der aktuelle Datenwert nochmals geschrieben werden kann, während der Watchdog aktiv ist. |

Tabelle 41: Registeradresse 0x1001

| Registeradresse 0x1001 (4097 _{dez}) | |
|---|--|
| Wert | Watchdog-Funktion Codiermaske, Funktionscode 1...16 |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Mittels dieser Maske sind die Funktionscodes einstellbar, um die Watchdog-Funktion zu triggern. Mit der „1“ kann der Funktionscode ausgewählt werden (= $2^{(\text{Funktionscode}-1)} + 2^{(\text{Funktionscode}-1)} + \dots$) Beispiel: Funktionscode 5 $\rightarrow 2^{(5-1)} = 2^4 \rightarrow$ Bit 4 wird auf 1 gesetzt Ein Wert ungleich Null startet die Watchdog-Funktion. Wenn in die Maske ausschließlich Codes von nicht unterstützten Funktionen eingetragen werden, startet der Watchdog nicht. Ein bestehender Fehler wird zurückgesetzt und das Prozessabbild kann wieder beschrieben werden. Auch hier kann bei laufendem Watchdog keine Änderung erfolgen. Während der Watchdog aktiv ist, gibt es keinen Code, durch den der aktuelle Datenwert nochmals geschrieben werden kann. |

Tabelle 42: Registeradresse 0x1002

| Registeradresse 0x1002 (4098 _{dez}) | |
|---|---|
| Wert | Watchdog-Funktion Codiermaske, Funktionscode 17...32 |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Gleiche Funktion wie zuvor, aber mit den Funktionscodes 17 bis 32. Diese Codes werden nicht unterstützt. Dieses Register sollte deshalb auf dem Vorgabewert belassen werden. Es gibt keinen Ausnahmecode durch den der aktuelle Datenwert nochmals geschrieben werden kann, während der Watchdog aktiv ist. |

Tabelle 43: Registeradresse 0x1003

| Registeradresse 0x1003 (4099_{dez}) | |
|--|---|
| Wert | Watchdog-Trigger |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Dieses Register wird für eine alternative Trigger-Methode benutzt. Durch das Schreiben unterschiedlicher Werte in dieses Register wird der Watchdog getriggert. Aufeinanderfolgende Werte müssen sich in der Größe unterscheiden. Das Schreiben eines Werts ungleich Null startet den Watchdog. Ein Watchdog-Fehler wird zurückgesetzt und das Schreiben der Prozessdaten wird wieder ermöglicht. |

Tabelle 44: Registeradresse 0x1004

| Registeradresse 0x1004 (4100_{dez}) | |
|--|--|
| Wert | Minimale aktuelle Trigger-Zeit |
| Zugang | Lesen/schreiben |
| Standard | 0xFFFF |
| Beschreibung | Dieses Register speichert die aktuell kleinste Watchdog-Trigger-Zeit. Bei einem Triggern des Watchdogs, wird der gespeicherte Wert mit dem aktuellen verglichen. Ist der aktuelle Wert kleiner als der gespeicherte, wird dieser durch den aktuellen Wert ersetzt. Die Einheit ist 100 ms/Digit. Durch das Schreiben neuer Werte wird der gespeicherte Wert geändert. Dies hat keine Auswirkung auf den Watchdog. Der Wert 0x0000 ist nicht erlaubt. |

Tabelle 45: Registeradresse 0x1005

| Registeradresse 0x1005 (4101_{dez}) | |
|--|--|
| Wert | Watchdog stoppen |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Wird der Wert 0xAAAA gefolgt von dem Wert 0x5555 in dieses Register geschrieben, stoppt der Watchdog. Die Watchdog-Fehlerreaktion wird gesperrt. Ein Watchdog-Fehler wird zurückgesetzt und das Schreiben auf die Prozessdaten wird wieder ermöglicht. |

Tabelle 46: Registeradresse 0x1006

| Registeradresse 0x1006 (4102_{dez}) | |
|--|--|
| Wert | Während Watchdog läuft |
| Zugang | Lesen |
| Standard | 0x0000 |
| Beschreibung | Aktueller Watchdog-Status bei 0x0000: Watchdog nicht aktiv bei 0x0001: Watchdog aktiv bei 0x0002: Watchdog abgelaufen |

Tabelle 47: Registeradresse 0x1007

| Registeradresse 0x1007 (4103 _{dez}) | |
|---|---|
| Wert | Watchdog neu starten |
| Zugang | Lesen/schreiben |
| Standard | 0x0001 |
| Beschreibung | Schreiben von 0x1 in das Register startet den Watchdog wieder. Wurde der Watchdog vor dem Überlauf gestoppt, wird er nicht wieder gestartet. |

Tabelle 48: Registeradresse 0x1008

| Registeradresse 0x1008 (4104 _{dez}) | |
|---|--|
| Wert | Watchdog einfach anhalten |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Durch Schreiben der Werte 0x0AA55 oder 0X55AA wird der Watchdog angehalten, falls er aktiv war. Die Watchdog-Fehlerreaktion wird vorübergehend deaktiviert. Ein anstehender Watchdog-Fehler wird zurückgesetzt und ein Schreiben ins Watchdog-Register ist wieder möglich. |

Tabelle 49: Registeradresse 0x1009

| Registeradresse 0x1009 (4105 _{dez}) | |
|---|--|
| Wert | MODBUS-Socket nach Watchdog-Time-out schließen |
| Zugang | Lesen/schreiben |
| Beschreibung | 0: MODBUS-Socket wird nicht geschlossen 1: MODBUS-Socket wird geschlossen |

Tabelle 50: Registeradresse 0x100A

| Registeradresse 0x100A (4106 _{dez}) | |
|---|---|
| Wert | Alternativer Watchdog |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Schreiben eines Zeitwertes in Register 0x1000 Register 0x100A = 0x0001: Watchdog wird aktiv geschaltet Mit dem ersten MODBUS-Telegramm wird der Watchdog gestartet. Der Watchdog wird mit jedem MODBUS/TCP-Befehl getriggert. Nach Ablauf der Watchdog-Zeit werden alle Ausgänge auf Null gesetzt. Die Ausgänge können durch erneutes Schreiben wieder gesetzt werden. Das Register 0x100A ist remanent und damit auch das Register 0x1000. Bei eingeschaltetem Watchdog lässt sich der Zeitwert in Register 0x1000 nicht mehr ändern. |

Tabelle 51: Registeradresse 0x100B

| Registeradresse 0x100B (4107 _{dez}) | |
|---|---|
| Wert | Save-Watchdog-Parameter |
| Zugang | Schreiben |
| Standard | 0x0000 |
| Beschreibung | Mit Schreiben von „1“ in Register 0x100B werden die Register 0x1000, 0x1001, 0x1002 auf remanent gesetzt. |

5.3.4.2 Diagnoseregister

Folgende Register können gelesen werden, um einen Fehler des Feldbusknotens zu bestimmen:

Tabelle 52: Registeradresse 0x1020

| Registeradresse 0x1020 (4128 _{dez}) | |
|---|--------------------------|
| Wert | LED Error Group |
| Zugang | Lesen |
| Beschreibung | Angabe der Gruppennummer |

Tabelle 53: Registeradresse 0x1021

| Registeradresse 0x1021 (4129 _{dez}) | |
|---|--|
| Wert | LED Error Code/Argument |
| Zugang | Lesen |
| Beschreibung | Angabe des Fehlercodes und Fehlerarguments |

5.3.4.3 Konfigurationsregister

Folgende Register können gelesen werden, um die Konfiguration der angeschlossenen I/O-Module zu bestimmen:

Tabelle 54: Registeradresse 0x1022

| Registeradresse 0x1022 (4130 _{dez}) | |
|---|---|
| Wert | CnfLen.AnalogOut |
| Zugang | Lesen |
| Beschreibung | Anzahl E/A-Bits bei den Prozessdatenworten der Ausgänge |

Tabelle 55: Registeradresse 0x1023

| Registeradresse 0x1023 (4131 _{dez}) | |
|---|---|
| Wert | CnfLen.AnalogInp |
| Zugang | Lesen |
| Beschreibung | Anzahl E/A-Bits bei den Prozessdatenworten der Eingänge |

Tabelle 56: Registeradresse 0x1024

| Registeradresse 0x1024 (4132 _{dez}) | |
|---|---|
| Wert | CnfLen.DigitalOut |
| Zugang | Lesen |
| Beschreibung | Anzahl E/A-Bits bei den Prozessdatenbits der Ausgänge |

Tabelle 57: Registeradresse 0x1025

| Registeradresse 0x1025 (4133 _{dez}) | |
|---|---|
| Wert | CnfLen.DigitalInp |
| Zugang | Lesen |
| Beschreibung | Anzahl E/A-Bits bei den Prozessdatenbits der Eingänge |

Tabelle 58: Registeradresse 0x1028

| Registeradresse 0x1028 (4136_{dez}) | |
|--|--|
| Wert | Bootoptions |
| Zugang | Lesen/schreiben |
| Beschreibung | Bootkonfiguration: 1: BootP 2: DHCP 4: EEPROM |

Tabelle 59: Registeradresse 0x1029

| Registeradresse 0x1029 (4137_{dez}) mit bis zu 9 Worten | |
|--|---|
| Wert | MODBUS/TCP-Statistik |
| Zugang | Lesen/schreiben |
| Beschreibung | <p>1 Wort SlaveDeviceFailure → Klemmenbusfehler, Feldbusfehler bei eingeschaltetem Watchdog</p> <p>1 Wort BadProtocol → Fehler im MODBUS/TCP-Header</p> <p>1 Wort BadLength → Falsche Telegrammlänge</p> <p>1 Wort BadFunction → Ungültiger Funktionscode</p> <p>1 Wort Bad Address → Ungültige Registeradresse</p> <p>1 Wort BadData → Ungültiger Wert</p> <p>1 Wort TooManyRegisters → Anzahl der zu bearbeitenden Register zu groß, Lesen/Schreiben 125/100</p> <p>1 Wort TooManyBits → Anzahl der zu bearbeitenden Coils zu groß, Lesen/Schreiben 2000/800</p> <p>1 Wort ModTcpMessageCounter → Anzahl der empfangenen MODBUS/TCP-Telegramme</p> <p>Durch Schreiben von 0xAA55 oder 0x55AA wird das Register zurückgesetzt.</p> |

Tabelle 60: Registeradresse 0x102A

| Registeradresse 0x102A (4138_{dez}) mit bis zu 1 Wort | |
|--|-----------------------------|
| Wert | MODBUS/TCP-Connections |
| Zugang | Lesen |
| Beschreibung | Anzahl der TCP-Verbindungen |

Tabelle 61: Registeradresse 0x1030

| Registeradresse 0x1030 (4144_{dez}) mit bis zu 1 Wort | |
|--|--|
| Wert | Konfiguration MODBUS/TCP-Time-out |
| Zugang | Lesen/schreiben |
| Standard | 0x0000 |
| Beschreibung | Dieses Register speichert den Wert für eine TCP-Verbindungsüberwachung. Die Zeitbasis ist 100 ms, der Minimalwert ist 10 ms. Geöffnete TCP-Verbindungen werden automatisch geschlossen, wenn die eingetragene Zeit je Verbindung überschritten wurde. Der Watchdog wird mit einer Anfrage auf der Verbindung getriggert. |

Tabelle 62: Registeradresse 0x1031

| Registeradresse 0x1031 (4145_{dez}) mit bis zu 3 Worten | |
|--|--------------------------------------|
| Wert | Lesen der MAC-ID des Feldbuskopplers |
| Zugang | Lesen |
| Beschreibung | Ausgabe der MAC-ID, Länge 3 Worte |

Tabelle 63: Registeradresse 0x2030

| Registeradresse 0x2030 (8240_{dez}) mit bis zu 65 Worten | |
|---|---|
| Wert | Beschreibung der angeschlossenen I/O-Modul |
| Zugang | Lesen der I/O-Module 0...64 |
| Beschreibung | Ausgabe: z.B. „3801“ für das Eingangsmodul 767-3801 |

Tabelle 64: Registeradresse 0x2040

| Registeradresse 0x2040 (8256_{dez}) | |
|--|---|
| Wert | Ausführen eines Software-Resets |
| Zugang | Schreiben (Schreibsequenz 0xAA55 oder 0x55AA) |
| Beschreibung | Durch Schreiben der Werte 0xAA55 oder 0x55AA führt der Feldbuskoppler einen Neustart durch. |

Tabelle 65: Registeradresse 0x2041

| Registeradresse 0x2041 (8257_{dez}) | |
|--|---|
| Wert | Flash-Format |
| Zugang | Schreiben (Schreibsequenz 0xAA55 oder 0x55AA) |
| Beschreibung | Das Flash-Dateisystem wird neu formatiert |

Tabelle 66: Registeradresse 0x2042

| Registeradresse 0x2042 (8258_{dez}) | |
|--|---|
| Wert | Dateien extrahieren |
| Zugang | Schreiben (Schreibsequenz 0xAA55 oder 0x55AA) |
| Beschreibung | Die Standarddateien (HTML-Seiten) des Feldbuskopplers werden extrahiert und in das Flash-Dateisystem geschrieben. |

Tabelle 67: Registeradresse 0x2043

| Registeradresse 0x2043 (8259_{dez}) | |
|--|--------------------|
| Wert | 0x55AA |
| Zugang | Schreiben |
| Beschreibung | Werkseinstellungen |

5.3.4.4 Firmware-Informationsregister

Folgende Register werden genutzt, um Informationen zur Firmware des Feldbuskopplers auszulesen:

Tabelle 68: Registeradresse 0x2010

| Registeradresse 0x2010 (8208_{dez}) mit bis zu 1 Wort | |
|--|--|
| Wert | Revision |
| Zugang | Lesen |
| Beschreibung | Firmware-Index, z. B. 0005 für Version 5 |

Tabelle 69: Registeradresse 0x2011

| Registeradresse 0x2011 (8209_{dez}) mit bis zu 1 Wort | |
|--|--|
| Wert | Series code |
| Zugang | Lesen |
| Beschreibung | WAGO-Baureihennummer, z. B. 0767 für WAGO-SPEEDWAY 767 |

Tabelle 70: Registeradresse 0x2012

| Registeradresse 0x2012 (8210_{dez}) mit bis zu 1 Wort | |
|--|---|
| Wert | Item number |
| Zugang | Lesen |
| Beschreibung | WAGO-Bestellnummer, z. B. 1301 für den Feldbuskoppler 767-1301 etc. |

Tabelle 71: Registeradresse 0x2013

| Registeradresse 0x2013 (8211_{dez}) mit bis zu 1 Wort | |
|--|---------------------------------|
| Wert | Major sub item code |
| Zugang | Lesen |
| Beschreibung | Firmware-Version Major-Revision |

Tabelle 72: Registeradresse 0x2014

| Registeradresse 0x2014 (8212_{dez}) mit bis zu 1 Wort | |
|--|---------------------------------|
| Wert | Minor sub item code |
| Zugang | Lesen |
| Beschreibung | Firmware-Version Minor-Revision |

Tabelle 73: Registeradresse 0x2020

| Registeradresse 0x2020 (8224_{dez}) mit bis zu 16 Worten | |
|---|--|
| Wert | Description |
| Zugang | Lesen |
| Beschreibung | Informationen zum Feldbuskoppler, 16 Worte |

Tabelle 74: Registeradresse 0x2021

| Registeradresse 0x2021 (8225_{dez}) mit bis zu 8 Worten | |
|--|-----------------------------------|
| Wert | Description |
| Zugang | Lesen |
| Beschreibung | Zeit des Firmwarestandes, 8 Worte |

Tabelle 75: Registeradresse 0x2022

| Registeradresse 0x2022 (8226_{dez}) mit bis zu 8 Worten | |
|--|------------------------------------|
| Wert | Description |
| Zugang | Lesen |
| Beschreibung | Datum des Firmwarestandes, 8 Worte |

Tabelle 76: Registeradresse 0x2023

| Registeradresse 0x2023 (8227_{dez}) mit bis zu 32 Worten | |
|---|---|
| Wert | Description |
| Zugang | Lesen |
| Beschreibung | Info über Programmierung der Firmware, 32 Worte |

5.3.4.5 Konstantenregister

Folgende Register enthalten Konstanten, die genutzt werden können, um die Kommunikation mit dem Master zu testen:

Tabelle 77: Registeradresse 0x2000

| Registeradresse 0x2000 (8192 _{dez}) | |
|---|--------------------|
| Wert | Null |
| Zugang | Lesen |
| Beschreibung | Konstante mit Null |

Tabelle 78: Registeradresse 0x2001

| Registeradresse 0x2001 (8193 _{dez}) | |
|---|---|
| Wert | Einsen |
| Zugang | Lesen |
| Beschreibung | Konstante mit Einsen. <ul style="list-style-type: none"> • „-1“, wenn Konstante als „signed int“ deklariert ist • „MAXVALUE“, wenn Konstante als „unsigned int“ deklariert ist |

Tabelle 79: Registeradresse 0x2002

| Registeradresse 0x2002 (8194 _{dez}) | |
|---|--|
| Wert | 1,2,3,4 |
| Zugang | Lesen |
| Beschreibung | Konstanter Wert, zum Testen, ob High- und Low-Byte getauscht sind (Intel/Motorola Format). Sollte im Master als 0x1234 erscheinen. Erscheint 0x3412, müssen High- und Low-Byte getauscht werden. |

Tabelle 80: Registeradresse 0x2003

| Registeradresse 0x2003 (8195 _{dez}) | |
|---|--|
| Wert | Maske 1 |
| Zugang | Lesen |
| Beschreibung | Konstante, die anzeigt, ob alle Bits vorhanden sind. Wird zusammen mit Register 0x2004 genutzt. |

Tabelle 81: Registeradresse 0x2004

| Registeradresse 0x2004 (8196 _{dez}) | |
|---|--|
| Wert | Maske 1 |
| Zugang | Lesen |
| Beschreibung | Konstante, die anzeigt, ob alle Bits vorhanden sind. Wird zusammen mit Register 0x2003 genutzt. |

Tabelle 82: Registeradresse 0x2005

| Registeradresse 0x2005 (8197 _{dez}) | |
|---|---|
| Wert | Größte positive Zahl |
| Zugang | Lesen |
| Beschreibung | Konstante, um die Arithmetik zu kontrollieren |

Tabelle 83: Registeradresse 0x2006

| Registeradresse 0x2006 (8198_{dez}) | |
|--|---|
| Wert | Größte negative Zahl |
| Zugang | Lesen |
| Beschreibung | Konstante, um die Arithmetik zu kontrollieren |

Tabelle 84: Registeradresse 0x2007

| Registeradresse 0x2007 (8199_{dez}) | |
|--|---|
| Wert | Größte halbe positive Zahl |
| Zugang | Lesen |
| Beschreibung | Konstante, um die Arithmetik zu kontrollieren |

Tabelle 85: Registeradresse 0x2008

| Registeradresse 0x2008 (8200_{dez}) | |
|--|---|
| Wert | Größte halbe negative Zahl |
| Zugang | Lesen |
| Beschreibung | Konstante, um die Arithmetik zu kontrollieren |

6 Die Ethernet Schnittstelle

6.1 IP-Adressvergabe

Der Feldbuskoppler kann seine IP-Adresse dynamisch von einem Server beziehen oder mit einer statischen (festen) IP-Adresse konfiguriert werden.

Die dynamische IP-Adresszuordnung kann über das BootP- oder DHCP-Protokoll erfolgen. Dynamisch zugewiesene IP-Adressen werden nicht gespeichert. Daher ist bei jedem Einschalten des Feldbuskopplers die Anwesenheit eines BootP- oder DHCP-Servers erforderlich.

Die Konfiguration einer IP-Adresse können Sie über

- den DIP-Schalter am Feldbuskoppler,
- das Web-based Management oder
- die Konfigurationssoftware „WAGOframe“ durchführen.

Im Auslieferungszustand ist die dynamische IP-Adressvergabe über das BootP-Protokoll aktiviert.

Um mit einem Internet-Browser auf das Web-based Management zuzugreifen, benötigt der Feldbuskoppler eine bekannte IP-Adresse. Diese initiale IP-Adresse kann über BootP, DIP-Schalter oder mittels der Konfigurationssoftware WAGOframe zugewiesen werden.

Ausgehend vom Auslieferungszustand wird nachfolgend die Vergabe über DIP-Schalter und über BootP vorgestellt. Eine vollständige Liste zur IP-Adressvergabe finden Sie im Handbuch zum 767-1301.

6.1.1 Zuweisen einer festen IP-Adresse mittels DIP-Schalter

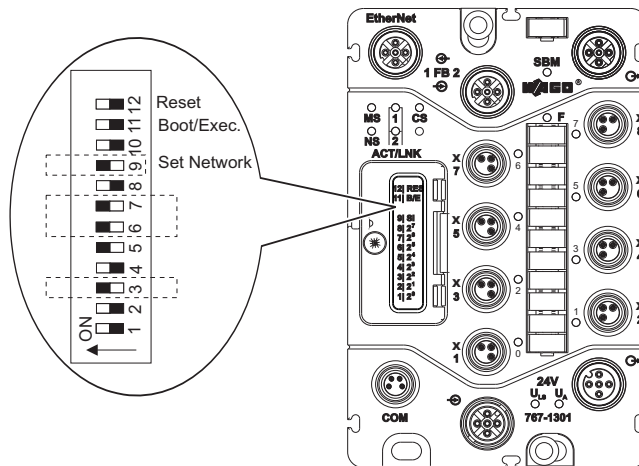
Im Auslieferungszustand sind die oberen 3 Byte (Network-ID) mit „192.168.1.“ und der Subnetzmaske 255.255.255.0 vorbelegt.

Über den DIP-Schalterblock (Schalter 1 – 8) stellen Sie das letzte Byte (Host-ID) der IP-Adresse ein. Dazu muss der Schalter 9 (Set Network) auf „On“ gestellt werden.

| Schalter | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|--------------|--------------|--------------|--------------|---------------|---------------|---------------|----------------|----------------|-----|------------------|-------|
| Binärwerte zur Adresseinstellung/ Funktionen | 2^0 (1) | 2^1 (2) | 2^2 (4) | 2^3 (8) | 2^4 (16) | 2^5 (32) | 2^6 (64) | 2^7 (128) | Set Network | - | Boot/ Execute | Reset |
| Schalterstellung | Off | Off | Off | Off | Off | Off | Off | Off | Off | Off | Off | Off |

Um beispielsweise die IP-Adresse 192.168.1.100 einzustellen, gehen Sie folgend vor:

1. Öffnen Sie die Abdeckklappe, indem Sie die Schraube mit einem Schraubendreher herausdrehen.
2. Schieben Sie den Schalter 9 (Set Network) auf „On“, um die IP-Adresse-Vergabe mittels DIP-Schalter zu aktivieren.
3. Stellen Sie für die Host-ID 100 die Schalter 3, 6, 7 auf „On“ ($2^2+2^5+2^6=4+32+64=100$).
4. Schließen Sie die Abdeckklappe wieder und schrauben Sie diese fest, um die Schutzart IP 67 einzuhalten.



Sie haben mit den acht Schaltern die Möglichkeit, Adressen von 1 – 254 einzustellen. Die Adressen 0 und 255 sind reserviert und dürfen nicht verwendet werden.

6.1.2 Zuweisen einer dynamischen IP-Adresse mittels BootP

Im Auslieferungszustand ist die IP-Adressvergabe über das BootP-Protokoll aktiviert.

Das Zuweisen der IP-Adresse mittels BOOTP ist hier exemplarisch an dem BootP-Server von WAGO erläutert.

BootP-Server konfigurieren

Um mit dem Feldbuskoppler im lokalen Netzwerk kommunizieren zu können, müssen Sie erst die MAC-Adresse des Feldbuskopplers ermitteln und ihr anschließend eine IP-Adresse zuordnen.

1. Notieren Sie sich die Hardware-Adresse (MAC-Adresse) des Feldbuskopplers. Diese finden Sie auf einem Etikett auf der Seite des Feldbuskopplers.

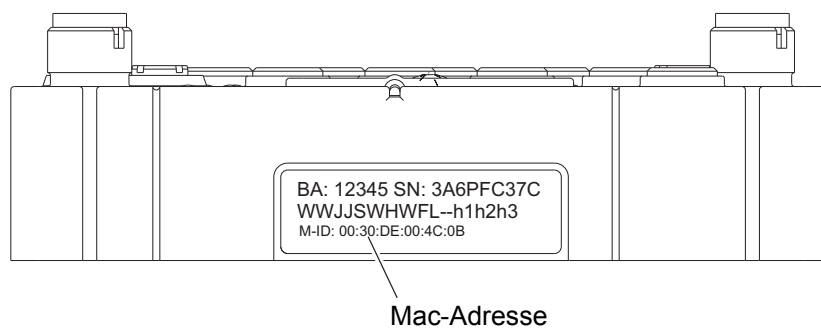


Abb. 1: Beispiel einer MAC-Adresse im Aufkleber auf der linken Seite des Feldbuskopplers

2. Klicken Sie auf die Schaltfläche **[Start]** und starten Sie den WAGO-BootP Server unter **Programme > WAGO Software > WAGO BootP Server**.
3. Öffnen Sie die Konfigurationsdatei, in der Sie die MAC-Adresse einer IP-Adresse aus dem lokalen Netzwerk zuordnen. Klicken Sie dazu im BootP-Server auf die Schaltfläche **[Edit BootPtab]** klicken. Es öffnet sich die Konfigurationsdatei.
4. Klicken Sie in die Zeile, die mit „node“ beginnt, z. B.


```
„my1301:ht=1:ha=0030DE00001C:ip=192.168.1.100:sm=255.255.255.0:“.
```
5. Ersetzen Sie die aus zwölf Zeichen bestehende MAC-Adresse hinter „:ha=“ mit der, die auf dem Etikett (13) aufgedruckt ist.

6. Geben Sie eine IP-Adresse nach „ip=“ ein. In diesem Beispiel ist dies 192.168.1.100. Trennen Sie die einzelnen Nummern durch einen Dezimalpunkt.
7. Um mehr als einen Feldbusknoten zu adressieren, fügen Sie jeweils eine Zeile mit der entsprechenden Zuordnung für jeden zusätzlichen Feldbuskoppler in der Datei bootptab.txt ein. Wiederholen Sie dazu die Handlungsschritte 5 bis 7.
8. Speichern Sie die neuen Einstellungen in der Datei „bootptab.txt“. Klicken Sie dazu in das Menü „Datei“ und wählen Sie „Sichern“.
9. Schließen Sie den Editor.

IP-Adresse mittels des WAGO-BootP-Servers vergeben

1. Zum Starten des BootP-Servers klicken Sie im geöffneten BootP-Dialogfenster auf die Schaltfläche **[Start]**. Diverse Meldungen werden im BootP-Dialogfenster angezeigt. Die Fehlermeldungen weisen darauf hin, dass einige Dienste (z. B. Port 67, Port 68) im Betriebssystem nicht definiert worden sind. Diese Fehlermeldung brauchen Sie nicht zu beachten.

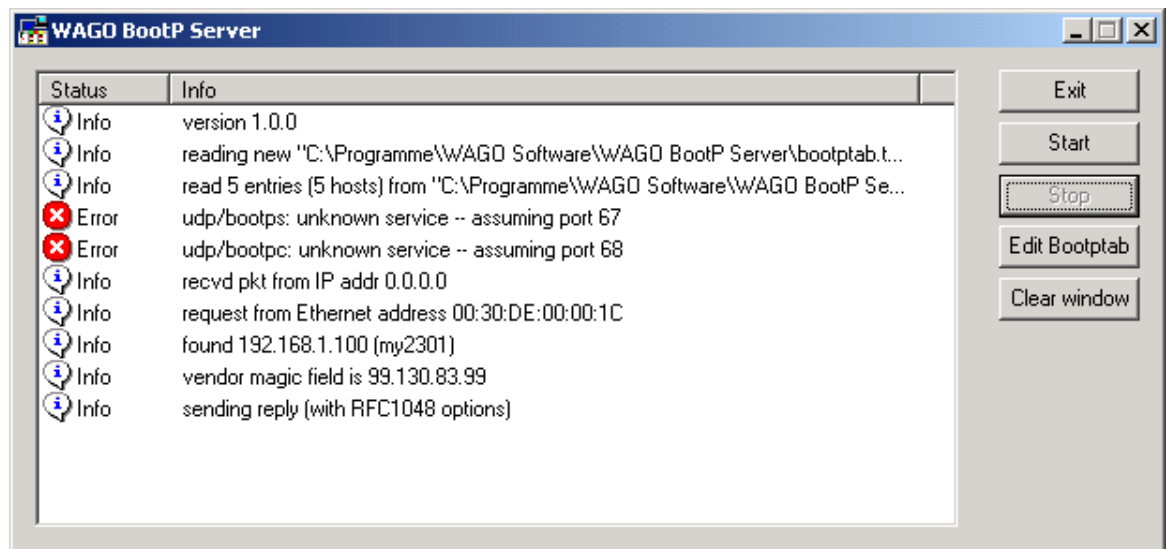


Abb. 3-10: Dialogfenster des WAGO-BootP-Servers mit Nachrichten

2. Führen Sie einen Neustart des Feldbuskopplers durch, indem Sie die den DIP-Schalter „Reset“ auf ON und danach auf OFF schieben. Oder schalten Sie die Spannung zum Feldbuskoppler für zwei Sekunden aus und anschließend wieder ein.
Es erscheint eine Anfrage (request) vom Feldbuskoppler. Der BootP-Server antwortet, dass die IP-Adresse akzeptiert wurde (keine Fehler). Die IP-Adresse ist nun vorübergehend im Feldbuskoppler vorhanden, aber nicht remanent gespeichert. Bei einem Neustart erwartet der Feldbuskoppler erneut eine IP-Adresse vom BootP-Server zu bekommen.
3. Klicken Sie auf die Schaltfläche **[Stop]** und danach auf die Schaltfläche **[Exit]**, um den BootP-Server zu schließen.
4. Um die IP-Adresse dauerhaft im Feldbuskoppler zu speichern, wählen Sie im Web-based Management in der Ansicht „TCP/IP“ die Option „Static IP“ aus. Oder nutzen Sie dazu den WAGOframe, indem Sie unter dem Parameter „IP-Adresse beziehen von“ die Auswahl „Gerätespeicher“ treffen.

6.1.3 Test der Netzwerkverbindung

Um zu überprüfen, ob Sie den Feldbuskoppler unter der von Ihnen vergebenen IP-Adresse im Netzwerk erreichen, führen Sie den Netzwerkdienst ping durch.

Öffnen Sie dazu das DOS-Fenster, indem Sie auf die Schaltfläche **[Start]** klicken und **Programme > Ausführen** wählen. Geben Sie im „Ausführen“-Dialog `cmd` ein und klicken Sie auf **[OK]**.

1. Geben Sie im DOS-Fenster den Befehl ping und die IP-Adresse des Feldbuskopplers ein: Beispiel: ping 192.168.1.100
2. Drücken Sie die **[Enter]**-Taste Ihr PC sendet eine Anfrage, die vom Feldbuskoppler beantwortet wird. Die Antwort erscheint im DOS-Fenster. Wenn die Fehlermeldung „Timeout“ erscheint, hat der Feldbuskoppler sich nicht ordnungsgemäß gemeldet. Überprüfen Sie bitte Ihre Netzwerkeinstellung.

```

C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

U:\>ping 192.168.1.100

Ping wird ausgeführt für 192.168.1.100 mit 32 Bytes Daten:

Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128
Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128
Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128
Antwort von 192.168.1.100: Bytes=32 Zeit<10ms TTL=128

Ping-Statistik für 192.168.1.100:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
  
```

6.2 Benutzerverwaltung

Der Zugriff auf das Web-based Management und über FTP ist durch Passwörter geschützt. Im Auslieferungszustand sind folgende Benutzer und Zugriffsrechte angelegt:

| Benutzername | Passwort | WBM | FTP | WAGOframe |
|--------------|----------|-------------|-------------|-------------|
| admin | wago | Vollzugriff | Vollzugriff | Vollzugriff |
| user | user | Teilzugriff | Vollzugriff | Teilzugriff |
| guest | guest | Teilzugriff | Vollzugriff | Leserecht |

Die Passwörter lassen sich mittels WBM und WAGOframe ändern.

6.3 Das Dateisystem

Für den Anwender stehen zwei Partitionen des Dateisystems zur Verfügung. Dieses besteht aus einer RAM-Disk und einer Partition im Flash-Speicher:

Tabelle 86: Überblick der Partitionen des Dateisystems

| Partition | Format | Typ | Größe | Verwendung |
|-----------|--------|------------|-------|------------|
| R:\ | FAT 12 | RAM-Disk | 1MB | - |
| U:\ | FAT 12 | Flash-Disk | 1MB | WBM-Seiten |

Die Partition R:\ ist als nicht remanente RAM-Disk ausgelegt und kann zum Zwischenspeichern von Daten verwendet werden. Diese Dateien gehen bei einem Neustart verloren. Um Dateien dauerhaft zu speichern, verwenden Sie die Partitionen „U:\“

Auf die oben aufgeführten Partitionen können Sie mittels FTP zugreifen.

6.3.1 Zugriff per FTP

Die Partition U:\ ist das Startverzeichnis für eingehende FTP-Verbindungen.

```

C:\WINNT\system32\cmd.exe - ftp 192.168.1.100
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

U:\>ftp 192.168.1.100
Verbindung mit 192.168.1.100 wurde hergestellt.
220 Nucleus FTP Server (Version 1.7) ready.
Benutzer (192.168.1.100:(none)): admin
331 User name okay, need password.
Kennwort:
230 User logged in, proceed.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
drw-rw-rw-  1 owner  group           0 Apr 10 12:55 images
drw-rw-rw-  1 owner  group           0 Apr 10 12:55 private
-rw-rw-rw-  1 owner  group       2977 Apr 10 12:55 state.ssi
-rw-rw-rw-  1 owner  group       6959 Apr 10 12:55 styles.css
-rw-rw-rw-  1 owner  group        679 Apr 10 12:55 index.htm
226 Closing data connection. Requested file action successful.
FTP: 346 Bytes empfangen in 0,02Sekunden 21,63KB/s
ftp> cd private
250 Requested file action okay, completed.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
drw-rw-rw-  1 owner  group           0 Apr 10 12:55 .
drw-rw-rw-  1 owner  group           0 Apr 10 12:55 ..
-rw-rw-rw-  1 owner  group       2560 Apr 10 12:55 trace.ssi
-rw-rw-rw-  1 owner  group       2287 Apr 10 12:55 administration.ssi
-rw-rw-rw-  1 owner  group       2011 Apr 10 12:55 users.ssi
-rw-rw-rw-  1 owner  group       3736 Apr 10 12:55 watchdog.ssi
-rw-rw-rw-  1 owner  group       8356 Apr 10 12:55 ethernet.ssi
-rw-rw-rw-  1 owner  group       4038 Apr 10 12:55 tcpip.ssi
-rw-rw-rw-  1 owner  group       3203 Apr 10 12:55 port.ssi
-rw-rw-rw-  1 owner  group       2287 Apr 10 12:55 clock.ssi
226 Closing data connection. Requested file action successful.
FTP: 699 Bytes empfangen in 0,01Sekunden 46,60KB/s
ftp> quit_

```

Das Beispiel zeigt den Inhalt des Ordners „U:\private\“ an. Diese Dateien sind der Quellcode des Web-based Management.

6.4 Das Web-based Management (WBM)

Fast alle Einstellungen lassen sich auch über das Web-based Management einsehen und verändern. Starten Sie dazu ihren Browser und geben Sie in die Adresszeile „http://“ gefolgt von der IP-Adresse des 767-Knoten ein (Bsp.: http://192.168.1.100).

WAGO
INNOVATIVE CONNECTIONS

Web Based Management

Navigation

- Information
- TCP/IP
- Port
- Watchdog
- Clock
- Ethernet
- Users
- Administration

Status Information

Coupler Details

| | |
|-------------------|-------------------|
| Order Number | 0767-2301 |
| Mac Address | 00:30:DE:00:00:00 |
| Firmware Revision | 01.01.04 (00) |

Network Details

| | |
|-------------|--------------|
| IP address | 192.168.1.3 |
| Subnet mask | 255.255.0.0 |
| Gateway | 0.0.0.0 |
| Hostname | 0030DE000000 |
| Domainname | wago.com |

Module Status

| | |
|----------------------------|---------------------|
| State Modbus Watchdog TCP: | Disabled |
| State Modbus Watchdog UDP: | Disabled |
| Error Group | 0 |
| Error Code | 0 |
| Error Argument | 0 |
| Error Description | Coupler running, OK |

Alle verlinkten Seiten des WBM sind Passwortgeschützt.

Netzwerkknwort eingeben

Geben Sie Benutzernamen und Kennwort ein.

Site: 192.168.1.100

Bereich: /INDEX.HTM

Benutzername:

Kennwort:

Dieses Kennwort in der Kennwortliste speichern

OK Abbrechen

| Benutzer | Passwort |
|----------|----------|
| admin | wago |
| user | user |
| guest | guest |

Weitere Informationen finden Sie auf der jeweiligen WBM-Seite oder im Handbuch zum 767-1301.

7 WAGOframe

Der WAGOframe ist eine FDT/DTM-Rahmenapplikation für die Konfiguration, Diagnose und Aktualisierung von FDT-konformen Feldgeräten.

FDT/DTM ist ein herstellerübergreifendes Konzept für die Parametrierung von Feldgeräten unterschiedlicher Hersteller mit nur einem Programm. Der Begriff „Field Device Tool“ (FDT) steht nicht für ein konkretes Programm, sondern definiert die Schnittstellen, die ein Programm erfüllen muss, um mit DTMs verschiedener Hersteller kooperieren zu können.

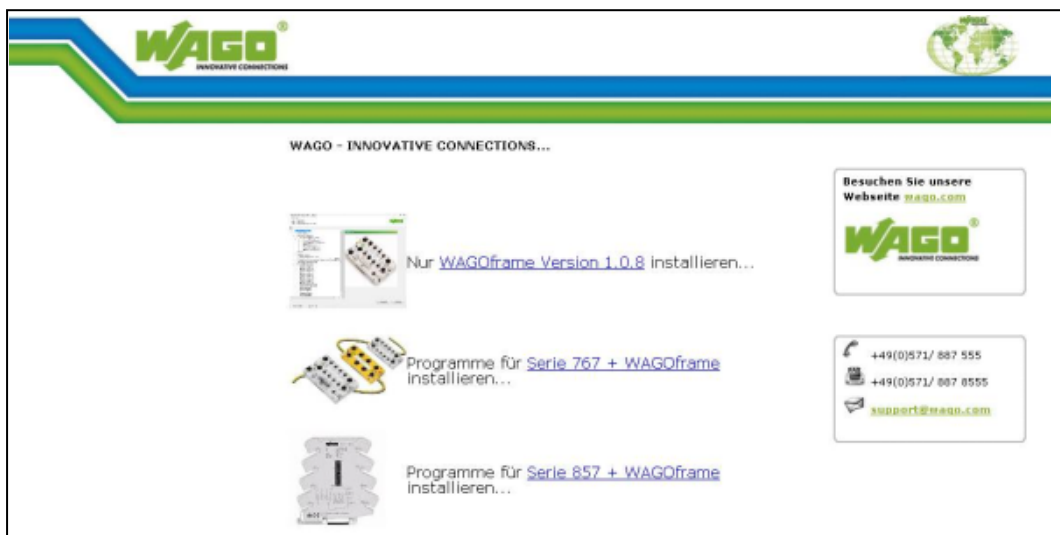
Ein „Device Type Manager“ (DTM) kapselt alle Einstellmöglichkeiten eines Feldgerätes (inklusive grafischer Oberfläche) in einem Programm, das in einer FDT/DTM-Rahmenapplikation ausgeführt wird. Die Spezifikation unterscheidet zwischen Geräte-DTM, Kommunikations-DTM und Gateway-DTM.

7.1 Installation

Ein browserbasierter Assistent führt durch die Installation der benötigten Treiber, DTM und Programme. Bei der Installation von der CD öffnet sich selbstständig der verwendete Internet-Browser mit folgender Startseite:

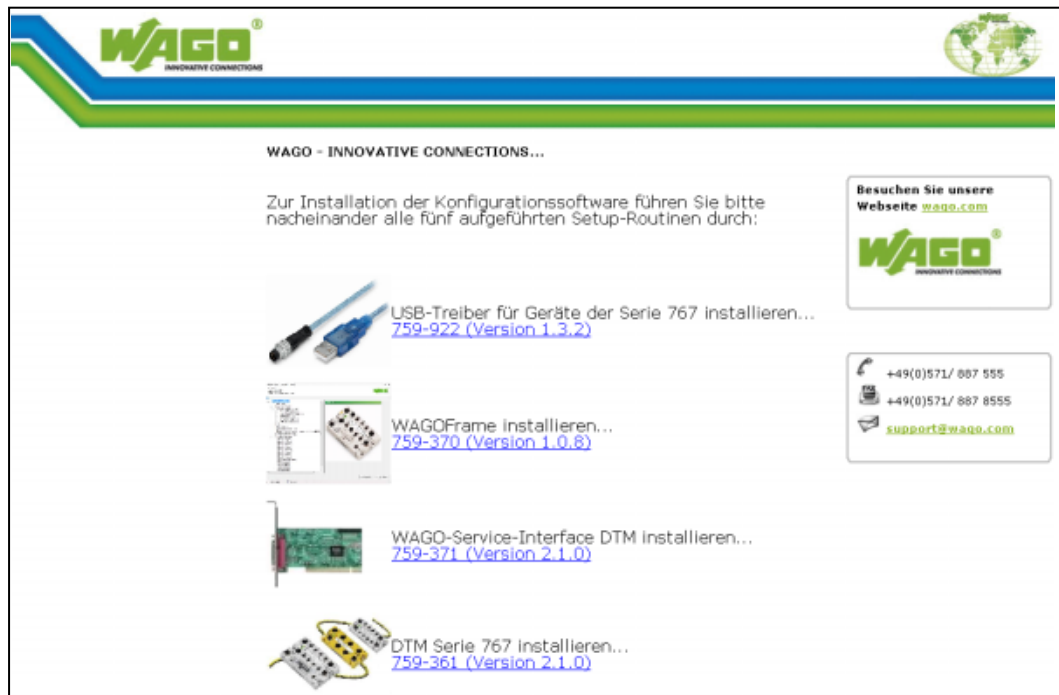


1. Bei der Installation aus dem Dateisystem wechseln Sie in das Verzeichnis „~/WAGOframe CD-ROM_v3.0.0“ und öffnen Sie die Datei „Deutsch_main.htm“.



2. Um alle benötigten Komponenten zu installieren, wählen Sie „Serie 767“.

3. Es erscheint folgende Seite. Zur Installation der Konfigurationssoftware führen Sie bitte nacheinander alle fünf Installationsprogramme aus.



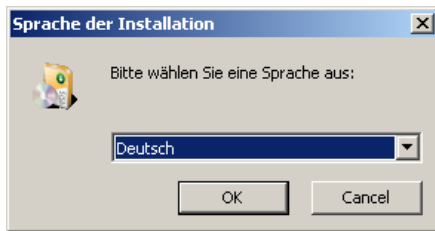
Ist bereits eine ältere Version des WAGOframe installiert, so ist diese vor der Installation zu deinstallieren. Gleiches gilt für die DTM.

Hinweis



Die Installation des USB-Treibers 759-922 kann entfallen, wenn dieser bereits über das „CoDeSys 3“-Setup (für den Feldbuskoppler nicht notwendig) auf ihrem Rechner installiert wurde. Ein vorhandener USB-Treiber wird ggf. aktualisiert.

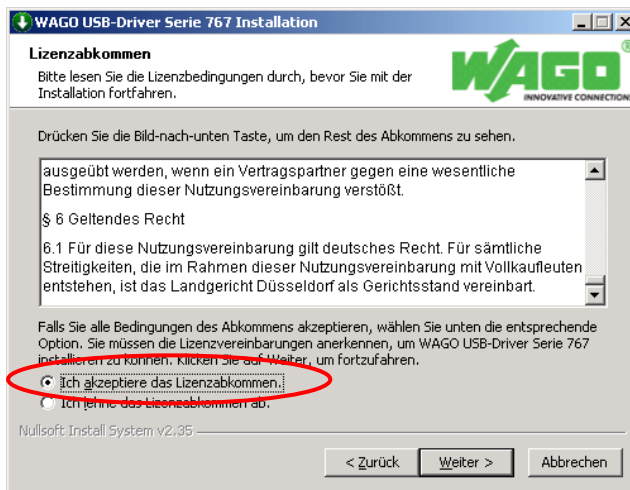
Der Ablauf der Installation der einzelnen Komponenten (hier der USB-Treiber 759-922) ist stets identisch und beginnt mit der Sprachauswahl.



1. Weiter mit der Schaltfläche [OK].

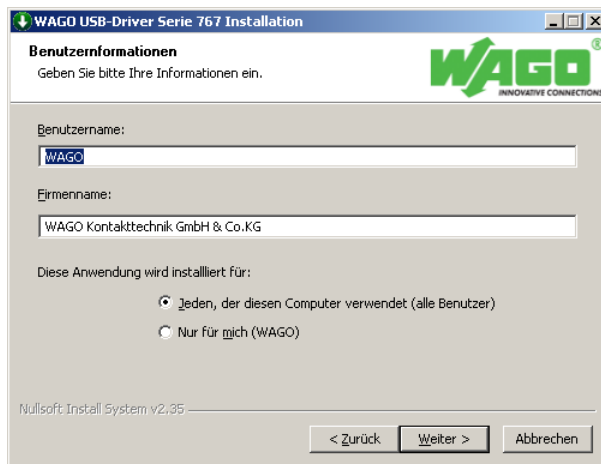


2. Weiter mit der Schaltfläche [Weiter >].

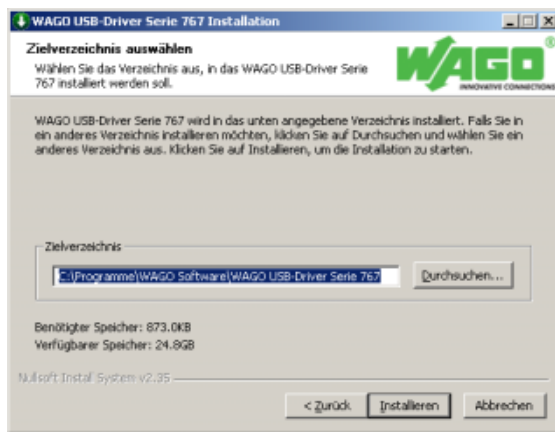


3. Lesen Sie sich die Nutzungsbedingungen für den WAGO-USB-Treiber durch, um die Auswahl „Ich akzeptiere.“ zu aktivieren. Bestätigen Sie diese Auswahl. Klicken Sie anschließend auf die Schaltfläche [Weiter].

4. Überprüfen Sie die persönlichen Einstellungen.

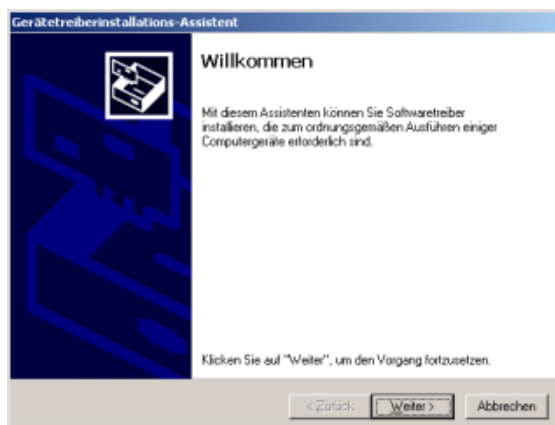


5. Weiter mit der Schaltfläche [Weiter >].

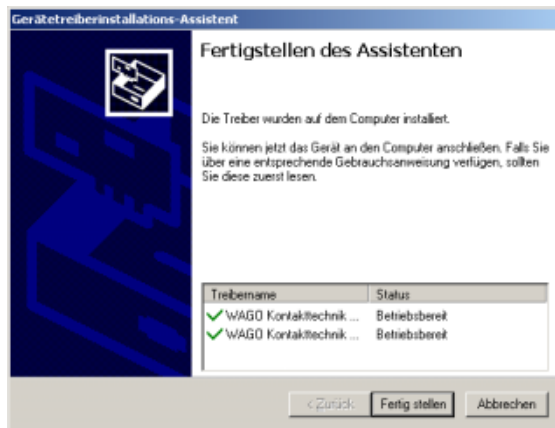


6. Wählen Sie den Installationspfad, unter dem die Anwendung gespeichert werden soll.

7. Starten Sie die Installation durch Betätigen der Schaltfläche [Installieren].



8. Weiter mit der Schaltfläche [Weiter >].



9. Abschluss der Installation durch Betätigen der Schaltfläche **[Fertig stellen]**.



10. Beenden Sie den Assistenten durch Betätigen der Schaltfläche **[Fertig stellen]**.

Wiederholen Sie diese Schritte für die restlichen Softwarekomponenten:

- WAGOframe: Bestellnummer 759-370 (Version 3.0.0)
- DTM WAGO-Service-Interface: Bestellnummer 759-371 (Version 2.1.0)
- DTM für die Feldbuskoppler und I/O-Module:
Bestellnummer 759-361 (Version 2.1.0)
- DTM für das System-Update: Bestellnummer 759-362 (Version 1.0.0)

7.2 Inbetriebnahme des Feldbuskopplers

Schalten Sie die Versorgungsspannung des Feldbuskopplers ein und verbinden Sie seine Service-Schnittstelle über das USB-Kommunikationskabel 756-4101/0042-0030 mit einem freien USB-Steckplatz an ihrem Computer.

Die Kommunikation zwischen dem WAGOframe und dem Feldbuskoppler erfolgt über den Kommunikations-DTM „WAGO Service Interface“. Der verwendete COM-Port ist abhängig vom USB-Steckplatz und ist im „WAGO Service Interface“ entsprechend zu parametrieren.

Hinweis

Wird zu einem späteren Zeitpunkt der Feldbuskoppler an einem anderen USB-Steckplatz angeschlossen, ändert sich dadurch auch der von „COM_Service“ verwendete COM-Port. In diesem Fall muss die Einstellung für die serielle Schnittstelle im Kommunikations-DTM angepasst werden.

7.3 Bedienung

Starten Sie das Programm WAGOframe über das dazugehörige Piktogramm auf dem Desktop oder über **Start > Programme > WAGO Software > WAGOframe > WAGOframe**.



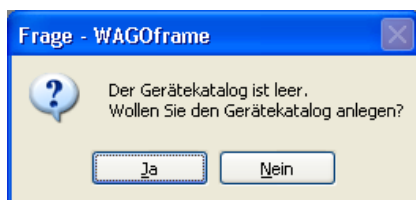
Nach dem Start können Sie zwischen „Punkt zu Punkt Modus“ und „Expertenmodus“ wählen. Der „Punkt zu Punkt Modus“ ist speziell für Konfiguration einfacher Geräte, die keinen Sub-Bus (S-BUS) besitzen, wie z. B. WAGO Jumpflex. Zur Parametrierung von 767-Komponenten verwenden Sie den „Expertenmodus“.



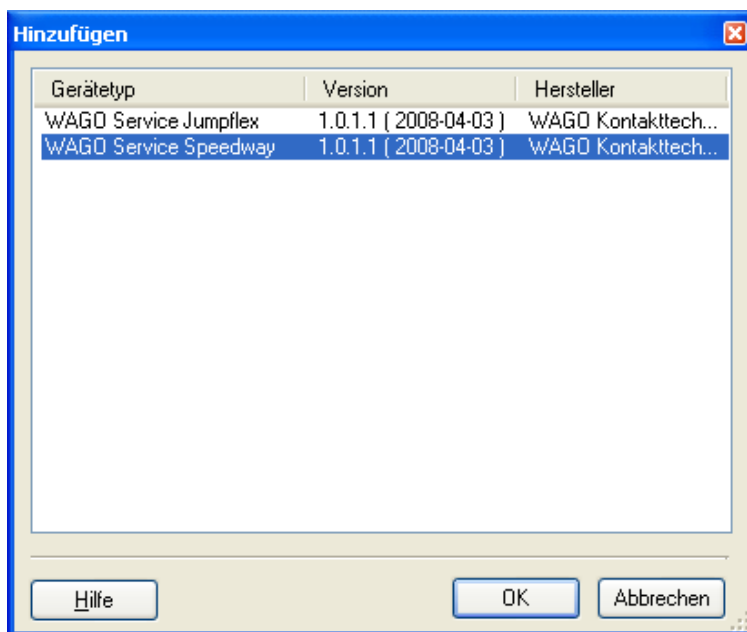
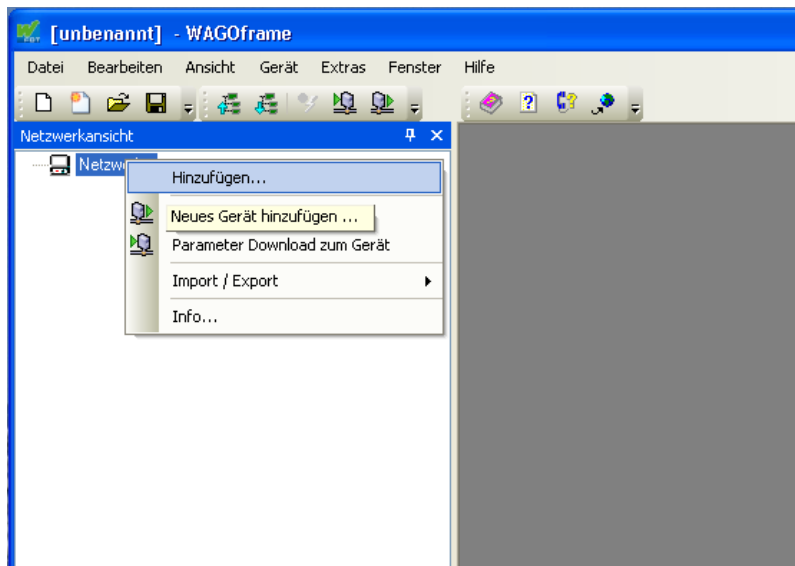
1. Wechseln Sie in den „Expertenmodus“ und betätigen Sie die Schaltfläche **[Weiter >]**.

Beim ersten Aufruf ist noch kein Gerätecatalog angelegt.

2. Betätigen Sie die Schaltfläche **[Ja]** im nachfolgenden Dialog, um den Gerätecatalog auf ihrem PC einzurichten

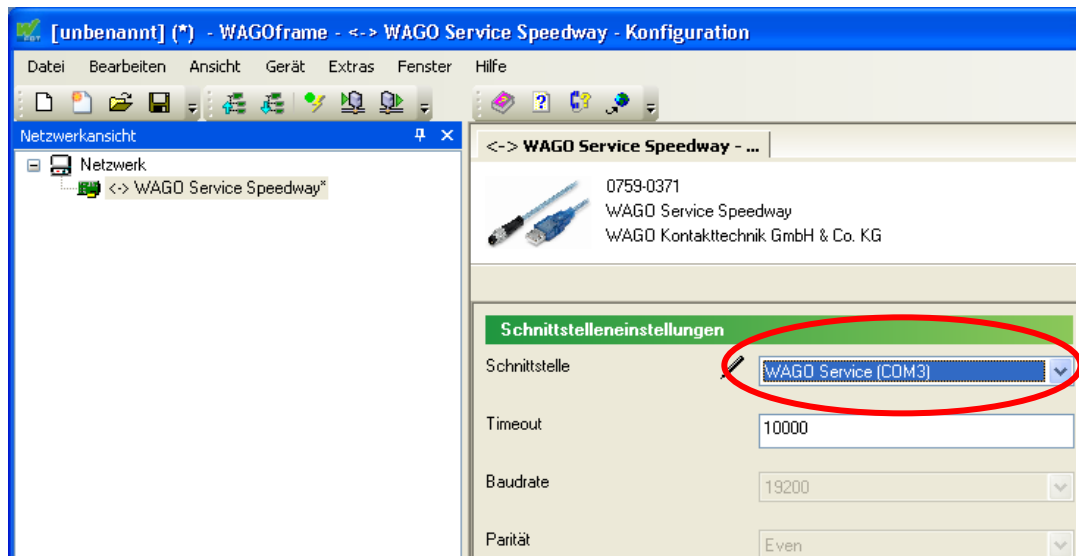


- Markieren Sie im Fenster „Netzwerkansicht“ das Element „Netzwerk“ und wählen Sie aus dem Kontextmenü (rechte Maustaste) **Hinzufügen...** Es öffnet sich ein Dialog, der alle verfügbaren Kommunikationstreiber anzeigt.



- Wählen Sie den Kommunikationstreiber „WAGO Service Speedway“ aus und betätigen Sie die Schaltfläche **[OK]**.
- Führen Sie im Fenster „Netzwerkansicht“ einen Doppelklick auf das neu eingefügte Element „<->WAGO Service Speedway“ aus.

Einstellen der seriellen Schnittstellenparameter des Kommunikations-DTM
„WAGO Service Speedway“.

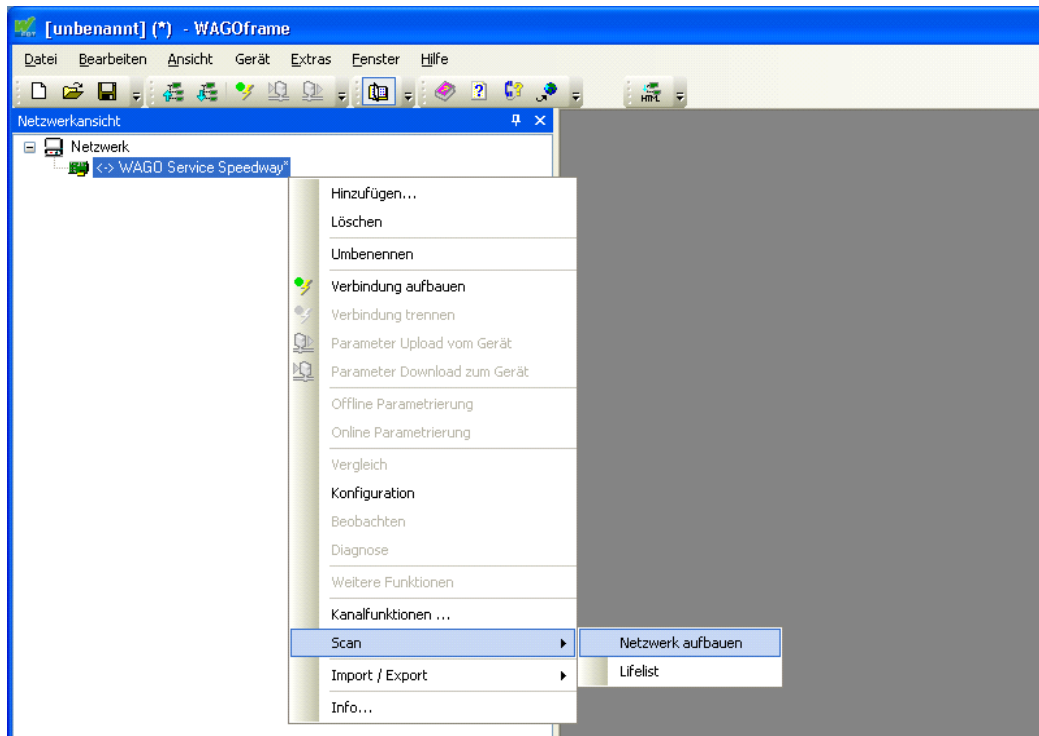


1. Wählen Sie die verwendete Schnittstelle aus der Liste und übernehmen Sie die Einstellung.
Ist die Liste der verfügbaren Schnittstellen leer, überprüfen Sie, ob der Feldbuskoppler eingeschaltet und über das USB-Kommunikationskabel mit Ihrem PC verbunden ist.
2. Um die Netzwerkkonfiguration zu vervollständigen, können Sie nun die einzelnen DTMs aus dem Katalog auswählen oder einen „Netzwerk-Scan“ durchführen.

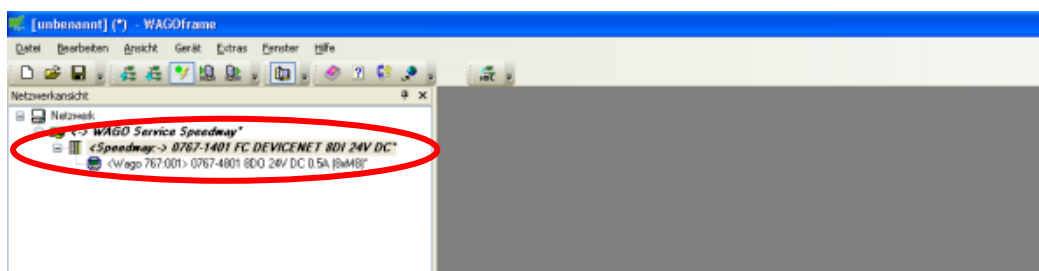
Beachten Sie beim manuellen Hinzufügen, dass jede FDT-Rahmenanwendung zwischen Online- und Offline-Modus unterscheidet. Jeder Modus stellt einen separaten Funktionsumfang bereit.

Im Online-Modus besteht eine direkte Kopplung zwischen Anzeige und angeschlossenen 767-Komponenten.
Der Offline-Modus ermöglicht die Parametrierung eines noch nicht vorhandenen Gerätes. Dieser wird weiterhin auch zur Reduzierung des Datentransfers zwischen dem WAGOframe und dem Gerät verwendet.
Befindet sich ein Gerät im Online-Modus, wird dessen Name im Netzwerkfenster ***kursiv-fett*** dargestellt.

- Um einen Netzwerk-Scan auszuführen, öffnen Sie im Fenster „Netzwerk“ das Kontextmenü (rechte Maustaste) des Elements „WAGO Service Speedway“ und wählen im Menüpunkt Scan den Eintrag **Netzwerk aufbauen**.

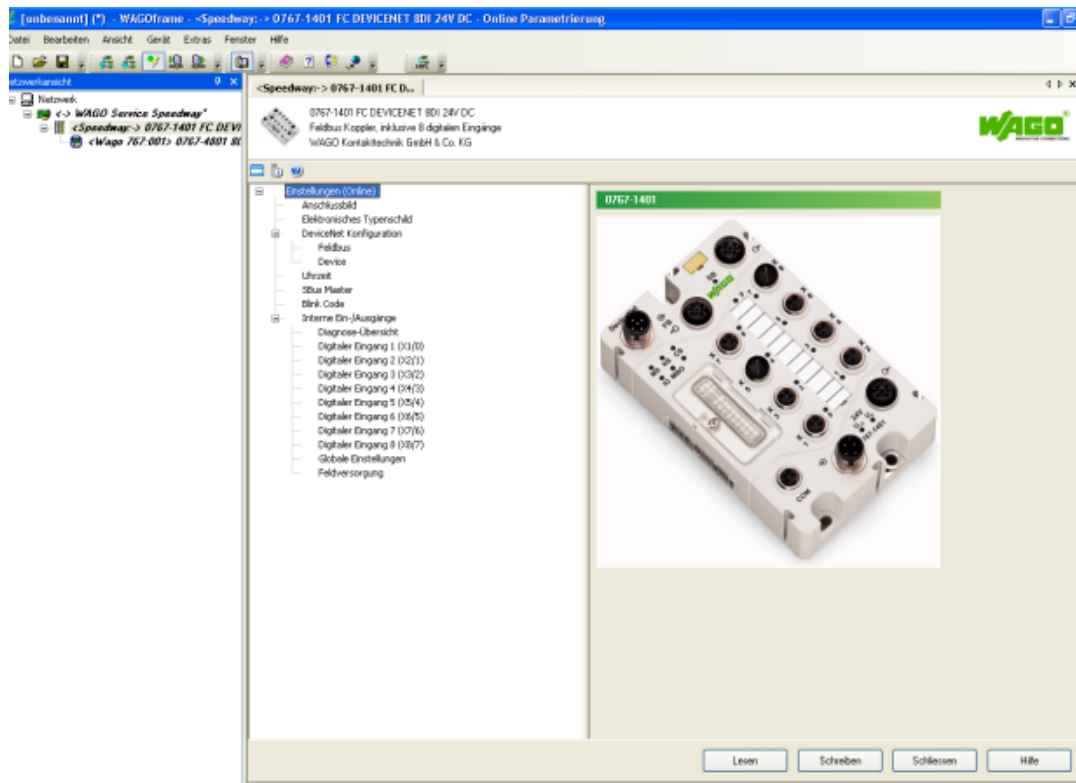


- Nach Ausführung des Netzwerk-Scans sollte der WAGOframe den Feldbuskoppler sowie die I/O-Module gefunden haben.
- Werden während der Ausführung des Netzwerk-Scans keine passende DTM für den Feldbuskoppler gefunden, wird eine sogenannte „Live-List“ erstellt. Klicken Sie in dieser auf **[Alle Hinzufügen und fortfahren]**.



- Beachten Sie, dass sich der Feldbuskoppler im Online-Modus befindet, das Modul jedoch im Offline-Modus. Markieren Sie das Modul und wählen Sie aus dem Kontextmenü den Eintrag **[Verbindung aufbauen]**, um in den Online-Modus zu wechseln.

7. Öffnen Sie die Detailansicht mit einem Doppelklick auf den Feldbuskoppler. Nach dem Anpassen der Fensterbreite und Schließen des Gerätekatalogs sollte sich der WAGOframe wie folgt darstellen:



Die aktuelle Betriebsart wird in Klammern hinter dem ersten Knoten der Baumansicht angezeigt (siehe linkes Fenster). Beachten Sie, dass ein Wechsel zwischen den Betriebsarten nur möglich ist, wenn das betreffende Detailfenster geschlossen ist.

Erläuterungen der Schaltflächen:

- **[Lesen]**
Auslesen und Anzeigen der im Feldbuskoppler befindlichen Parameter.
- **[Schreiben]**
Speichert die geänderten Werte im Feldbuskoppler.
- **[Schließen]**
Beendet die Anwendung WAGOframe.
- **[Hilfe]**
Öffnet für einen zuvor selektierten Eintrag (z. B. digitaler Eingang, Blink Code) die Online-Hilfe.

Information Weitere Informationen finden Sie im Handbüchern des WAGOframe und des Feldbuskopplers.



8 System-Update

Mit dem System-Update aktualisieren Sie die Firmware der 767-Komponenten. Damit der Feldbusknoten nach der Firmware-Aktualisierung konsistent und lauffähig bleibt, wird das System-Update sowohl für den Feldbuskoppler als auch für die angeschlossenen I/O-Module durchgeführt.

ACHTUNG System-Update!

Vor der Durchführung des System-Updates ist Folgendes sicherzustellen, um einen möglichen Schaden am 767-System zu vermeiden:

- Die Spannungsversorgung darf während des System-Updates nicht unterbrochen werden.
 - Um eine Beeinflussung durch den Feldbus auszuschließen, muss der Feldbusstecker vor dem System-Update vom Feldbuskoppler getrennt werden.
-

Voraussetzung:

- Sie haben den WAGOframe (759-370) installiert.
- Sie haben den WAGO-Service-Interface DTM (759-371) installiert.
- Sie haben den USB-Treiber für Geräte der Serie 767 installiert (759-922).
- Sie haben den System-Update-DTM (759-362) installiert.
- Es stehen für die angeschlossenen 767-Komponenten „Update-Pakete“ zur Verfügung.

Ablauf des System-Updates

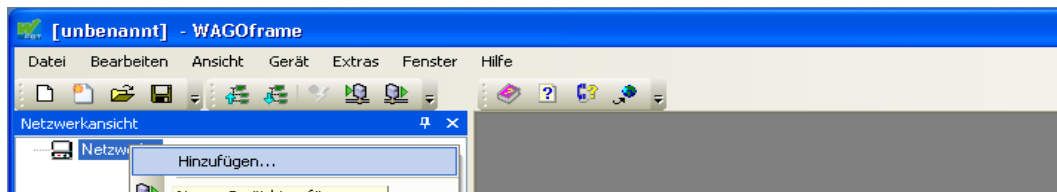
Folgende Schritte werden beim System-Update für jede 767-Komponente durchgeführt:

1. Parametrierung aus den 767-Komponenten auslesen und auf dem PC speichern.
2. Aktualisierung der Firmware der 767-Komponenten.
3. Zurückschreiben der Parametrierung vom PC in die 767-Komponenten.
4. Parametrierung gültig setzen und fertigstellen.

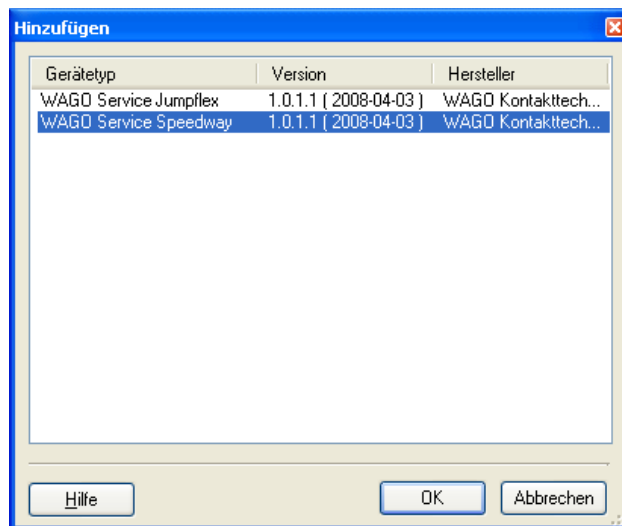
8.1 Hinzufügen des System-Update-DTM

Zum Hinzufügen des System-Update-DTM in den WAGOframe gehen Sie folgendermaßen vor (Schritte 1 – 4 entfallen, wenn Sie das „WAGO Service Speedway“ schon installiert haben):

1. Klicken Sie im Fenster „Netzwerkansicht“ mit der rechten Maustaste auf „Netzwerk“.

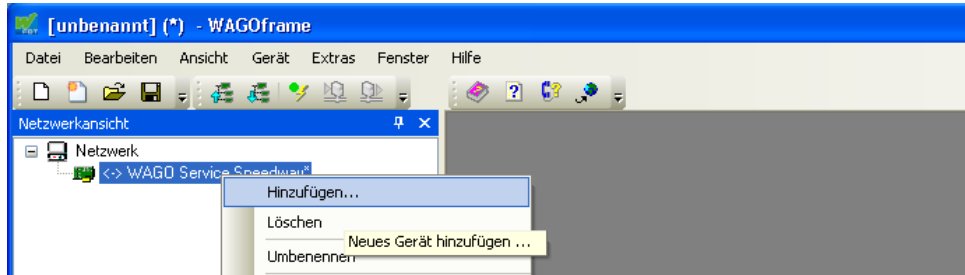


2. Wählen Sie im Kontextmenü den Eintrag **Hinzufügen ...**. Es öffnet sich der Dialog „Hinzufügen“.

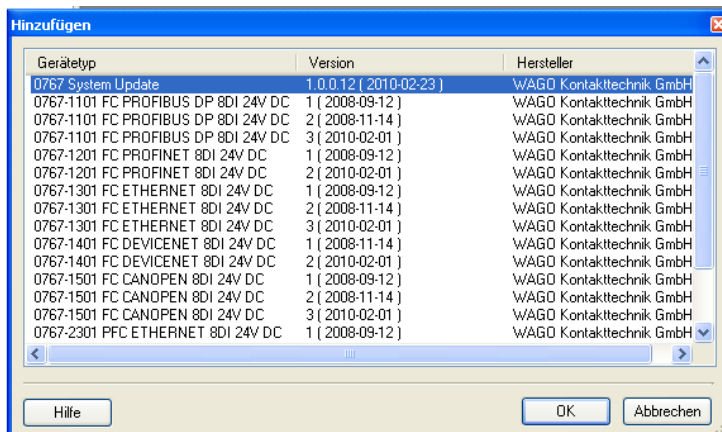


3. Selektieren Sie im Dialog „Hinzufügen“ den DTM **WAGO Service Speedway**.
4. Klicken Sie auf **[OK]**, um Ihre Auswahl zu übernehmen.

5. Klicken Sie im Fenster „Netzwerkansicht“ mit der rechten Maustaste auf den Gerätetreiber „WAGO Service Speedway“.
6. Wählen Sie im Kontextmenü den Eintrag **Hinzufügen ...**. Es öffnet sich der Dialog „Hinzufügen“.



7. Selektieren Sie im Dialog „Hinzufügen“ den DTM 0767 System-Update.

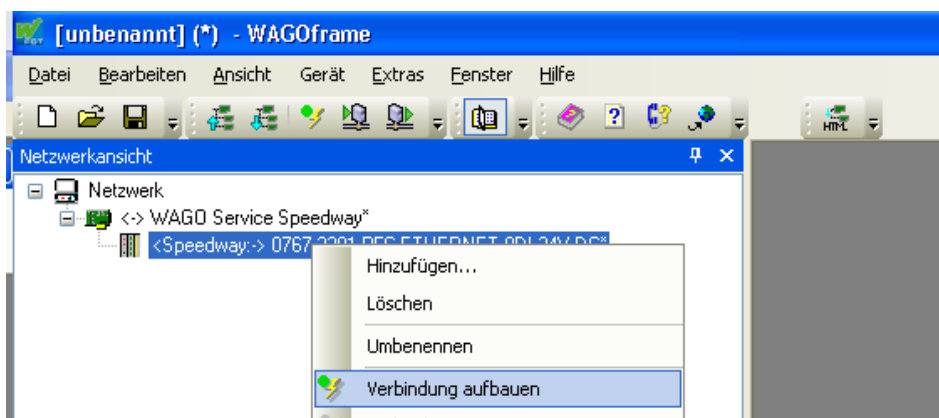


8. Klicken Sie auf [OK], um Ihre Auswahl zu übernehmen.

8.2 Verbindung zum 767-Knoten mittels Update-DTM aufbauen

Die Firmware kann nur aktualisiert werden, wenn eine Kommunikationsverbindung zwischen Update-DTM und 767-Knoten besteht. Gehen Sie dazu folgendermaßen vor:

1. Klicken Sie im Fenster „Netzwerkansicht“ mit der rechten Maustaste auf den Gerätetreiber „<Speedway:> 0767 System-Update“.
2. Wählen Sie im Kontextmenü den Eintrag **Verbindung aufbauen**. Wenn die Fortschrittsanzeige 100 % erreicht hat und anschließend der Eintrag in der Netzwerkansicht *kursiv-fett* dargestellt wird, ist die Kommunikationsverbindung hergestellt.



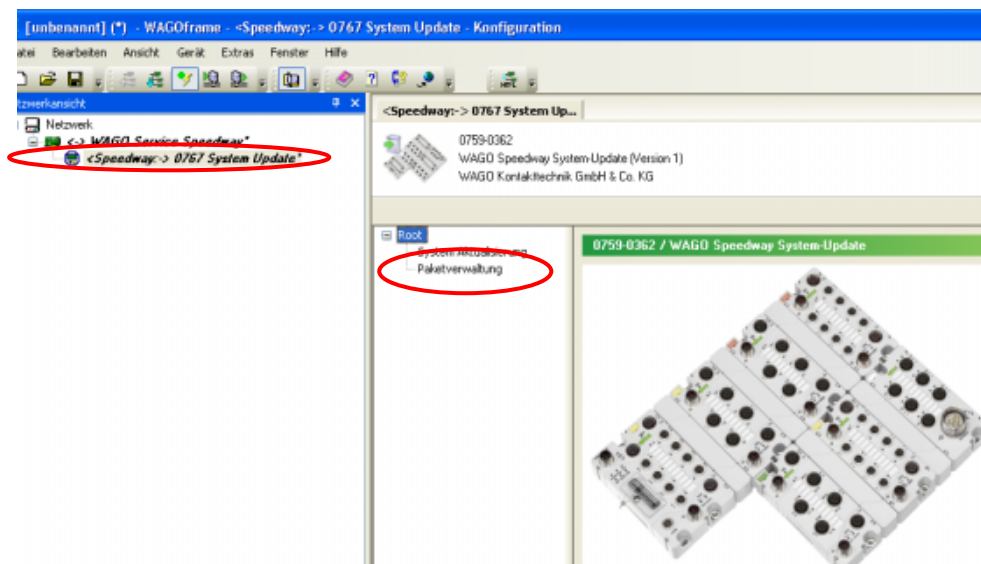
8.3 Aktualisieren der 767-Komponenten

Die aktuelle Firmware erhalten Sie über den WAGO-Support. Senden Sie dazu eine E-Mail mit dem Betreff „Aktuelle Speedway-Firmware“ sowie den Bestellnummern der entsprechenden 767-Komponenten an: support@wago.com.

Firmware-Pakete importieren

Damit Sie die zugesendeten Firmware-Pakete verwenden können, sind diese in den System-Update-DTM zu importieren. Gehen Sie dazu wie folgt vor:

1. Speichern Sie die zugesendeten Dateien mit der Endung „*.wup“ in einem beliebigen Verzeichnis auf Ihrem PC.
2. Öffnen Sie die Benutzer-Oberfläche des DTM per Doppelklick auf den Eintrag **0767 System Update** in der Netzwerkansicht.

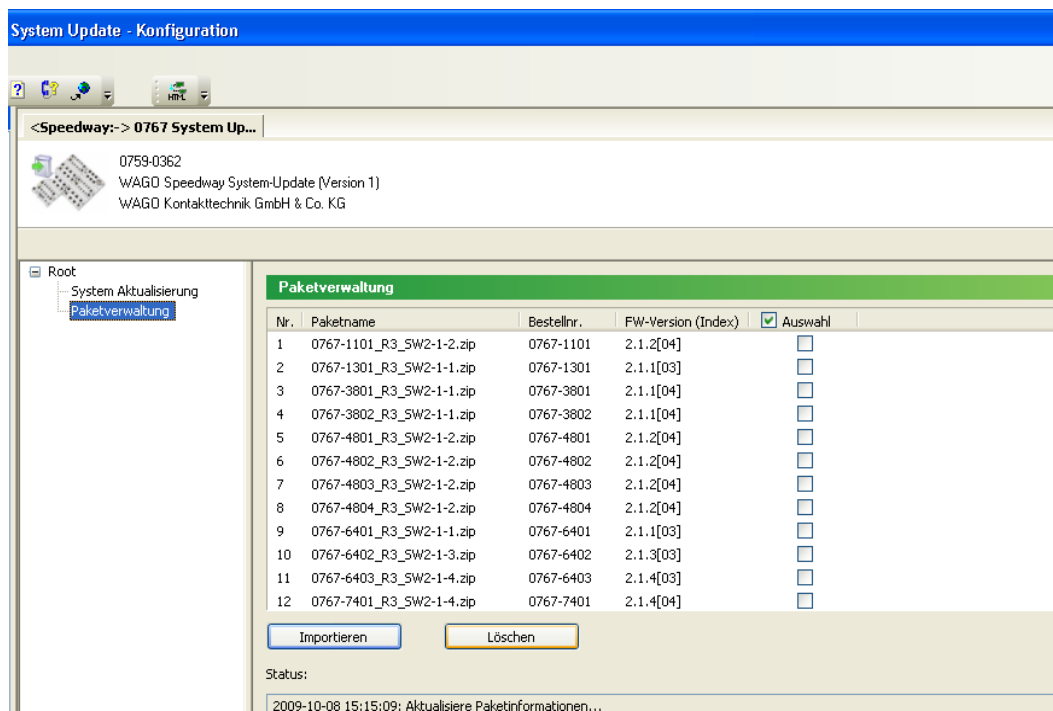


3. Klicken Sie im linken Fenster des WAGOframe auf „Paketverwaltung“.
4. Zum Importieren der zugesendeten Firmware-Dateien klicken Sie auf **[Importieren]**. Wählen Sie in dem sich öffnenden Fenster das Verzeichnis aus, in dem Sie die Firmware-Dateien abgelegt haben, und wählen Sie die zu verwendende Datei aus. Zum Übernehmen der Dateien klicken Sie auf **[Öffnen]**.

Firmware-Pakete löschen

Um die Oberfläche der „Paketverwaltung“ übersichtlich zu halten, können Sie nicht mehr benötigte Update-Pakete aus der Ansicht entfernen. Gehen Sie dazu wie folgt vor:

1. Klicken Sie im rechten Fenster auf die Checkbox für die nicht benötigten Firmware-Dateien (Haken setzen).
2. Entfernen Sie die selektierten Firmware-Pakete mittels der Schaltfläche **[Löschen]**.



Systemaktualisierung

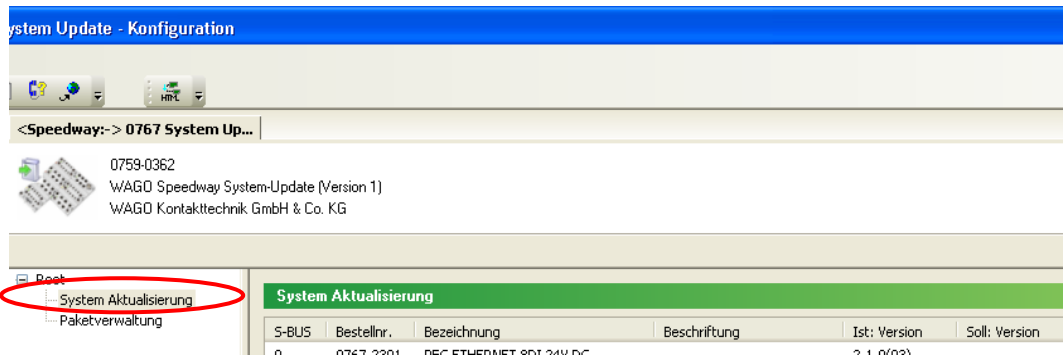
Hinweis



Bei einer Aktualisierung der Firmware des Feldbuskopplers können die gespeicherten Modulparameter überschrieben werden. Überprüfen Sie deshalb nach einer Aktualisierung der Firmware Ihre bestehende Parametrierung.

Hier führen Sie die Systemaktualisierung durch. Die von Ihnen vorgenommenen Moduleinstellungen bleiben dabei im Normalfall erhalten. Andernfalls erscheint eine entsprechende Warnmeldung. Wenn Sie dennoch eine Aktualisierung der Firmware durchführen, dann werden die 767-Komponenten in den Auslieferungszustand zurückversetzt.

1. Klicken Sie im linken Fenster auf „Systemaktualisierung“.

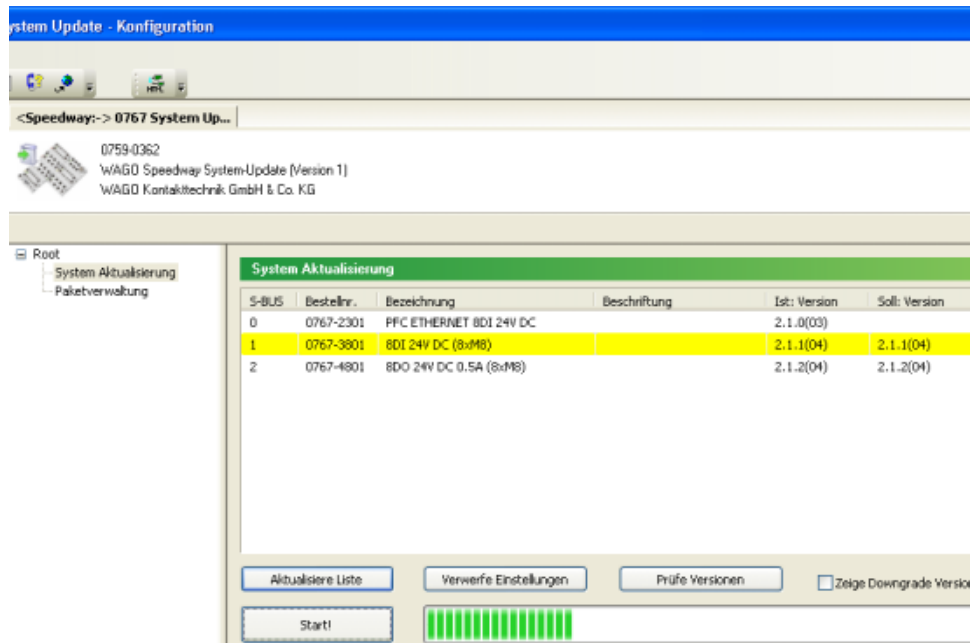


2. Im rechten Fenster ist der Feldbuskoppler mit allen angeschlossenen I/O-Modulen aufgelistet. Zunächst sind alle 767-Komponenten, für die eine Aktualisierung möglich ist, vorgewählt. Ist diese Vorauswahl nicht korrekt oder sollen bestimmte 767-Komponenten nicht aktualisiert werden, so wählen Sie diese ab.
 „Ist-Version“: Aktuell im Gerät vorhandene Firmware
 „Soll-Version“: Version der Firmware, die in die 767-Komponenten geladen werden soll. Stehen mehrere Soll-Versionen zur Auswahl, wählen Sie die für Sie relevante aus.

3. Zur Systemaktualisierung klicken Sie auf **[Start!]**. Während der Aktualisierung der ausgewählten 767-Komponenten sind diese gelb markiert.

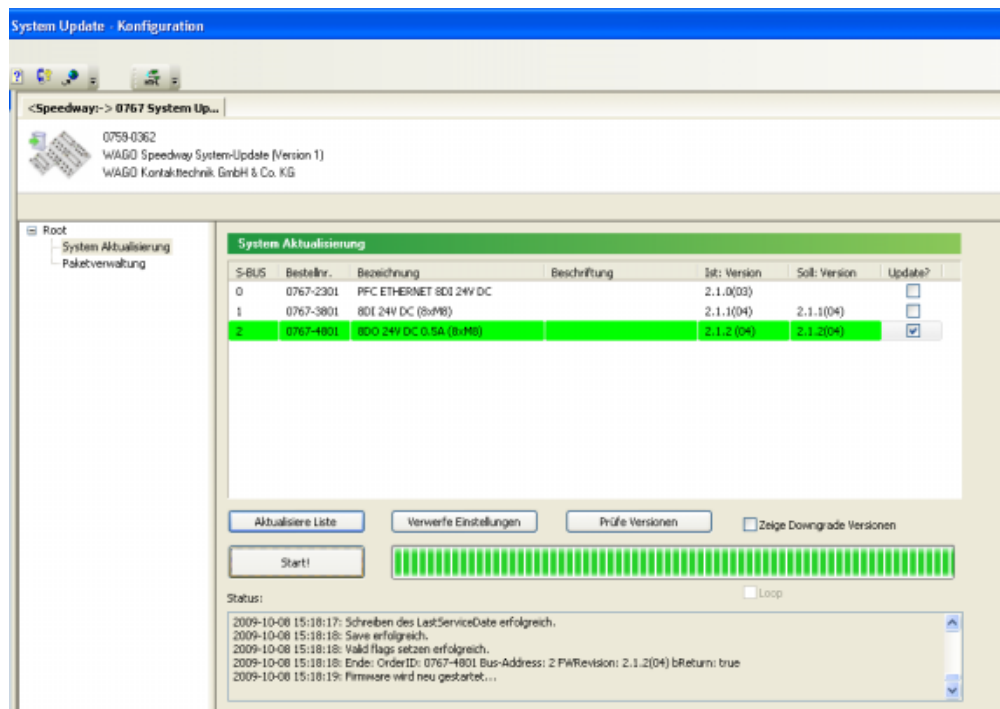
Hinweis

Während der Firmware-Aktualisierung trennt der Feldbuskoppler alle seine COM-Ports. Auf dem PC unter Windows 2000 wird das Ereignis erkannt und eine Windows-Meldung erscheint. Hierbei handelt es sich nicht um einen Fehler. Bestätigen Sie die Meldung mit **[OK]**.



| Schaltfläche | Beschreibung |
|---------------------------------|---|
| [Aktualisiere Liste] | Mit Hilfe dieser Funktion wird der Knotenaufbau neu ausgelesen und die Ansicht aktualisiert. |
| [Verwerfe Einstellungen] | Löschen der von Ihnen vorgenommenen Selektionen und Einstellungen. |
| [Prüfe Versionen] | Haben Sie Ihre Auswahl vorgenommen, führen Sie über diese Schaltfläche eine Plausibilitätsprüfung durch. Es wird geprüft, ob die von Ihnen gewählte Konstellation möglich ist (dies wird auch beim Start des System-Updates automatisch ausgeführt) |
| Zeige Downgrade-Versionen | Ist dieses Kontrollfeld aktiviert, werden in der Liste der Soll-Versionen auch die Versionen zum Downgrade eines Gerätes angezeigt. |
| [Start!]/[Abbrechen] | Starten/Abbrechen der Systemaktualisierung. |

4. Ist die Systemaktualisierung abgeschlossen, werden die aktualisierten 767-Komponenten grün gekennzeichnet (siehe Abb.).



Während der Systemaktualisierung werden alle dafür benötigten Informationen auf dem PC abgelegt. Sollte die Systemaktualisierung fehlschlagen (Komponenten werden rot angezeigt), kann diese anschließend wiederholt werden. Die ursprüngliche Parametrierung bleibt dabei erhalten.

Sollte das System-Update erneut fehlschlagen, wenden Sie sich bitte an den WAGO-Support.

Information Weitere Informationen finden Sie in den Handbüchern des WAGOframe und des 767-Feldbuskopplers.



WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 D-32385 Minden
Hansastraße 27 D-32423 Minden
Telefon: 05 71/8 87 – 0
Telefax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: <http://www.wago.com>

