

Your Global Automation Partner

**TURCK**

# TBEN-L5-4RFID-8DXP-WIN

## Kompaktes RFID- und I/O-Modul

Betriebsanleitung



# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Über diese Anleitung</b>                       | <b>7</b>  |
| 1.1      | Zielgruppen                                       | 7         |
| 1.2      | Symbolerläuterung                                 | 7         |
| 1.3      | Weitere Unterlagen                                | 7         |
| 1.4      | Feedback zu dieser Anleitung                      | 8         |
| <b>2</b> | <b>Hinweise zum Produkt</b>                       | <b>9</b>  |
| 2.1      | Produktidentifizierung                            | 9         |
| 2.2      | Lieferumfang                                      | 9         |
| 2.3      | Rechtliche Anforderungen                          | 9         |
| 2.4      | Hersteller und Service                            | 9         |
| <b>3</b> | <b>Zu Ihrer Sicherheit</b>                        | <b>10</b> |
| 3.1      | Bestimmungsgemäße Verwendung                      | 10        |
| 3.2      | Allgemeine Sicherheitshinweise                    | 10        |
| <b>4</b> | <b>Produktbeschreibung</b>                        | <b>11</b> |
| 4.1      | Geräteübersicht                                   | 11        |
| 4.1.1    | Anzeigeelemente                                   | 11        |
| 4.2      | Eigenschaften und Merkmale                        | 11        |
| 4.3      | Funktionsprinzip                                  | 12        |
| 4.4      | Funktionen und Betriebsarten                      | 12        |
| 4.4.1    | Funktionen im Turck Service Tool                  | 12        |
| <b>5</b> | <b>Montieren</b>                                  | <b>13</b> |
| 5.1      | Gerät erden                                       | 14        |
| 5.1.1    | Erdungs- und Schirmungskonzept                    | 14        |
| 5.1.2    | Modul erden (FE)                                  | 15        |
| <b>6</b> | <b>Anschließen</b>                                | <b>16</b> |
| 6.1      | Module an das Ethernet anschließen                | 16        |
| 6.2      | Versorgungsspannung anschließen                   | 17        |
| 6.3      | RFID-Schreib-Lese-Köpfe anschließen               | 18        |
| 6.4      | Digitale Sensoren und Aktuatoren anschließen      | 19        |
| <b>7</b> | <b>In Betrieb nehmen</b>                          | <b>20</b> |
| 7.1      | IP-Adresse einstellen                             | 20        |
| 7.1.1    | IP-Adresse über Schalter am Gerät einstellen      | 20        |
| 7.1.2    | IP-Adresse über das Turck Service Tool einstellen | 22        |
| 7.2      | Systemstart                                       | 25        |
| 7.2.1    | Voraussetzungen                                   | 25        |
| 7.2.2    | Anwendungen erstellen                             | 25        |
| 7.3      | Treiber   | 26        |
| 7.3.1    | Ethernet  | 26        |
| 7.3.2    | NandFlash   | 26        |
| 7.3.3    | USB-Host  | 26        |
| 7.3.4    | USB OTG   | 26        |
| 7.3.5    | 4.2.5. UART                                       | 27        |
| 7.3.6    | GPIO  | 27        |

|           |   |           |
|-----------|---|-----------|
| 7.3.7     | SPI   | 33        |
| 7.3.8     | I2C   | 36        |
| 7.3.9     | RTC   | 37        |
| 7.3.10    | Anwendung einsetzen                         | 37        |
| 7.3.11    | Debugging der Anwendung                     | 38        |
| 7.3.12    | Verwendung eines Netzwerk-Sockets in C#     | 38        |
| 7.3.13    | TBOX-API-Bibliothek verwenden               | 40        |
| 7.3.14    | Vorgehensweise mit einer C#-Anwendung       | 42        |
| 7.4       | Spezifische Einstellungen/Implementierungen | 43        |
| 7.4.1     | Anwendung automatisch starten               | 43        |
| 7.4.2     | Abbildversion auslesen                      | 43        |
| 7.4.3     | Adressierungsmodus auslesen                 | 43        |
| <b>8</b>  | <b>Betreiben</b>                            | <b>44</b> |
| 8.1       | LED-Anzeigen                                | 44        |
| <b>9</b>  | <b>Störungen beseitigen</b>                 | <b>45</b> |
| <b>10</b> | <b>Instand halten</b>                       | <b>45</b> |
| 10.1      | Geräte-Update durchführen                   | 45        |
| 10.2      | Firmware-Update durchführen                 | 46        |
| <b>11</b> | <b>Reparieren</b>                           | <b>56</b> |
| 11.1      | Geräte zurücksenden                         | 56        |
| <b>12</b> | <b>Entsorgen</b>                            | <b>56</b> |
| <b>13</b> | <b>Technische Daten</b>                     | <b>57</b> |





# 1 Über diese Anleitung

Die Anleitung beschreibt den Aufbau, die Funktionen und den Einsatz des Produkts und hilft Ihnen, das Produkt bestimmungsgemäß zu betreiben. Lesen Sie die Anleitung vor dem Gebrauch des Produkts aufmerksam durch. So vermeiden Sie mögliche Personen-, Sach- und Geräteschäden. Bewahren Sie die Anleitung auf, solange das Produkt genutzt wird. Falls Sie das Produkt weitergeben, geben Sie auch diese Anleitung mit.

## 1.1 Zielgruppen

Die vorliegende Anleitung richtet sich an fachlich geschultes Personal und muss von jeder Person sorgfältig gelesen werden, die das Gerät montiert, in Betrieb nimmt, betreibt, instand hält, demontiert oder entsorgt.

Der Systemintegrator muss über Kenntnisse im Umsetzen von Applikationen unter Windows verfügen und in der Lage sein, die Inbetriebnahme des Geräts ohne zusätzlichen Support für die Implementierung auf dem Windows-Betriebssystem durchzuführen. Zudem sind Kenntnisse zur Integration von RFID-Schreib-Lese-Kopf-Protokollen erforderlich.

## 1.2 Symbolerläuterung

In dieser Anleitung werden folgende Symbole verwendet:



### GEFAHR

GEFAHR kennzeichnet eine unmittelbar gefährliche Situation mit hohem Risiko, die zu Tod oder schwerer Verletzung führt, wenn sie nicht vermieden wird.



### WARNUNG

WARNUNG kennzeichnet eine möglicherweise gefährliche Situation mit mittlerem Risiko, die zu Tod oder schwerer Verletzung führen kann, wenn sie nicht vermieden wird.



### ACHTUNG

ACHTUNG kennzeichnet eine Situation, die möglicherweise zu Sachschäden führt, wenn sie nicht vermieden wird.



### HINWEIS

Unter HINWEIS finden Sie Tipps, Empfehlungen und wichtige Informationen. Die Hinweise erleichtern die Arbeit, enthalten Infos zu speziellen Handlungsschritten und helfen, Mehrarbeit durch falsches Vorgehen zu vermeiden.



### HANDLUNGSAUFFORDERUNG

Dieses Zeichen kennzeichnet Handlungsschritte, die der Anwender auszuführen hat.



### HANDLUNGSERGEBNIS

Dieses Zeichen kennzeichnet relevante Resultate von Handlungen und Handlungsabfolgen.

## 1.3 Weitere Unterlagen

Ergänzend zu diesem Dokument finden Sie im Internet unter [www.turck.com](http://www.turck.com) folgende Unterlagen:

- Projektierungshandbuch
- Betriebsanleitungen der Schreib-Lese-Köpfe

### 1.4 Feedback zu dieser Anleitung

Wir sind bestrebt, diese Anleitung ständig so informativ und übersichtlich wie möglich zu gestalten. Haben Sie Anregungen für eine bessere Gestaltung oder fehlen Ihnen Angaben in der Anleitung, schicken Sie Ihre Vorschläge an [techdoc@turck.com](mailto:techdoc@turck.com).



## 2 Hinweise zum Produkt

### 2.1 Produktidentifizierung

Diese Anleitung gilt für die folgenden kompakten RFID-Interfaces:

- TBEN-L5-4RFID-8DXP-WIN

### 2.2 Lieferumfang

Im Lieferumfang sind enthalten:

- Kompaktes RFID-Interface
- Verschlusskappen für M12-Buchsen
- Kurzanleitung

### 2.3 Rechtliche Anforderungen

Das Gerät fällt unter folgende EU-Richtlinien:

- 2014/30/EU (Elektromagnetische Verträglichkeit)
- 2011/65/EU (RoHS 2)

### 2.4 Hersteller und Service

Turck unterstützt Sie bei Ihren Projekten von der ersten Analyse bis zur Inbetriebnahme Ihrer Applikation. In der Turck-Produktdatenbank finden Sie Software-Tools für Programmierung, Konfiguration oder Inbetriebnahme, Datenblätter und CAD-Dateien in vielen Exportformaten. Über folgende Adresse gelangen Sie direkt in die Produktdatenbank: [www.turck.de/produkte](http://www.turck.de/produkte)  
Für weitere Fragen ist das Sales-und-Service-Team in Deutschland telefonisch unter folgenden Nummern zu erreichen:

Vertrieb: +49 208 4952-380

Technik: +49 208 4952-390

Internet: [www.turck.com/support](http://www.turck.com/support)

Außerhalb Deutschlands wenden Sie sich bitte an Ihre Turck-Landesvertretung.

Hans Turck GmbH & Co. KG  
Witzlebenstraße 7  
45472 Mülheim an der Ruhr  
Germany

## 3 Zu Ihrer Sicherheit

Das Produkt ist nach dem Stand der Technik konzipiert. Dennoch gibt es Restgefahren. Um Personen- und Sachschäden zu vermeiden, müssen Sie die Sicherheits- und Warnhinweise beachten. Für Schäden durch Nichtbeachtung von Sicherheits- und Warnhinweisen übernimmt Turck keine Haftung.

### 3.1 Bestimmungsgemäße Verwendung

Die Geräte sind ausschließlich zum Einsatz im industriellen Bereich bestimmt. Das Blockmodul TBEN-L5-4RFID-8DXP-WIN ist ein RFID-Interface zum Einsatz im Turck RFID-System. Das Turck RFID-System dient dem berührungslosen Austausch von Daten zwischen einem Datenträger und einem Schreib-Lese-Kopf zur Identifizierung von Objekten. Die Interfaces kommunizieren über TCP/IP mit Drittsystemen wie beispielsweise ERP-Systemen. Die Gerätefunktionen können über das Betriebssystem Windows Embedded Compact 2013 mit .Net, C++ oder C# programmiert werden. Zudem können Middleware-Funktionalitäten auf dem Gerät integriert werden.

Die Geräte dürfen nur wie in dieser Anleitung beschrieben verwendet werden. Jede andere Verwendung gilt als nicht bestimmungsgemäß; für daraus resultierende Schäden übernimmt Turck keine Haftung.

### 3.2 Allgemeine Sicherheitshinweise

- Nur fachlich geschultes Personal darf das Gerät montieren, installieren, betreiben und instand halten.
- Das Gerät nur in Übereinstimmung mit den geltenden nationalen und internationalen Bestimmungen, Normen und Gesetzen einsetzen.
- Das Gerät erfüllt ausschließlich die EMV-Anforderungen für den industriellen Bereich und ist nicht zum Einsatz in Wohngebieten geeignet.

## 4 Produktbeschreibung

Die Geräte sind in einem vollvergossenen Kunststoffgehäuse in Schutzart IP67/IP69K ausgeführt. Zum Anschluss von Schreib-Lese-Köpfen stehen vier RFID-Kanäle zur Verfügung. Zusätzlich lassen sich Sensoren und Aktuatoren über acht frei als Eingänge oder Ausgänge konfigurierbare digitale I/O-Kanäle anschließen. Die Anschlüsse für Schreib-Lese-Köpfe und für digitale I/Os sind als M12-Buchsen ausgeführt. Zum Anschluss an das Ethernet steht eine M12-Buchse zur Verfügung.

### 4.1 Geräteübersicht

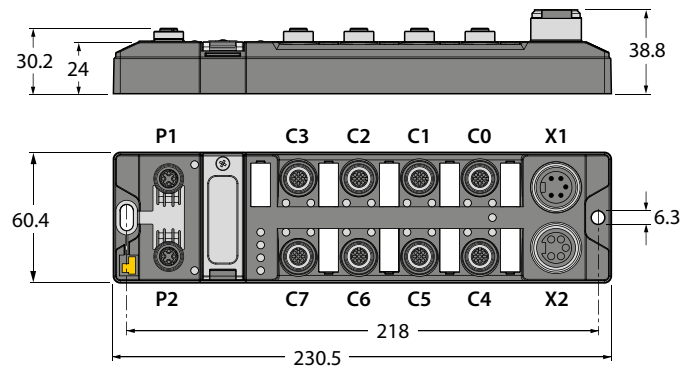


Abb. 1: Abmessungen

#### 4.1.1 Anzeigeelemente

Die Geräte verfügen über konfigurierbare Mehrfarben-LEDs.

### 4.2 Eigenschaften und Merkmale

- Kommunikation über TCP/IP
- Frei programmierbares Kompaktmodul basierend auf Windows Embedded Compact 2013
- Programmiersprache .Net, C++, C#
- API auf Anfrage verfügbar
- 4 Kanäle mit M12-Anschluss für RFID
- 8 konfigurierbare digitale Kanäle als pnp-Eingänge und/oder Ausgänge 2 A
- Integrierter Ethernet-Switch ermöglicht Linientopologie
- Übertragungsrate 10 Mbps/100 Mbps
- Vollvergossene Modulelektronik
- Schutzart IP65/IP67/IP69K
- LEDs zur Statusanzeige

## 4.3 Funktionsprinzip

Die RFID-Interfaces verbinden das RFID-System mit weiteren Systemen, die über TCP/IP kommunizieren (z. B. ERP-Systemen). Die Interfaces verfügen über eine Ethernet-Schnittstelle und RFID-Schnittstellen.

Über die TCP/IP-Schnittstelle wird das RFID-System an ein Drittsystem wie beispielsweise ein ERP-System angekoppelt. Über die RFID-Schnittstellen werden die Schreib-Lese-Köpfe an die Interfaces angeschlossen. Zusätzlich können die Interfaces Signale von Sensoren und Aktuatoren über acht konfigurierbare digitale Kanäle verarbeiten.

## 4.4 Funktionen und Betriebsarten

An die RFID-Kanäle können HF- und UHF-Schreib-Lese-Köpfe angeschlossen werden. Auch der parallele Betrieb von HF- und UHF-Schreib-Lese-Köpfen an einem Gerät ist möglich. Das Interface kann autarke Steuerungsfunktionen ausführen.

Die Gerätefunktionen können über das Betriebssystem Windows Embedded Compact 2013 mit .Net, C++ oder C# programmiert werden. Zudem können Middleware-Funktionalitäten auf dem Gerät integriert werden.

### 4.4.1 Funktionen im Turck Service Tool

Das Gerät unterstützt die folgenden Funktionen des Turck Service Tools:

- IP-Adresse ändern
- Gerät auf Werkseinstellungen zurücksetzen
- Spannungsreset durchführen
- Wink-Kommando
- Firmware-Version abfragen

## 5 Montieren

Die Geräte müssen auf einer ebenen, vorgebohrten und geerdeten Montagefläche befestigt werden.

- Modul mit zwei M6-Schrauben auf der Montagefläche befestigen. Das maximale Anzugsdrehmoment für die Befestigung der Schrauben beträgt 1,5 Nm.

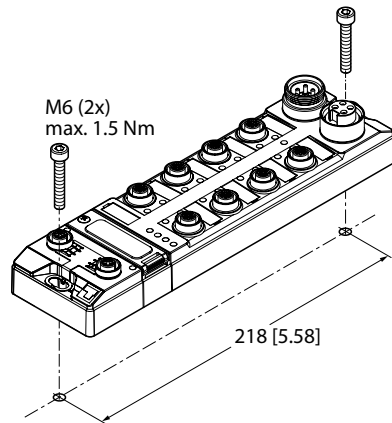


Abb. 2: Gerät auf Montageplatte befestigen

## 5.1 Gerät erden

### 5.1.1 Erdungs- und Schirmungskonzept

Das Erdungs- und Schirmungskonzept der TBEN-L-Module ermöglicht das getrennte Erden von Feldbus- und I/O-Teil.

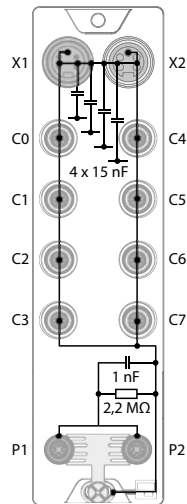


Abb. 3: Ersatzschaltbild, Schirmungskonzept

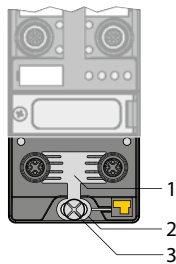


Abb. 4: Erdungskomponenten

Die Erdungsspanne (1) an den M12-Steckverbindern für den Feldbusanschluss (P1, P2) verbindet den Schirm der Feldbusleitungen.

Der Erdungsring (2) ist unterhalb der Erdungsspanne angebracht und verbindet die Funktionserde der 7/8"-Steckverbinder (Pin 3) für die Spannungsversorgung mit der Funktionserde der M12-Steckverbinder (Pin 5) für den Anschluss der Schreib-Lese-Köpfe, Sensoren und Aktuatoren.

Die Erdungsschraube (3) verbindet das Gerät mit dem Bezugspotenzial der Anlage.

## 5.1.2 Modul erden (FE)

Erdungsspange und Metallring sind miteinander verbunden. Eine Befestigungsschraube durch das untere Montageloch des Moduls verbindet die Schirmung der Feldbusleitungen mit der Funktionserde von Spannungsversorgung und angeschlossenen Geräten und dem Bezugspotenzial der Anlage.

Ist ein gemeinsames Bezugspotenzial nicht erwünscht, Erdungsspange zur Entkopplung des Feldbuschirms entfernen oder Modul mit einer Kunststoffschraube befestigen.

### Erdungsspange entfernen

- ▶ Erdungsspange mit einem flachen Schlitz-Schraubendreher nach oben hebeln und entfernen.

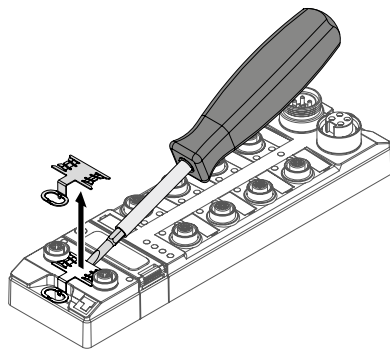


Abb. 5: Erdungsspange entfernen

### Erdungsspange montieren

- ▶ Erdungsspange ggf. mit Hilfe eines Schraubendrehers zwischen den Feldbus-Steckverbindern so wieder einsetzen, dass Kontakt zum Metallgehäuse der Steckverbinder besteht.
- ▶ Der Schirm der Feldbusleitungen liegt auf der Erdungsspange auf.

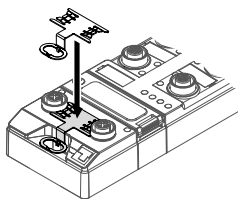


Abb. 6: Erdungsspange montieren

## 6 Anschließen

### 6.1 Module an das Ethernet anschließen

Zum Anschluss an ein TCP/IP-System verfügt das Gerät über einen integrierten Autocrossing-Switch mit zwei 4-poligen M12-Ethernet-Steckverbindern. Das max. Anzugdrehmoment beträgt 0,6 Nm.

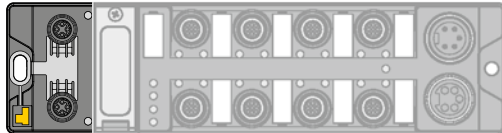


Abb. 7: M12-Ethernet-Steckverbinder zum Anschluss an ein TCP/IP-System

► Gerät gemäß unten stehender Pinbelegung an ein TCP/IP-System anschließen.

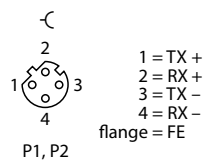


Abb. 8: Pinbelegung Ethernet-Anschlüsse



## 6.2 Versorgungsspannung anschließen

Zum Anschluss an die Versorgungsspannung verfügt das Gerät über zwei 5-polige 7/8"-Steckverbinder. V1 und V2 sind galvanisch voneinander getrennt. Das max. Anzugsdrehmoment beträgt 0,8 Nm.

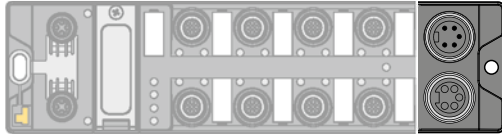


Abb. 9: 7/8"-Steckverbinder zum Anschluss an die Versorgungsspannung

► Gerät gemäß unten stehender Pinbelegung an die Versorgungsspannung anschließen.

| Pinbelegung |   |   |
|-------------|---|---|
|             | 1 BK = GND V2<br>2 BU = GND V1<br>3 GNYE = FE<br>4 BN = 24 VDC V1<br>5 WH = 24 VDC V2 | X1  |
|             |   | X2  |
|             |   | X1 Einspeisen der Spannung                            |
|             |   | X2 Weiterführen der Spannung zum nächsten Teilnehmer  |
|             |   | V1 Versorgungsspannung 1 (inkl. Elektronikversorgung) |
|             |   | V2 Versorgungsspannung 2                              |

Abb. 10: Pinbelegung Versorgungsspannungs-Anschlüsse



### HINWEIS

Die Systemspannung (V1) und die Lastspannung (V2) werden separat eingespeist und überwacht. Bei einer Unterschreitung der zulässigen Spannung werden die Steckplätze gemäß Versorgungskonzept des Modultyps abgeschaltet. Bei einer Unterschreitung von V2 wechselt die LED PWR von Grün auf Rot. Bei einer Unterschreitung von V1 erlischt die LED PWR.

### 6.3 RFID-Schreib-Lese-Köpfe anschließen

Zum Anschluss von RFID-Schreib-Lese-Köpfen verfügt das Gerät über vier 5-polige M12-Steckverbinder. Das max. Anzugsdrehmoment beträgt 0,8 Nm.

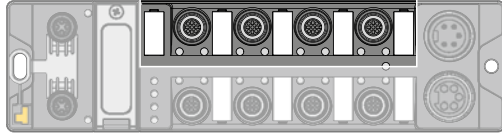


Abb. 11: M12-Steckverbinder zum Anschluss von Schreib-Lese-Köpfen

► Schreib-Lese-Köpfe gemäß unten stehender Pinbelegung an das Gerät anschließen.

| Pinbelegung   | Pin   |
|---|---|
| <p>1 = V<sub>AUX1</sub><br/>2 = Data A<br/>3 = GND<br/>4 = Data B<br/>5 = FE/Schirm</p> | <p>1 V<sub>AUX1</sub></p> <hr/> <p>2 TX/RX-</p> <hr/> <p>3 GND (V1)</p> <hr/> <p>4 TX/RX+</p> <hr/> <p>5 FE</p> |

Abb. 12: RS485 – Pinbelegung Schreib-Lese-Kopf-Anschlüsse

| Pinbelegung   | Pin   |
|---|---|
| <p>1 = BN (+)<br/>2 = BK (Data)<br/>3 = BU (GND)<br/>4 = WH (Data)<br/>5 = shield</p> | <p>1 V<sub>AUX1</sub></p> <hr/> <p>2 Data</p> <hr/> <p>3 GND (V1)</p> <hr/> <p>4 Data</p> <hr/> <p>5 FE</p> |

Abb. 13: Verbindungsleitungen .../S2500 – Pinbelegung Schreib-Lese-Kopf-Anschlüsse

| Pinbelegung   | Pin   |
|---|---|
| <p>1 = BN (+)<br/>2 = WH (Data)<br/>3 = BU (GND)<br/>4 = BK (Data)<br/>5 = shield</p> | <p>1 V<sub>AUX1</sub></p> <hr/> <p>2 Data</p> <hr/> <p>3 GND (V1)</p> <hr/> <p>4 Data</p> <hr/> <p>5 FE</p> |

Abb. 14: Verbindungsleitungen .../S2501 – Pinbelegung Schreib-Lese-Kopf-Anschlüsse

| Pinbelegung   | Pin   |
|---|---|
| <p>1 = RD (+)<br/>2 = BU (Data)<br/>3 = BK (GND)<br/>4 = WH (Data)<br/>5 = shield</p> | <p>1 V<sub>AUX1</sub></p> <hr/> <p>2 Data</p> <hr/> <p>3 GND (V1)</p> <hr/> <p>4 Data</p> <hr/> <p>5 FE</p> |

Abb. 15: Verbindungsleitungen .../S2503 – Pinbelegung Schreib-Lese-Kopf-Anschlüsse

## 6.4 Digitale Sensoren und Aktuatoren anschließen

Zum Anschluss von digitalen Sensoren und Aktuatoren verfügt das Gerät über vier 5-polige M12-Steckverbinder. Das max. Anzugsdrehmoment beträgt 0,8 Nm.

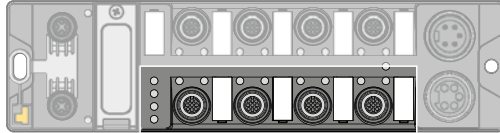


Abb. 16: M12-Steckverbinder zum Anschluss von digitalen Sensoren und Aktuatoren

► Sensoren und Aktuatoren gemäß unten stehender Pinbelegung an das Gerät anschließen.

### Pinbelegung



Abb. 17: Pinbelegung Anschlüsse für digitale Sensoren und Aktuatoren

## 7 In Betrieb nehmen

### 7.1 IP-Adresse einstellen

Die IP-Adresse lässt sich über zwei dezimale Drehcodierschalter und DIP-Schalter am Gerät oder über das Turck Service Tool einstellen.

#### 7.1.1 IP-Adresse über Schalter am Gerät einstellen

Die IP-Adresse kann über zwei dezimale Drehcodierschalter und den DIP-Schalter „Mode“ am Gerät eingestellt werden. Die Schalter befinden sich gemeinsam mit den USB-Ports und dem SET-Taster unter einer Abdeckung.

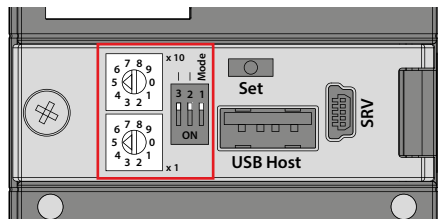


Abb. 18: Schalter zum Einstellen der IP-Adresse

- ▶ Abdeckung über den Schaltern öffnen.
- ▶ Drehcodierschalter gemäß unten stehender Tabelle auf die gewünschte Position einstellen.
- ▶ DIP-Schalter „Mode“ gemäß unten stehender Tabelle auf die gewünschte Position einstellen.
- ▶ Spannungsreset durchführen.
- ▶ **ACHTUNG!** Bei geöffneter Abdeckung über den Drehcodierschaltern ist die Schutzart IP67 oder IP69K nicht gewährleistet. Geräteschäden durch eindringende Fremdkörper oder Flüssigkeiten sind möglich. Abdeckung über den Schaltern fest verschließen.

## Adressierungsmöglichkeiten

Die IP-Adresse der Interfaces lässt sich auf unterschiedliche Weise einstellen. Folgende Adressierungsmöglichkeiten können über die Schalter am Gerät ausgewählt werden. Änderungen der Einstellung werden nach einem Spannungsreset aktiv.

| Einstellmöglichkeit | DIP-Schalter „MODE“ | Drehcodierschalter | Beschreibung   |
|---------------------|---------------------|--------------------|--|
| Default-Adresse     | 0                   | 00                 | IP-Adresse: 192.168.1.100<br>Subnetzmaske: 255.255.255.0<br>Gateway: 192.168.1.1   |
| Rotary-Modus        | 0                   | 01...99            | Im Rotary-Modus kann das letzte Byte der IP-Adresse manuell am Gateway eingestellt werden. Die weiteren Netzwerkeinstellungen sind nichtflüchtig im Speicher des Gateways hinterlegt und können im Rotary-Modus nicht verändert werden. Einstellbar sind Adressen von 1...99.  |
| DHCP-Modus          | 1                   | 40                 | Im DHCP-Modus wird die vollständige IP-Adresse automatisch von einem DHCP-Server im Netzwerk vergeben. Die vom DHCP-Server zugewiesene Subnetzmaske und die Default-Gateway-Adresse werden nichtflüchtig im Speicher des Gateways hinterlegt. DHCP unterstützt drei Arten der IP-Adresszuweisung:<br>Automatische Adressvergabe: Der DHCP-Server vergibt eine permanente IP-Adresse an den Client.<br>Dynamische Adressvergabe: Die vom Server vergebene IP-Adresse ist immer nur für einen bestimmten Zeitraum reserviert. Nach Ablauf dieser Zeit oder nach der expliziten Freigabe durch einen Client wird die IP-Adresse neu vergeben.<br>Manuelle Adressvergabe: Ein Netzwerk-Administrator weist dem Client eine IP-Adresse zu. DHCP wird in diesem Fall nur zur Übermittlung der zugewiesenen IP-Adresse an den Client genutzt. |
| PGM-Modus           | 1                   | 50                 | Im PGM-Modus wird die vollständige IP-Adresse manuell über das Turck Service Tool vergeben. Im PGM-Modus werden die eingestellte IP-Adresse und die Subnetzmaske im Speicher des Gateways hinterlegt. Alle Netzwerk-Einstellungen (IP-Adresse, Subnetzmaske, Default-Gateway) werden vom internen EEPROM des Moduls übernommen.  |
| F_Reset             | 1                   | 90                 | Dieser Modus setzt alle Einstellungen des Geräts auf die Default-Werte zurück und löscht alle Daten im internen Flash des Geräts. Die folgenden Werte werden zurückgesetzt bzw. gelöscht:<br>– IP-Adresse und Subnetzmaske<br>– Parameter  |
| Restore             | 1                   | 00                 | IP-Adresse: 192.168.1.100<br>Netzwerkmaske: 255.255.255.0<br>Gateway: 192.168.1.1  |

## 7.1.2 IP-Adresse über das Turck Service Tool einstellen

Im Lieferzustand besitzt das Gerät die IP-Adresse 192.168.1.100. Die IP-Adresse muss über das Turck Service Tool eingestellt werden. Das Turck Service Tool steht unter [www.turck.com](http://www.turck.com) zum kostenlosen Download zur Verfügung.

- Turck Service Tool öffnen.
- „Suchen“ klicken oder F5 drücken.

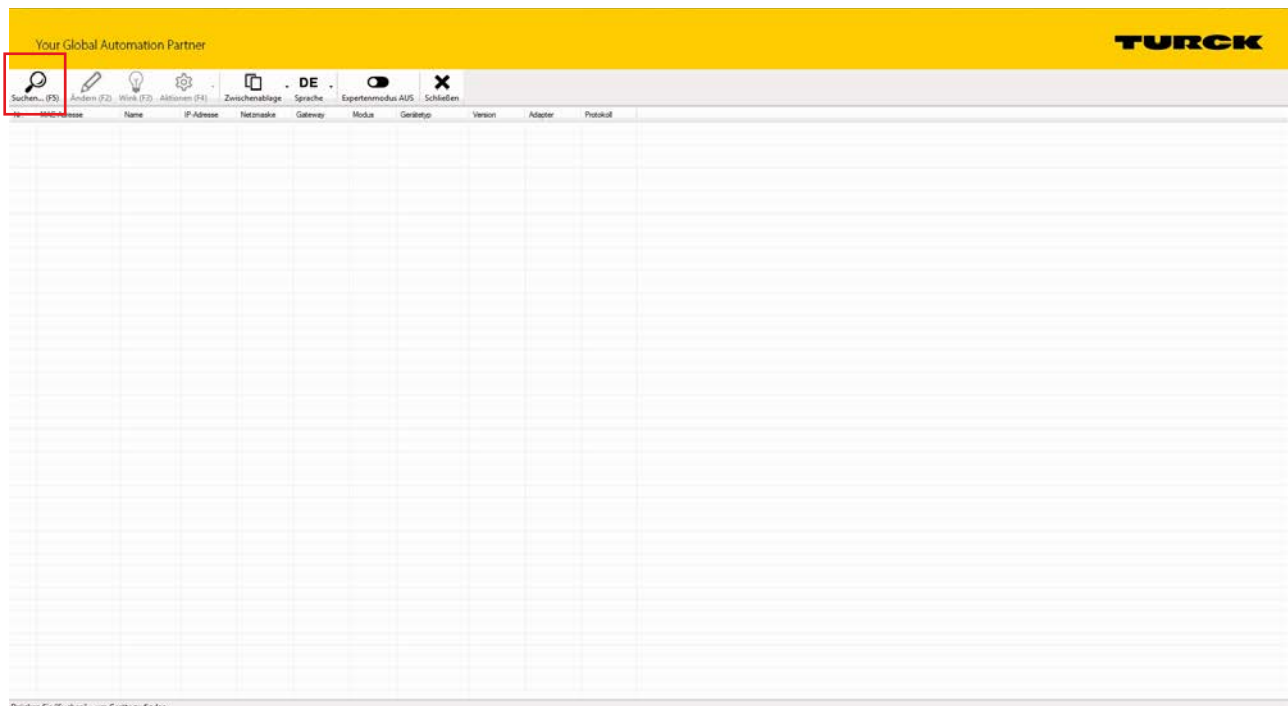


Abb. 19: Turck Service Tool – Startbildschirm

➔ Das Turck Service Tool zeigt die angeschlossenenen Geräte an.

| Nr. | MAC-Adresse       | Name | IP-Adresse    | Netzmaske     | Gateway     | Modus    | Gerätetyp              | Version | Adapter      | Protokoll |
|-----|-------------------|------|---------------|---------------|-------------|----------|------------------------|---------|--------------|-----------|
| 1   | 00:07:46:FF:A2:B7 |      | 192.168.1.100 | 255.255.255.0 | 192.168.1.1 | PGM_DHCP | TBEN-L5-4RFID-8DXP-WIN | 2.1.1.0 | 192.168.1.50 | Turck     |

Abb. 20: Turck Service Tool – Gefundene Geräte

- Gewünschtes Gerät anklicken.
- „Ändern“ klicken oder F2 drücken.

| Nr. | MAC-Adresse       | Name | IP-Adresse    | Netzmaske     | Gateway     | Modus    | Gerätetyp              | Version | Adapter      | Protokoll |
|-----|-------------------|------|---------------|---------------|-------------|----------|------------------------|---------|--------------|-----------|
| 1   | 00:07:46:FF:A2:B7 |      | 192.168.1.100 | 255.255.255.0 | 192.168.1.1 | PGM_DHCP | TBEN-L5-4RFID-8DXP-WIN | 2.1.1.0 | 192.168.1.50 | Turck     |

Abb. 21: Turck Service Tool – zu adressierendes Gerät auswählen.

- IP-Adresse sowie ggf. Netzwerkmaske und Gateway ändern.
- Änderungen mit einem Klick auf „Im Gerät setzen“ übernehmen.

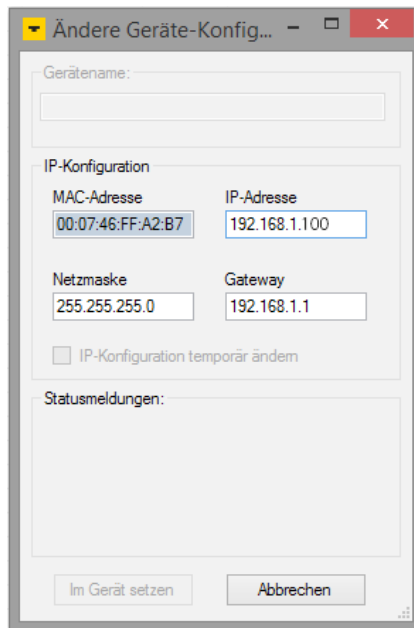


Abb. 22: Turck Service Tool – Geräte-Konfiguration ändern



## 7.2 Systemstart

### 7.2.1 Voraussetzungen

- Windows-Betriebssystem (mind. Windows 7)
- Visual Studio 2012 / 2013 / 2015  
(<https://www.visualstudio.com/de-de/products/visual-studio-express-vs.aspx>)
- Application Builder (<https://www.microsoft.com/en-us/download/details.aspx?id=38819>)
- SDK (e.g. Turck\_AM335x\_RFID\_SDKx.msi)

### 7.2.2 Anwendungen erstellen

#### Anwendung in C/C++ erstellen

- ▶ Sicherstellen, dass der SDK im System installiert ist (z. B. Turck\_AM335x\_RFID\_SDKx.msi).
- ▶ Visual Studio starten.
- ▶ New Project... wählen.
- ▶ Im linken Fensterbereich unter Templates → Visual C++ -> Windows Embedded Compact, „AM335x\_Turck\_RFID\_SDKx“ wählen.
- ▶ Im mittleren Fensterbereich „Win32 Console Application“ wählen.
- ▶ Im unteren Bereich einen Namen eingeben.
- ▶ OK klicken. Darauf startet Visual Studio im Application Builder View mit einer automatisch generierten, einfachen Console-Anwendung.
- ▶ „#include “windows.h”“ hinzufügen.
- ▶ Die Funktion wmain mit den folgenden Zeilen ergänzen:

```
printf("Hello World\n");
Sleep(5000);
```

- ▶ Zur Kompilierung der Anwendung F7 drücken.

#### Anwendung in C# erstellen

- ▶ Sicherstellen, dass der SDK im System installiert ist (z. B. Turck\_AM335x\_RFID\_SDKx.msi).
- ▶ Visual Studio starten.
- ▶ New Project... wählen.
- ▶ Im linken Fensterbereich unter Templates → Other Languages -> Visual C++ -> Windows Embedded Compact, „AM335x\_Turck\_RFID\_SDKx“ wählen.
- ▶ Im mittleren Fensterbereich „Console Application“ wählen.
- ▶ Im unteren Bereich einen Namen eingeben.
- ▶ OK klicken. Darauf startet Visual Studio im Application Builder View mit einer automatisch generierten, einfachen Console-Anwendung.
- ▶ Am Anfang der Datei main.cs „using System.Threading;“ und „using System.Diagnostics;“ hinzufügen
- ▶ Im Main Method folgende Zeilen hinzufügen:

```
Debug.WriteLine("Hello World");
Thread.Sleep(5000);
```

- ▶ Zur Kompilierung der Anwendung F7 drücken.

## 7.3 Treiber

### 7.3.1 Ethernet

Der Ethernet-Treiber unterstützt den Ethernet-Controller AM335x CPSW3G im Schalt-Modus.

Die externen Ports des Ethernet-Controllers werden auf die Stecker X\_P1 und X\_P2 abgebildet. Der interne Port des Ethernet-Switches steht der Anlage als Ethernet-Device CPSW3G1 zur Verfügung.

Der FTP-Server und Telnet-Server werden standardmäßig aktiviert (ohne Authentifizierung).

Einzelheiten über WinSock finden Sie unter <https://msdn.microsoft.com/EN-US/library/ee494651%28v=VS.80,d=HV.2%29.aspx>.

### 7.3.2 NandFlash

Der NandFlash-Treiber unterstützt die Steuerung AM335x GPMC und die gesteckte NANDFlash-Karte.

Der NANDFlash-Treiber wird automatisch am Start geladen und bildet transparent die FAT-Partition standardmäßig auf das Verzeichnis "/Mounted\_Volume" ab.

Das Geräte-Verzeichnis wird standardmäßig permanent im Dateisystem NANDFlash gespeichert.

### 7.3.3 USB-Host

Der USB-Host unterstützt den Ethernet-Controller AM335x USB1 im Host-Modus. Der Host steht am Stecker X25 zur Verfügung.

Wenn das Gerät gesteckt ist und vom System unterstützt wird, werden USB-Gerätetreiber automatisch geladen. Human Interface Devices (MMIs) und USB-Massenspeicher werden standardmäßig unterstützt.

### 7.3.4 USB OTG

Im Device-Modus unterstützt der USB-OTG-Treiber den Ethernet-Controller AM335x USB0. Der USB OTG steht am Stecker X18 zur Verfügung.

Standardmäßig wird der USB-OTG-Gerätetreiber auf USB Serial-Modus konfiguriert.

### 7.3.5 UART

Der UART-Treiber unterstützt die UART1-, UART2-, UART3- und UART4-Geräte der AM335x. Nur RX/TX werden vom Treiber unterstützt. Signale für eine Flusskontrolle sind nicht verfügbar. Die UARTs werden als COM1: - COM4 erreicht: Geräte, die die Win32 Serial Port API verwenden. Einzelheiten finden Sie unter

<https://msdn.microsoft.com/EN-US/library/ee488234%28v=VS.80,d=HV.2%29.aspx>.

Um den TBOX-Kanälen zu entsprechen, werden die Prozessor-UARTs auf die COM-Ports wie folgt abgebildet:

- UART1 → COM3:
- UART2 → COM4:
- UART3 → COM1:
- UART4 → COM2:

Durch Ändern des Werts von Key

"HKEY\_LOCAL\_MACHINE\Drivers\BuiltIn\UARTX\RxFifoTriggerLevel" (mögliche Werte 1-63) kann das Empfangs-FIFO des UARTs beeinflusst werden.

### 7.3.6 GPIO

Der GPIO-Treiber unterstützt die AM335x-GPIOs sowie die PCA9506-GPIO-Expander und die XMC-LED-GPIOs.

Der GPIO-Treiber steht als GIO1: Gerät zur Verfügung. Der Treiber ist über die Treiber-Funktionen des Stream-Interfaces verfügbar. Für Einzelheiten siehe

<https://msdn.microsoft.com/EN-US/library/ee488234%28v=VS.80,d=HV.2%29.aspx>.

Die GPIOs müssen über ihre GPIO-ID identifiziert werden. Sie werden in der Datei gpio\_defines.h definiert, die in der SDK enthalten ist. Folgende GPIOs werden von der BSP unterstützt:

| AM335x GPIO | GPIO ID | GPIO define |
|-------------|---------|-------------|
| GPIO0_7     | 7       | GPIO_7      |
| GPIO0_12    | 12      | GPIO_12     |
| GPIO0_13    | 13      | GPIO_13     |
| GPIO0_19    | 19      | GPIO_19     |
| GPIO0_23    | 23      | GPIO_23     |
| GPIO1_15    | 47      | GPIO_47     |
| GPIO1_31    | 63      | GPIO_63     |
| GPIO2_22    | 86      | GPIO_86     |
| GPIO2_23    | 87      | GPIO_87     |
| GPIO2_24    | 88      | GPIO_88     |
| GPIO2_25    | 89      | GPIO_89     |
| GPIO3_14    | 110     | GPIO_110    |
| GPIO3_15    | 111     | GPIO_111    |
| GPIO3_16    | 112     | GPIO_112    |
| GPIO3_17    | 113     | GPIO_113    |
| GPIO3_18    | 114     | GPIO_114    |
| GPIO3_20    | 116     | GPIO_116    |
| GPIO3_21    | 117     | GPIO_117    |

| XMC GPIO  | GPIO ID | GPIO define |
|-----------|---------|-------------|
| P0.13     | 208     | XGPIO_0     |
| P0.12     | 209     | XGPIO_1     |
| P1.1      | 210     | XGPIO_2     |
| P1.0      | 211     | XGPIO_3     |
| P0.5      | 212     | XGPIO_4     |
| P0.4      | 213     | XGPIO_5     |
| P0.11     | 214     | XGPIO_6     |
| P0.10     | 215     | XGPIO_7     |
| P1.6      | 216     | XGPIO_8     |
| PCA GPIO  | GPIO ID | GPIO define |
| PCA0_0[0] | 128     | EGPIO_0     |
| ...       | ...     | ...         |
| PCA0_4[7] | 167     | EGPIO_39    |
| PCA1_0[0] | 168     | EGPIO_40    |
| ...       | ...     | ...         |
| PCA1_4[7] | 207     | EGPIO_79    |

Der Treiber unterstützt folgende IOControl Codes (in gpio\_ioctls.h definiert):

#### IOCTL\_GPIO\_SETBIT

Setzt den entsprechenden GPIO auf Level 1.

| Parameter  |  |
|------------|--|
| lpInBuffer | Pointer auf DWORD, das die zu setzende GPIO ID enthält |

#### IOCTL\_GPIO\_CLRBIT

Setzt den entsprechenden GPIO auf Level 0.

| Parameter  |  |
|------------|--|
| lpInBuffer | Pointer auf DWORD, das die zu setzende GPIO ID enthält |

#### IOCTL\_GPIO\_GETBIT

Liest aus dem Level des entsprechenden GPIOs.

| Parameter   |  |
|-------------|--|
| lpInBuffer  | Pointer auf DWORD, das die zu lesende GPIO ID enthält. |
| lpOutBuffer | Pointer auf DWORD, das den aktuellen Level empfängt    |

## IOCTL\_GPIO\_SETMODE

Konfiguriert den Modus des entsprechenden GPIOs.

| Parameter |   |
|-----------|---|
| pInBuffer | Pointer auf Array von zwei DWORDs, die die GPIO ID (Array-Element 0) und den zu setzenden Modus (Array-Element 1) enthält |

Folgende Modi werden unterstützt (in gpio\_defines.h definiert):

- GPIO\_DIR\_OUTPUT : konfiguriert den GPIO als Ausgang
- GPIO\_DIR\_INPUT : konfiguriert den GPIO als Eingang
- GPIO\_INT\_LOW\_HIGH : Aktivierung des Interrupts bei steigender Flanke
- GPIO\_INT\_HIGH\_LOW : Aktivierung des Interrupts bei fallender Flanke
- GPIO\_INT\_LOW : Aktivierung des Low-Level-Interrupts
- GPIO\_INT\_HIGH : Aktivierung des High-Level-Interrupt
- GPIO\_DEBOUNCE\_ENABLE : Aktivierung des Entprellens

Die Modi GPIO\_INT\_LOW\_HIGH, GPIO\_INT\_HIGH\_LOW, GPIO\_INT\_LOW, GPIO\_INT\_HIGH und GPIO\_DEBOUNCE\_ENABLE werden von den PCA- (EGPIO\_0 – EGPIO\_79) und XMC-GPIOs (XGPIO\_0 – XGPIO\_8) nicht unterstützt.

## IOCTL\_GPIO\_GETMODE

Gibt den aktuellen Modus des entsprechenden GPIOs zurück.

| Parameter   |   |
|-------------|---|
| lpInBuffer  | Pointer auf DWORD, das die GPIO ID enthält          |
| lpOutBuffer | Pointer auf DWORD, das den aktuellen Modus empfängt |

## IOCTL\_GPIO\_CONFIGURE\_PERIODIC\_UPDATE

Setzt den entsprechenden GPIO in Toggle Modus mit 1 Hz oder 2Hz.

| Parameter  |  |
|------------|--|
| lpInBuffer | Pointer auf eine Struktur IOCTL_GPIO_PERIODIC_INFO (in gpio_ioctls.h definiert), die die GPIO ID und den einzustellenden periodischen Modus enthält: |
|            | <pre>typedef struct {     UINT uGpioID;     UINT dwPeriMode; } IOCTL_GPIO_PERIODIC_INFO;</pre>   |

Folgende Modi werden unterstützt (in gpio\_defines.h definiert):

- GPIO\_PERIODIC\_DISABLE : Deaktiviert das periodische Umschalten (Toggling) des GPIOs
- GPIO\_PERIODIC\_1HZ : Aktivierung des periodischen Umschaltens des GPIOs mit 1Hz
- GPIO\_PERIODIC\_2HZ : Aktivierung des periodischen Umschaltens des GPIOs mit 2Hz

Diese IOCONTROL-Funktion wird von XGPIO\_8 GPIO nicht unterstützt.

## IOCTL\_GPIO\_CONFIGURE\_RXTX\_UPDATE

Konfiguriert den entsprechenden GPIO für RX/TX-Signale.

| Parameter  |   |
|------------|---|
| lpInBuffer | Pointer auf DWORD, das die zu konfigurierende GPIO ID enthält |

Die RX/TX-Signalisierung lässt sich nur durch Aktivierung eines anderen Modus über IOCTL\_GPIO\_CONFIGURE\_PERIODIC\_UPDATE, IOCTL\_GPIO\_SETBIT oder IOCTL\_GPIO\_CLRBIT deaktivieren.

Diese IOCONTROL-Funktion wird von GPIOs XGPIO\_0 bis XGPIO\_8 unterstützt.

## IOCTL\_GPIO\_CONFIGURE\_AUTO\_RECOVERY

Konfiguriert die automatische Wiederherstellung des GPIOs.

| Parameter  |   |
|------------|---|
| lpInBuffer | <p>Pointer auf eine Struktur IOCTL_OC_INFO (in gpio_ioctls.h definiert), die die GPIO ID zur Aktivierung der automatischen Wiederherstellung, die Überstromerkennung des GPIOs und den Modus (automatische oder manuelle Wiederherstellung) enthält:</p> <pre>typedef struct {     UINT uGpioID;     UINT uOCGpioID;     BYTE bOCLevel;     BYTE bMode; }</pre> |

Für den Parameter bOCLevel (in gpio\_defines.h definiert) werden folgende Bedingungen des Trigger-Levels unterstützt:

- GPIO\_AUTOREC\_OC\_LOWLEV: Überstrombedingung von uGpioID wird von einem Low-Level-Interrupt an der uOCGpioID ausgelöst
- GPIO\_AUTOREC\_OC\_HIGHLEV: Überstrombedingung von uGpioID wird von einem High-Level-Interrupt an der uOCGpioID ausgelöst
- GPIO\_AUTOREC\_OC\_SAMELEV: Überstrombedingung von uGpioID wird ausgelöst, wenn uOCGpioID auf dem gleichen Level liegt
- GPIO\_AUTOREC\_OC\_DIFFLEV: Überstrombedingung von uGpioID wird ausgelöst, wenn uOCGpioID auf dem ergänzenden Level liegt

Für den Parameter bMode werden folgende Modi (in gpio\_defines.h definiert) unterstützt:

- GPIO\_AUTOREC\_OC\_DISABLED: Überstromerkennung deaktiviert
- GPIO\_AUTOREC\_OC\_SHUTDOWN: Überstrombedingung wird mit Abschaltung behandelt (manuelle Wiederherstellung)
- GPIO\_AUTOREC\_OC\_AUTOREC: Überstrombedingung wird mit Abschaltung und nach einer sicheren Zeit mit automatischer Wiederherstellung behandelt

## IOCTL\_GPIO\_GETIRQ

Gibt den IRQ des entsprechenden GPIOs zurück.

| Parameter   |  |
|-------------|--|
| lpInBuffer  | Pointer auf DWORD, das die GPIO ID enthält     |
| lpOutBuffer | Pointer auf DWORD, das die IRQ-Nummer empfängt |

Diese IOCTL-Funktion wird von den PCA (EGPIO\_0 – EGPIO\_79 ) und XMC-GPIOs (XGPIO\_0 – XGPIO\_8) nicht unterstützt.

## IOCTL\_GPIO\_SET\_DEBOUNCE\_TIME

Stellt die Entprellzeit des GPIOs (Bank) ein.

| Parameter  |  |
|------------|--|
| lpInBuffer | Pointer auf eine Struktur IOCTL_GPIO_SET_DEBOUNCE_TIME_IN (in gpio_ioctls.h definiert), die die GPIO ID und einzustellende Entprellzeit enthält: |
|            | <pre>typedef struct {     UINT gpioId;     UINT debounceTime; } IOCTL_GPIO_SET_DEBOUNCE_TIME_IN;</pre>   |

Die Entprellzeit wird wie folgt berechnet:

Entprellzeit = (DEBOUNCETIME + 1) × 31 µs. Die Entprellzeit gilt global für alle GPIOs derselben Bank.

Diese IOCTL-Funktion wird von den PCA (EGPIO\_0 – EGPIO\_79 ) und XMC-GPIOs (XGPIO\_0 – XGPIO\_8) nicht unterstützt.

## IOCTL\_GPIO\_GET\_DEBOUNCE\_TIME

Gibt die Entprellzeit des GPIOs (Bank) zurück.

| Parameter   |  |
|-------------|--|
| lpInBuffer  | Pointer auf DWORD, das die GPIO ID enthält       |
| lpOutBuffer | Pointer auf DWORD, das die Entprellzeit empfängt |

Diese IOCTL-Funktion wird von den PCA (EGPIO\_0 – EGPIO\_79 ) und XMC-GPIOs (XGPIO\_0 – XGPIO\_8) nicht unterstützt.

## IOCTL\_GPIO\_INIT\_INTERRUPT

Initialisiert den Interrupt für den GPIO.

| Parameter  |   |
|------------|---|
| lpInBuffer | Pointer auf eine Struktur IOCTL_GPIO_INIT_INTERRUPT_INFO (in gpio_ioctls.h definiert):  |
|            | <pre>typedef struct {     UINT    uGpioID;     DWORD   dwSysIntrID;     HANDLE  hEvent; } IOCTL_GPIO_INIT_INTERRUPT_INFO;</pre> |

uGpioID muss auf die GPIO ID und hEvent auf einen Event-Handle gesetzt werden. Der verwendete SysIntr wird im Element dwSysIntrID zurückgegeben.

Diese IOCTL-Funktion wird von den PCA (EGPIO\_0 – EGPIO\_79 ) und XMC-GPIOs (XGPIO\_0 – XGPIO\_8) nicht unterstützt.

## IOCTL\_GPIO\_ACK\_INTERRUPT

Quittiert einen GPIO-Interrupt.

| Parameter  |  |
|------------|--|
| lpInBuffer | Pointer auf eine Struktur IOCTL_GPIO_INTERRUPT_INFO (in gpio_ioctls.h definiert):                      |
|            | <pre>typedef struct {     UINT    uGpioID;     DWORD   dwSysIntrID; } IOCTL_GPIO_INTERRUPT_INFO;</pre> |

Diese IOCTL-Funktion wird von den PCA (EGPIO\_0 – EGPIO\_79 ) und XMC-GPIOs (XGPIO\_0 – XGPIO\_8) nicht unterstützt.

## IOCTL\_GPIO\_DISABLE\_INTERRUPT

Deaktiviert den Interrupt eines GPIOs.

| Parameter  |   |
|------------|---|
| lpInBuffer | Pointer auf eine Struktur IOCTL_GPIO_INTERRUPT_INFO |

Diese IOCTL-Funktion wird von den PCA (EGPIO\_0 – EGPIO\_79 ) und XMC-GPIOs (XGPIO\_0 – XGPIO\_8) nicht unterstützt.



7.3.7 SPI

Der SPI-Treiber unterstützt den AM335x MCSPI0-Interface. Der SPI0-Interface steht als SPI1:Gerät zur Verfügung.

Der Treiber unterstützt folgende Funktionen (in sdk\_spi.h definiert):

HANDLE SPIOpen(LPCTSTR pSpiName)

Öffnet den Treiber für spätere Verwendung.

| Parameter          |                                      |
|--------------------|--------------------------------------|
| pSpiName           | String mit dem Gerätenamen („SPI1:“) |
| Rückgabewert       |                                      |
| Handle auf Treiber |                                      |

VOID SPIClose(HANDLE hContext)

Beendet den Treiber nach Gebrauch.

| Parameter |                                     |
|-----------|-------------------------------------|
| hContext  | Handle über SPIOpen() zurückgegeben |

BOOL SPILockController(HANDLE hContext, DWORD dwTimeout)

Sperrt den Zugriff auf den Treiber zum aktuellen Thread.

| Parameter    |  |
|--------------|--|
| hContext     | Handle von SPIOpen() zurückgegeben     |
| dwTimeout    | Timeout für die Aktivierung der Sperre |
| Rückgabewert |  |
| TRUE         | Erfolgreich                            |
| FALSE        | Nicht erfolgreich                      |

BOOL SPIUnlockController(HANDLE hContext)

Entsperrt den Zugriff auf den Treiber.  
Sperrt den Zugriff auf den Treiber zum aktuellen Thread.

| Parameter    |                                      |
|--------------|--------------------------------------|
| hContext     | Handle durch SPIOpen() zurückgegeben |
| Rückgabewert |                                      |
| TRUE         | Erfolgreich                          |
| FALSE        | Nicht erfolgreich                    |

BOOL SPIConfigure(HANDLE hContext, DWORD address, DWORD config)

Konfiguriert das SPI-Gerät für weitere Aktionen.

| Parameter |   |
|-----------|---|
| hContext  | Handle durch SPIOpen() zurückgegeben  |
| address   | Chipselect-Nummer (unterstützt wird nur CS0)  |
| config    | DWORD, das die erwünschte Konfiguration enthält. Die Konfiguration muss entsprechend der Beschreibung des Registers MCSPI_CH0CONF im Handbuch der AM335x (Technical Reference Manual) erfolgen. |

| Rückgabewert |                   |
|--------------|-------------------|
| TRUE         | Erfolgreich       |
| FALSE        | Nicht erfolgreich |

BOOL SPIEnableChannel(HANDLE hContext)

Aktiviert den vom Adressen-Parameter SPIConfigure() konfigurierten Kanal und daher auch den entsprechenden Chipselect.

| Parameter    |                                      |
|--------------|--------------------------------------|
| hContext     | Handle durch SPIOpen() zurückgegeben |
| Rückgabewert |                                      |
| TRUE         | Erfolgreich                          |
| FALSE        | Nicht erfolgreich                    |

BOOL SPIDisableChannel(HANDLE hContext)

Deaktiviert den vorher von SPIEnableChannel() aktivierten Kanal.

| Parameter    |                                      |
|--------------|--------------------------------------|
| hContext     | Handle durch SPIOpen() zurückgegeben |
| Rückgabewert |                                      |
| TRUE         | Erfolgreich                          |
| FALSE        | Nicht erfolgreich                    |

BOOL SPISetSlaveMode(HANDLE hContext)

Konfiguriert den SPI-Controller für Slave-Modus.

| Parameter    |                                      |
|--------------|--------------------------------------|
| hContext     | Handle durch SPIOpen() zurückgegeben |
| Rückgabewert |                                      |
| TRUE         | Erfolgreich                          |
| FALSE        | Nicht erfolgreich                    |

DWORD SPIRead(HANDLE hContext, DWORD size, VOID \*pBuffer)

Liest vom SPI-Bus.

| Parameter                              |                                      |
|--|--------------------------------------|
| hContext                               | Handle durch SPIOpen() zurückgegeben |
| size                                   | Anzahl der zu lesenden Bytes         |
| pBuffer                                | Pointer auf den receivebuffer        |
| Rückgabewert                           |                                      |
| Anzahl der tatsächlich gelesenen Bytes |                                      |

DWORD SPIWrite(HANDLE hContext, DWORD size, VOID \*pBuffer)

Schreibt auf den SPI-Bus.

| Parameter                                  |                                      |
|--|--------------------------------------|
| hContext                                   | Handle durch SPIOpen() zurückgegeben |
| size                                       | Anzahl der zu schreibenden Bytes     |
| pBuffer                                    | Pointer auf sendbuffer               |
| Rückgabewert                               |                                      |
| Anzahl der tatsächlich geschriebenen Bytes |                                      |

DWORD SPIWriteRead(HANDLE hContext, DWORD size, VOID \*pOutBuffer, VOID \*pInBuffer)

Liest/schreibt gleichzeitig vom/zum SPI-Bus.

| Parameter  |   |
|--|---|
| hContext   | Handle durch SPIOpen() zurückgegeben      |
| size   | Anzahl der zu lesenden/schreibenden Bytes |
| pOutBuffer   | Pointer auf sendbuffer                    |
| pInBuffer  | Pointer auf den receivebuffer             |
| Rückgabewert   |   |
| Anzahl der tatsächlich gelesenen/geschriebenen Bytes |   |

DWORD SPIAsyncWriteRead(HANDLE hContext, DWORD size, VOID \*pOutBuffer, VOID \*pInBuffer)

Liest/schreibt gleichzeitig vom/zum SPI-Bus über DMA.

| Parameter                |   |
|--------------------------|---|
| hContext                 | Handle durch SPIOpen() zurückgegeben      |
| size                     | Anzahl der zu lesenden/schreibenden Bytes |
| pOutBuffer               | Pointer auf sendbuffer                    |
| pInBuffer                | Nicht verwendet, auf NULL setzen          |
| Rückgabewert             |   |
| Wert des size-Parameters |   |

DWORD SPIWaitForAsyncWriteReadCompleet(HANDLE hContext, DWORD size, VOID \*pOutBuffer)

Wartet auf Abschluss des DMA-Transfers.

| Parameter                |                                      |
|--------------------------|--------------------------------------|
| hContext                 | Handle durch SPIOpen() zurückgegeben |
| size                     | Anzahl der zu schreibenden Bytes     |
| pOutBuffer               | Pointer auf den receivebuffer        |
| Rückgabewert             |                                      |
| Wert des size-Parameters |                                      |

### 7.3.8 I2C

Der (im usermode verfügbare) I2C Proxy-Treiber unterstützt den AM335x-I2C0-Interface. Der I2C0-Interface steht als I2C1:Gerät zur Verfügung.

Der I2C-Proxy-Treiber steht über die Datei API (CreateFile(), ReadFile(), WriteFile(), SetFilePointer()) zur Verfügung.

Zur Auswahl der in anschließenden ReadFile()- WriteFile()-Aufrufen verwendeten base-Subadresse wird SetFilePointer() verwendet. Zur Auswahl der Adresse und Baudrate des I2C-Geräts stehen folgende IOControl-Codes zur Verfügung (in i2cproxy.h definiert):

#### IOCTL\_I2C\_SET\_SLAVE\_ADDRESS

Setzt die Slave-Adresse des anzusprechenden I2C-Geräts.

| Parameter  |  |
|------------|--|
| lpInBuffer | Pointer auf DWORD, das die Slave-Adresse enthält |

#### IOCTL\_I2C\_SET\_SUBADDRESS\_MODE

Setzt den Subadress-Modus.

| Parameter  |  |
|------------|--|
| lpInBuffer | Pointer auf DWORD, das den gewünschten Subadress-Modus enthält. Folgende Modi sind verfügbar (in sdk_i2c.h definiert):<br>- I2C_SUBADDRESS_MODE_0: Keine Geräte-Subadressen<br>- I2C_SUBADDRESS_MODE_8: 1 Byte Subadressen<br>- I2C_SUBADDRESS_MODE_16: 2 Byte Subadressen<br>- I2C_SUBADDRESS_MODE_24: 3 Byte Subadressen<br>- I2C_SUBADDRESS_MODE_32: 4 Byte Subadressen |

## IOCTL\_I2C\_SET\_BAUD\_INDEX

Setzt die Baudrate der I2C.

| Parameter  |   |
|------------|---|
| lpInBuffer | – Pointer auf DWORD, das die gewünschte Baudrate enthält. Folgende Baudraten sind verfügbar (in sdk_i2c.h definiert):<br>– SLOWSPEED_MODE: 100 KHz<br>– FULLSPEED_MODE: 400 KHz<br>– HIGHSPEED_MODE_1P16: 1.6 MHz<br>– HIGHSPEED_MODE_2P4: 2.4 MHz<br>– HIGHSPEED_MODE_3P2: 3.2 MHz |

### 7.3.9 RTC

Der OAL unterstützt die Onboard-RTC. Die RTC wird vom System (OAL) automatisch verwendet. Daher erfolgt das Auslesen und Einstellen der Zeit über z. B. die Funktionen GetSystemTime() und SetSystemTime() oder über Console-Funktionen für Datum und Zeit. Die RTC wird beim Systemstart (lesen) und beim Ändern des Werts (schreiben) synchronisiert.

### 7.3.10 Anwendung einsetzen

Damit das Gerät auf Deploy-/Debug-Anforderungen des Application Builders abhört, müssen geräteseitige Komponenten manuell nach dem Start des Geräts aktiviert werden. Dies kann zum Beispiel über telnet durchgeführt werden:

- Das laufende Gerät über telnet verbinden (Default-Adresse IP 192.168.1.100).
- "start conmanclient3 & cmanaccept3" in der Befehlszeile des Geräts ausführen.

Nachdem das Gerät für die Verbindung mit dem Application Builder vorbereitet ist, kann die Anwendung durch Drücken von F5 (oder über "Debug → Start Debugging") am Gerät verwendet und ausgeführt werden.

Der Application Builder sollte nach Debug View wechseln, die Anwendung zum Gerät übertragen und ausführen. Beim Ausführen des oben erwähnten Anwendungsbeispiel sollte der String "Hello World" im Output-Fenster (Debug) des Application Builders sichtbar sein.

## 7.3.11 Debugging der Anwendung

Zum Einfügen eines Haltepunkts in eine spezifische Zeile des Quellcodes klicken Sie auf den senkrechten grauen Balken vor der Zeile. Der Haltepunkt wird mit einem roten Punkt gekennzeichnet.

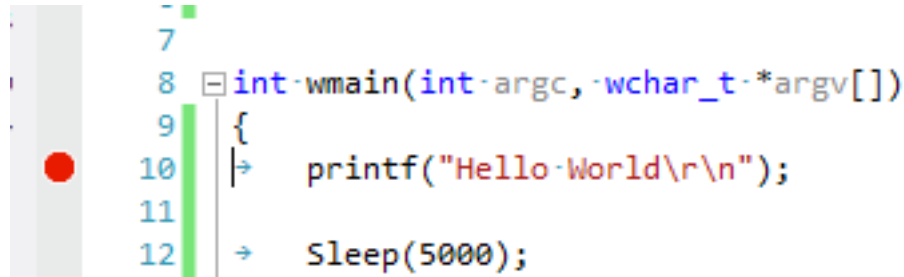


Abb. 23: Debugging der Anwendung

Beim nächsten Einsatz der Anwendung und beim Ausführen auf dem Zielgerät hält sie am/an den vorgewählten Haltepunkt/en. Dies wird durch einen gelben Pfeil oben auf dem Haltepunkt angezeigt.

Ein Remote Debugging über Visual Studio ist dem lokalen Debugging einer Anwendung, einschließlich der Einzelschritte, Callstack-Anzeige (Aufrufliste), Speicheranzeige usw. sehr ähnlich.

## 7.3.12 Verwendung eines Netzwerk-Sockets in C#

Über die Socket-Klasse kann eine Netzkommunikation in C# implementiert werden. Über folgende Beispielcodes kann ein TCP-Socket zu einem Server an 192.168.1.100 auf Port 80 erstellt und geöffnet werden. Weitere Informationen über die Kommunikation über den Socket sehen Sie in [https://msdn.microsoft.com/en-us/library/system.net.sockets.socket\\_members\(v=vs.90\)](https://msdn.microsoft.com/en-us/library/system.net.sockets.socket_members(v=vs.90)).

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Diagnostics;

using System.Net;
using System.Net.Sockets;

namespace TurckWinSock
{
    class Program
    {
        static void Main(string[] args)
        {
            Debug.WriteLine("Hello World");

            //einen Endpunkt mit dem Server IP und Port 80
            erstellen
            IPAddress ip = new IPAddress(new byte[] {192, 168, 4,
            80});
            IPEndPoint IpEnd = new IPEndPoint(ip, 80);
```

```

        //create a socket object
        Socket TestSocket = new Socket (IpEnd.AddressFamily,
SocketType.Stream, ProtocolType.Tcp);

        try
        {
            //Socket mit dem Server verbinden
            TestSocket.Connect (IpEnd);

            //Verbindung prüfen
            if (TestSocket.Connected)
            {
                Debug.WriteLine („socket connected“);
            }
            else
            {
                Debug.WriteLine („socket connection failed“);
            }

            //Etwas mit dem Socket machen
            Thread.Sleep (5000);

            //Socket schließen
            TestSocket.Shutdown (SocketShutdown.Both);
            TestSocket.Close ();
        }
        catch (Exception e)
        {
            Debug.WriteLine (“exception while connecting
socket”);
        }
    }
}

```

## 7.3.13 TBOX-API-Bibliothek verwenden

Zum Abstrahieren von Funktionen der DXPs und der COM-Ports steht eine API-Bibliothek (TBOX API) zur Verfügung.

Eine Übersicht und Beschreibung der Funktionen der API-Bibliothek steht in der Header-Datei TBOX\_API.h zur Verfügung.

### Vorgehensweise mit einer C/C++ Anwendung

Zur Verwendung der API-Bibliothek in einer C/C++ Anwendung, binden Sie die Header-Datei TBOX\_API.h ein und erstellen Sie eine statische Verknüpfung der TBOX\_API\_LIB.lib mit der Anwendung.

Kopieren Sie TBOX\_API.h und TBOX\_API\_LIB.lib in Ihr Anwendungsverzeichnis und fügen Sie die Bibliothek als zusätzliche Abhängigkeit zum Linker hinzu:

- „PROJECT“ → „Project Properties“ wählen (Alt + F7).
- Im linken Fensterbereich „Configuration Properties“ → „Linker“ → „Input“ wählen.
- Im rechten Fensterbereich auf die Dropdown-Liste „Additional Dependencies“ klicken und „<Edit...>“ wählen.

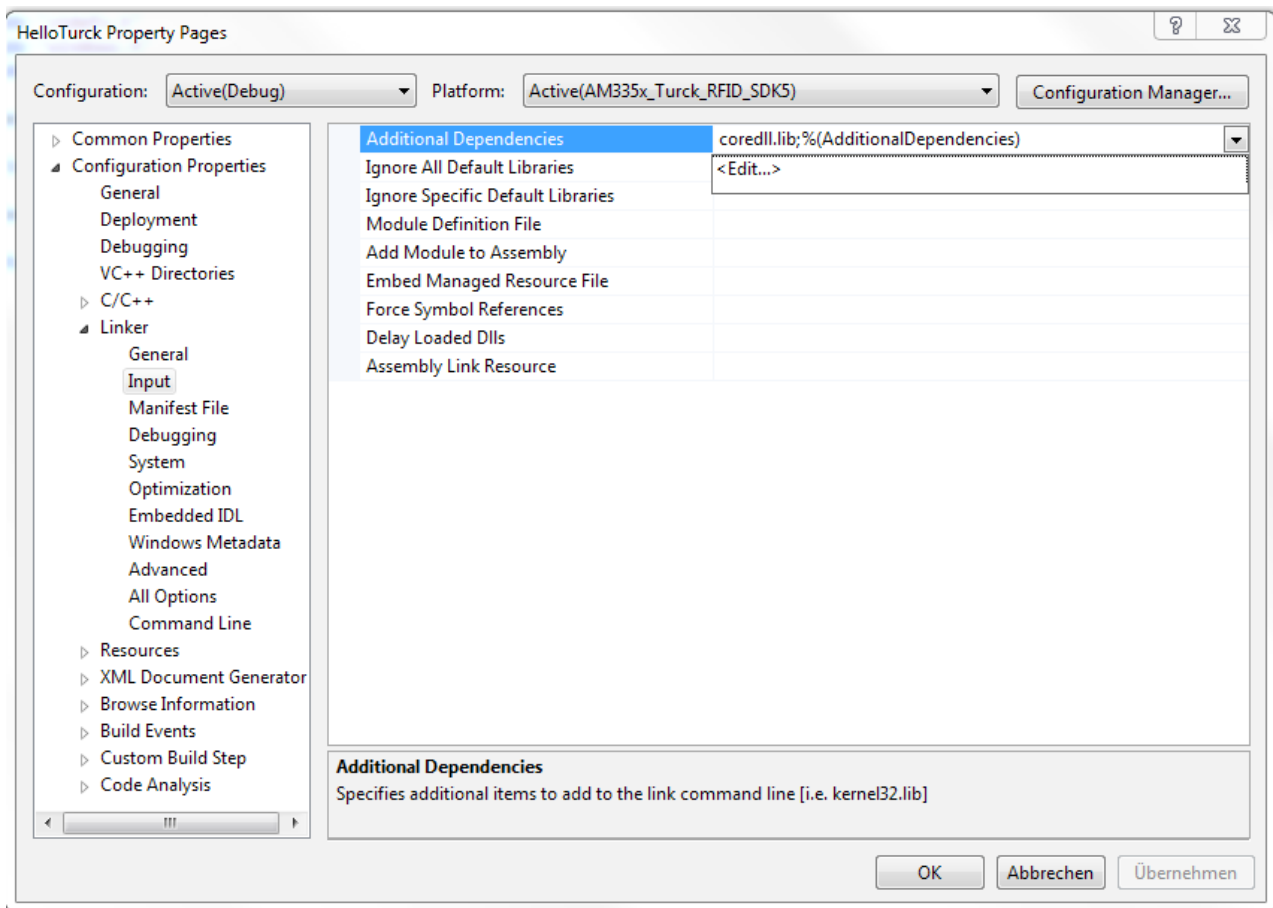


Abb. 24: Property-Seiten



► "TBOX\_API\_LIB.lib" im Editierfeld eingeben und bestätigen.

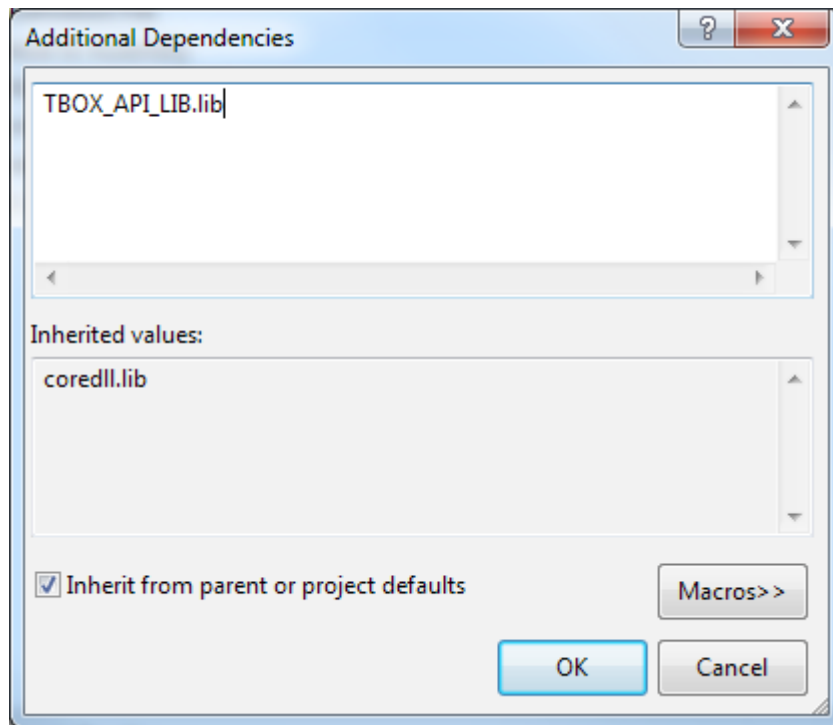


Abb. 25: Zusätzliche Abhängigkeiten

- Am Anfang der Haupt Quelldatei die Header-Datei TBOX\_API.h über "#include "TBOX\_API.h";" einbinden.
- Die Verwendung der TBOX API durch die Anwendung ist möglich. Vor Gebrauch von weiteren API-Funktionen muss "TBOX\_Init();" zuerst aufgerufen werden. Damit werden alle Einstellungen auf die Standardwerte zurückgesetzt und die Hardware in einen definierten Zustand versetzt.

**Beispiel 1: grüne LED App blinken lassen:**

```
TBOX_SYS_LED_GreenSet(LED_APP, LED_2HZ);
```

Die Einstellung LED\_TxD\_RxD\_SIGNALING wird nur für LEDs unterstützt, die einem COM-Port zugeordnet sind.

**Beispiel 2: Wert der Drehschalter einlesen:**

```
DWORD dwVal = TBOX_SYS_SWITCH_Get(0);
printf("Rotary Switches Value: %d\n", dwVal);
```

**Beispiel 3: DXP als Eingang konfigurieren und Stromwert lesen:**

```
TBOX_DXP_Init(8, FALSE, FALSE);
DWORD dwVal = TBOX_DXP_Get(8);
printf("DXP 8 Level: %d\n", dwVal2);
```

DXP0 bis DXP7 werden auf dieser Plattform nicht verwendet, weil sie von den COM-Ports belegt sind.

**Beispiel 4: Alle DXP-Eingänge sofort lesen:**

```
dwVal = TBOX_DXP_Get(-1);
for(unsigned int i=8; i < 16; i++)
    printf(„DXP %d Level: %d\n“, i, (dwVal & (1 << i)) >> i);
```

Die Werte aller DXPs werden ausgelesen, wenn "-1" als DxpNo-Parameter verwendet wird. Der Rückgabewert ist über ein Bitfeld verfügbar, dessen Bits der Nummer des DXP-Levels entsprechen (z. B. bit0 für DXP 0, bit1 für DXP1 usw.).

**Beispiel 5: Einen COM-Port mit den Einstellungen RS485-Modus, Swap-Lines, No Bias und No Termination konfigurieren. RX- und TX-Signale für die entsprechenden LEDs auch aktivieren und die Spannungsversorgung VAUX des Ports einschalten:**

```
TBOX_COM_HardwareInit(TBOX_COM1,
                      0,
                      TBOX_COM_RS485_MODE,
                      TBOX_COM_SWAP_AB_MODE,
                      TBOX_COM_BIAS_OFF_MODE,
                      TBOX_COM_TERM_OFF_MODE);

TBOX_SYS_LED_GreenSet(LED_COM1_TX, LED_TxD_RxD_SIGNALING);
TBOX_SYS_LED_GreenSet(LED_COM1_RX, LED_TxD_RxD_SIGNALING);

TBOX_SYS_VAUX_Set(TBOX_COM1, TBOX_COM_POWER_24V_MODE);
```

Die Baudeinstellung wird nach dem Einstellen ignoriert (Standardwert ist 115200 Baud). Nur der RS485-Modus wird unterstützt.

## 7.3.14 Vorgehensweise mit einer C#-Anwendung

Zur Nutzung der API in C#-Anwendungen steht die API auch als DLL (TBOX\_API\_DLL.dll) zur Verfügung. Die DLL ist nicht Teil des OS-Standard-Image und muss deshalb zum Gerät kopiert werden.

Vor Nutzung einer Funktion aus der DLL muss die DLL in Ihre C#-Anwendung importiert werden. Beispiel:

```
[DllImport(„TBOX_API_DLL.dll“)]
public extern static int TBOX_Init();

[DllImport(„TBOX_API_DLL.dll“)]
public extern static void TBOX_SYS_LED_GreenSet(int LedNo, int State);
```

Nach dem Import kann die DLL innerhalb der C#-Anwendung aufgerufen werden. Beispiel:

```
TBOX_Init();

TBOX_SYS_LED_GreenSet(4, 0xCCCC);
```

## 7.4 Spezifische Einstellungen/Implementierungen

### 7.4.1 Anwendung automatisch starten

Die Standard-Registry enthält einen Verweis auf eine bestimmte Anwendung im Flash-Speicher bezüglich des Autostarts. Somit ist es möglich, Ihre eigene Anwendung beim Hochfahren automatisch starten zu lassen, indem Sie sie in „\Mounted\_Volume\“ ablegen und „autostart.exe“ benennen.

### 7.4.2 Abbildversion auslesen

Auslesen der aktuellen WEC-Abbildversion: Fügen Sie aus dem SDK „bsp\_ioctls.h“ hinzu, und verwenden Sie den folgenden IOControl-Code mit KernelloControl:

`IOCTL_HAL_GET_BSP_VERSION`

Parameter `lpOutBuffer`:

Hinweisadresse auf eine `IOCTL_HAL_GET_BSP_VERSION_OUT` Struktur.

`IOCTL_HAL_GET_BSP_VERSION_OUT` ist wie folgt definiert:

```
typedef struct {
    DWORD dwVersionMajor;
    DWORD dwVersionMinor;
    DWORD dwVersionQFES;
    DWORD dwVersionIncremental;
} IOCTL_HAL_GET_BSP_VERSION_OUT;
```

### 7.4.3 Adressierungsmodus auslesen

Zum Auslesen des Adressierungsmodus, der bei Systemstart gemäß Taster, DIP- und Drehschalter definiert wird, kann der folgende IOControl-Code (in „bsp\_ioctls.h“ definiert) mit KernelloControl verwendet werden: `IOCTL_HAL_GET_ADDRMODE`

Parameter `lpOutBuffer`:

Hinweisadresse auf `DWORD` als Empfänger des Adressierungsmodus.

Die Ethernet-IP-Einstellungen werden automatisch beim Starten des WEC entsprechend des erkannten Modus geändert.

Die Adressierungsmodi sind im Abschnitt „Adressierungsmöglichkeiten“ auf S. 21 beschrieben.

## 8 Betreiben

### 8.1 LED-Anzeigen

Die Geräte verfügen über frei programmierbare Mehrfarben-LEDs zur Anzeige von Informationen zu:

- Versorgungsspannung
- Sammel- und Busfehler
- Status
- Diagnose

| LED PWR                                 |                   | Bedeutung                               |
|---|-------------------|---|
| aus                                     |                   | keine Spannung oder Unterspannung an V1 |
| leuchtet grün                           |                   | Spannung an V1 ok                       |
| leuchtet rot                            |                   | keine Spannung oder Unterspannung an V2 |
| LED BUS                                 |                   | Bedeutung                               |
| blinkt rot                              |                   | Wink-Kommando aktiv                     |
| COM-LEDs (RFID-Kanäle)                  |                   |   |
| LED TXD                                 | LED RXD           | Bedeutung                               |
| blinkt grün                             | aus               | Daten werden gesendet                   |
| aus                                     | blinkt grün       | Daten werden empfangen                  |
| blinkt rot                              | blinkt rot        | Kurzschluss Versorgungsspannung         |
|   | rot leuchtet      | Speicher-Überlauf                       |
| rot/grün leuchtet                       | rot/grün leuchtet | Fehlkonfiguration                       |
| DXP-LEDs (Digitale Kanäle, LEDs 8...15) |                   |   |
| LED grün                                | LED rot           | Bedeutung                               |
| aus                                     | aus               | kein I/O-Signal vorhanden               |
| leuchtet                                | aus               | I/O-Signal vorhanden                    |
| aus                                     | leuchtet          | Überlast am Ausgang                     |
| blinkt                                  | blinkt            | Überlast der Hilfsversorgung            |

## 9 Störungen beseitigen

Sollte das Gerät nicht wie erwartet funktionieren, überprüfen Sie zunächst, ob Umgebungsstörungen vorliegen. Sind keine umgebungsbedingten Störungen vorhanden, überprüfen Sie die Anschlüsse des Geräts auf Fehler.

Ist kein Fehler vorhanden, liegt eine Gerätestörung vor. In diesem Fall nehmen Sie das Gerät außer Betrieb und ersetzen Sie es durch ein neues Gerät des gleichen Typs.

## 10 Instand halten

Der ordnungsgemäße Zustand der Verbindungen und Kabel muss regelmäßig überprüft werden.

Die Geräte bei Bedarf trocken reinigen.

### 10.1 Geräte-Update durchführen

In der Standardkonfiguration umfasst das Systemabbild eine Update-Anwendung, mit der XLDR, Eboot, MAC-Konfig. und das Betriebssystem aktualisiert werden können.

**XLDR-Aktualisierung:** Verwenden Sie `update x xldrnan.1block` aus der Befehlszeile des Geräts.

**Eboot-Aktualisierung:** Verwenden Sie `update e eboot.nb0` aus der Befehlszeile des Geräts.

**Aktualisierung des Systemabbilds:** Verwenden Sie `update i NK.nb0` aus der Befehlszeile des Geräts.

**Aktualisierung der MAC-Konfiguration:** Verwenden Sie `update c bootmac_WINCE_Update_NAND_AUTOBOOT.nb0` aus der Befehlszeile des Geräts.

Um die Datei für das Update der MAC-Konfiguration zu erstellen, steht die Windows-(PC)-Anwendung `EbootCFG.exe` zur Verfügung.

Beispiel:

```
EbootCFG.exe 11:22:33:44:55:66
```

Die Anwendung gibt zwei Dateien aus:

- `Bootmac_initial.nb0`, die ausschließlich für die anfängliche Bereitstellung via externem Flashing verwendet werden soll.



#### **ACHTUNG**

Eboot nutzt beim Schreiben von `Bootmac_initial.nb0` Ethernet-Autostart **Betriebssystem startet nicht automatisch aus dem Flash**

- ▶ Datei ausschließlich für die erste Bereitstellung verwenden.

- `Bootmac_WINCE_Update_NAND_AUTOBOOT.nb0`, die für die Verwendung mit der Update-Anwendung auf dem Gerät vorgesehen ist. Enthält auch die Autostart-Einstellungen für NAND.

## 10.2 Firmware-Update durchführen

Die Firmware des Geräts lässt sich über die Tools PuTTY und WinSCP aktualisieren.



### ACHTUNG

Unterbrechung der Spannungsversorgung während des Firmware-Updates  
**Geräteschäden durch fehlerhaftes Firmware-Update**

- ▶ Spannungsversorgung des Geräts während des Firmware-Updates nicht unterbrechen.
- ▶ Während des Firmware-Updates keinen Spannungsreset durchführen.

Beispiel: Firmware aktualisieren

Im Beispiel werden die folgenden Einstellungen verwendet:

- Das Gerät ist über den Anschluss X1 an die Spannungsversorgung angeschlossen.
- Das Gerät ist über den Anschluss ETH2 an das Ethernet angeschlossen.
- Die Drehcodierschalter des Geräts befinden sich in Position 00 (Default). Die IP-Adresse des Geräts ist abhängig vom Firmware-Stand:

| Firmware-Stand                 | IP-Adresse    |
|--------------------------------|---------------|
| Version 2.1.0.0 oder niedriger | 192.168.1.1   |
| Version 2.1.1.0 oder höher     | 192.168.1.100 |

- Zwischen dem Gerät und dem Host-PC ist eine Verbindung hergestellt.
- Die Software-Tools PuTTY und WinSCP sind installiert.

- PuTTY öffnen.
- In PuTTY folgende Einstellungen eintragen:
  - Host Name: IP-Adresse des Geräts
  - Port: 23
- Optional: Namen für die aktuelle Session vergeben (hier: TBEN-Lx-WINCE). Bei späteren Wiederholungen kann die Session über „Load“ geladen werden.
- Bei gespeicherten Sessions: TBEN-Lx-WINCE auswählen und mit „Load“ bestätigen.
- „Open“ klicken.

**HINWEIS**

- Wenn keine Verbindung hergestellt werden kann, IP-Adresse des Host-PCs überprüfen.

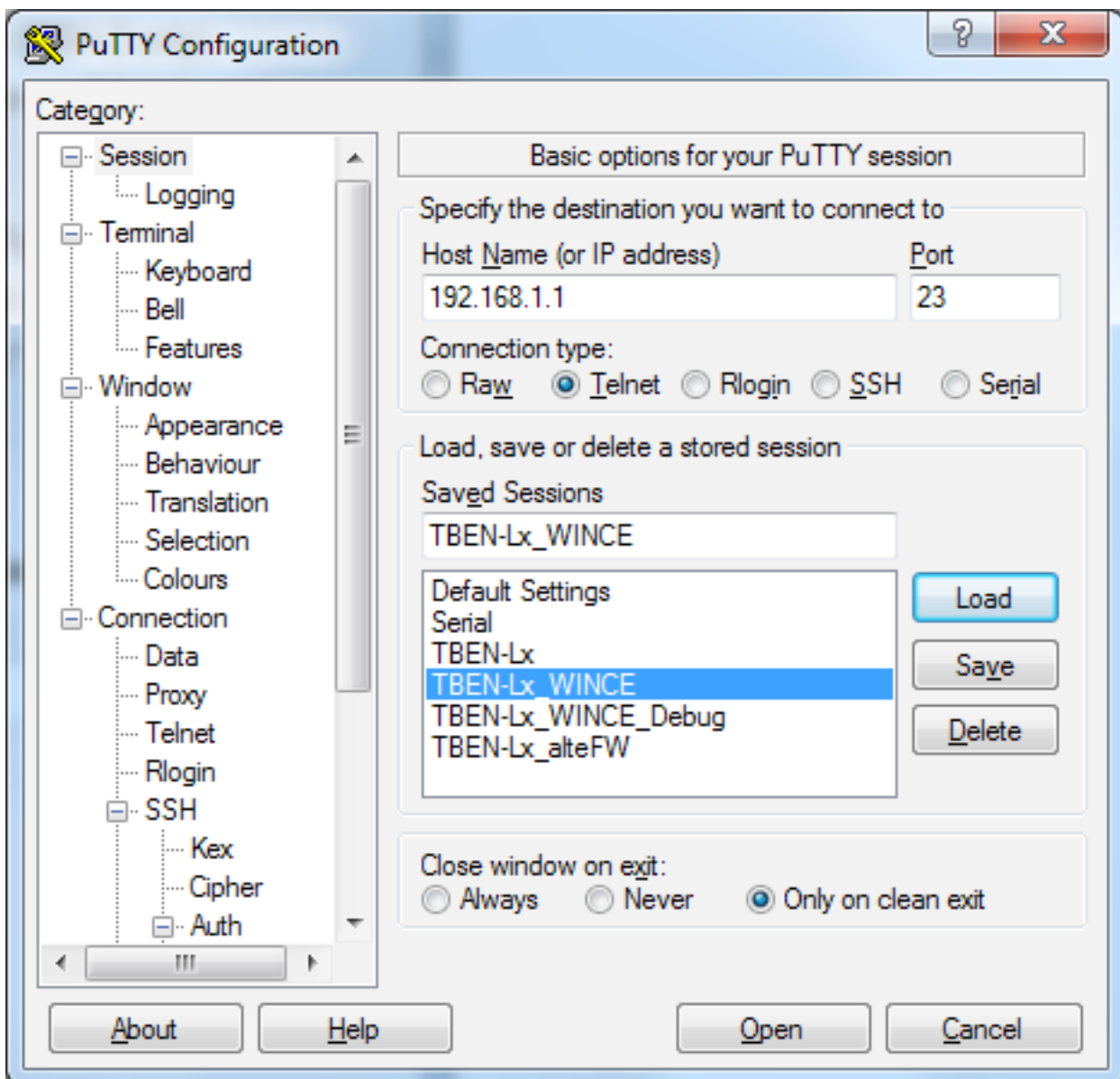


Abb. 26: PuTTY-Konfiguration

► Verzeichnis des Geräts mit `cd \windows` testweise öffnen.



**HINWEIS**

Das Verzeichnis wird evtl. erst nach mehreren Versuchen angezeigt.

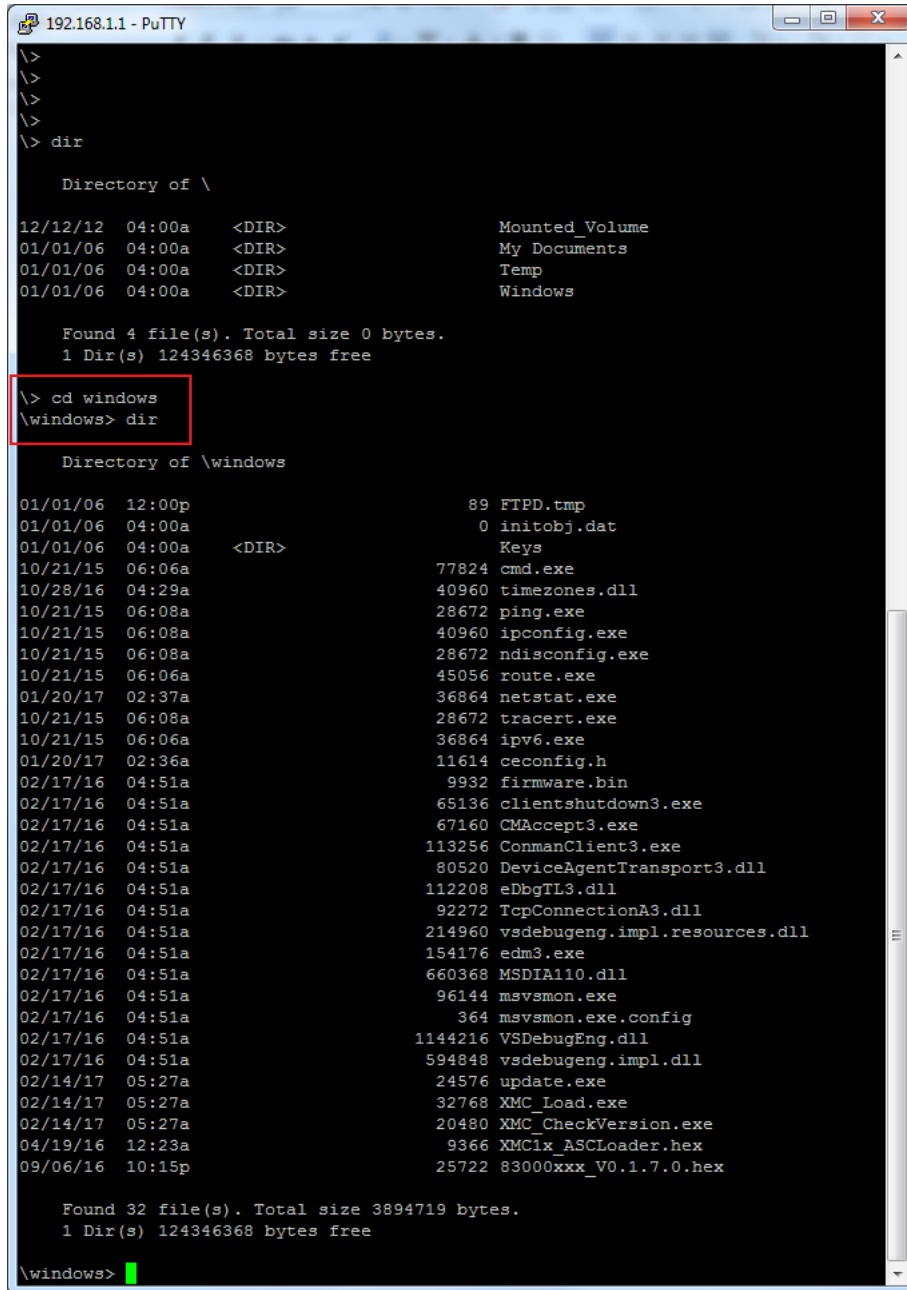


Abb. 27: Verzeichnis `\windows` in PuTTY



- Über WinSCP können Dateien zwischen dem Host-PC und dem Gerät ausgetauscht werden.
- WinSCP starten.
  - In WinSCP auf dem Gerät anmelden.
  - Checkbox „Anonymous“ aktivieren.

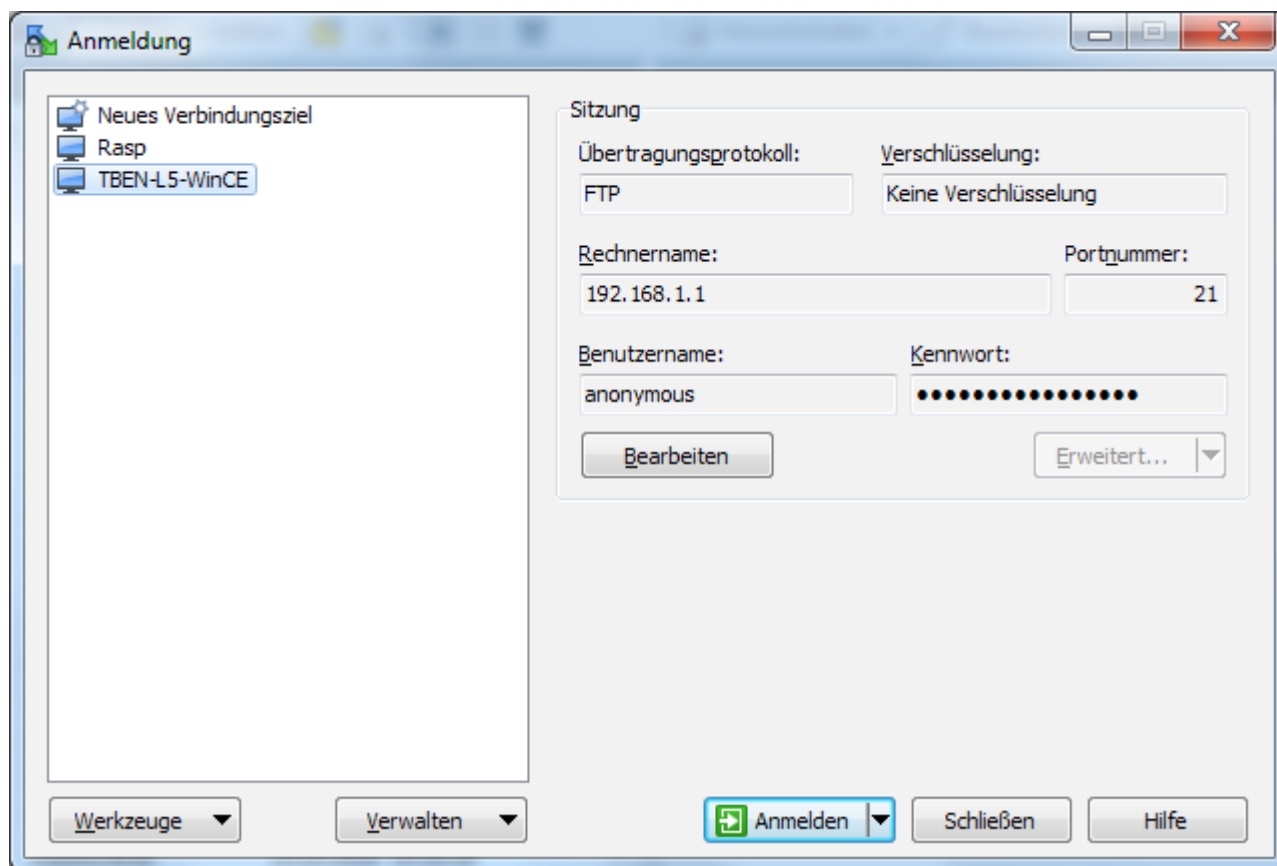


Abb. 28: Anmeldung in WinSCP

- In WinSCP die Ordner auf dem Host-PC und auf dem Gerät auswählen, zwischen denen Dateien ausgetauscht werden sollen.

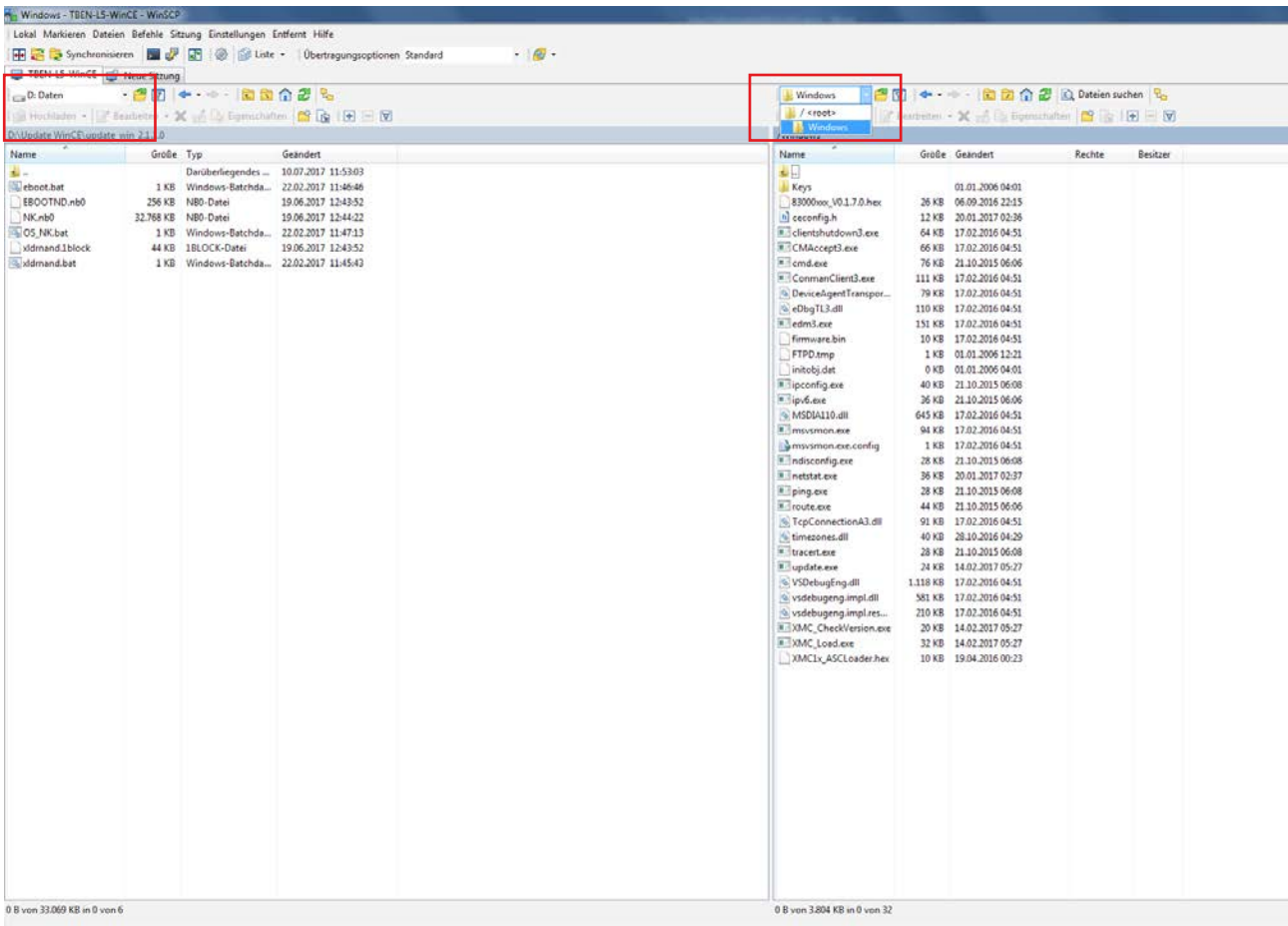


Abb. 29: WinSCP – Ordner auswählen

► „Hochladen“ klicken, um die Dateien an das Gerät zu übertragen.

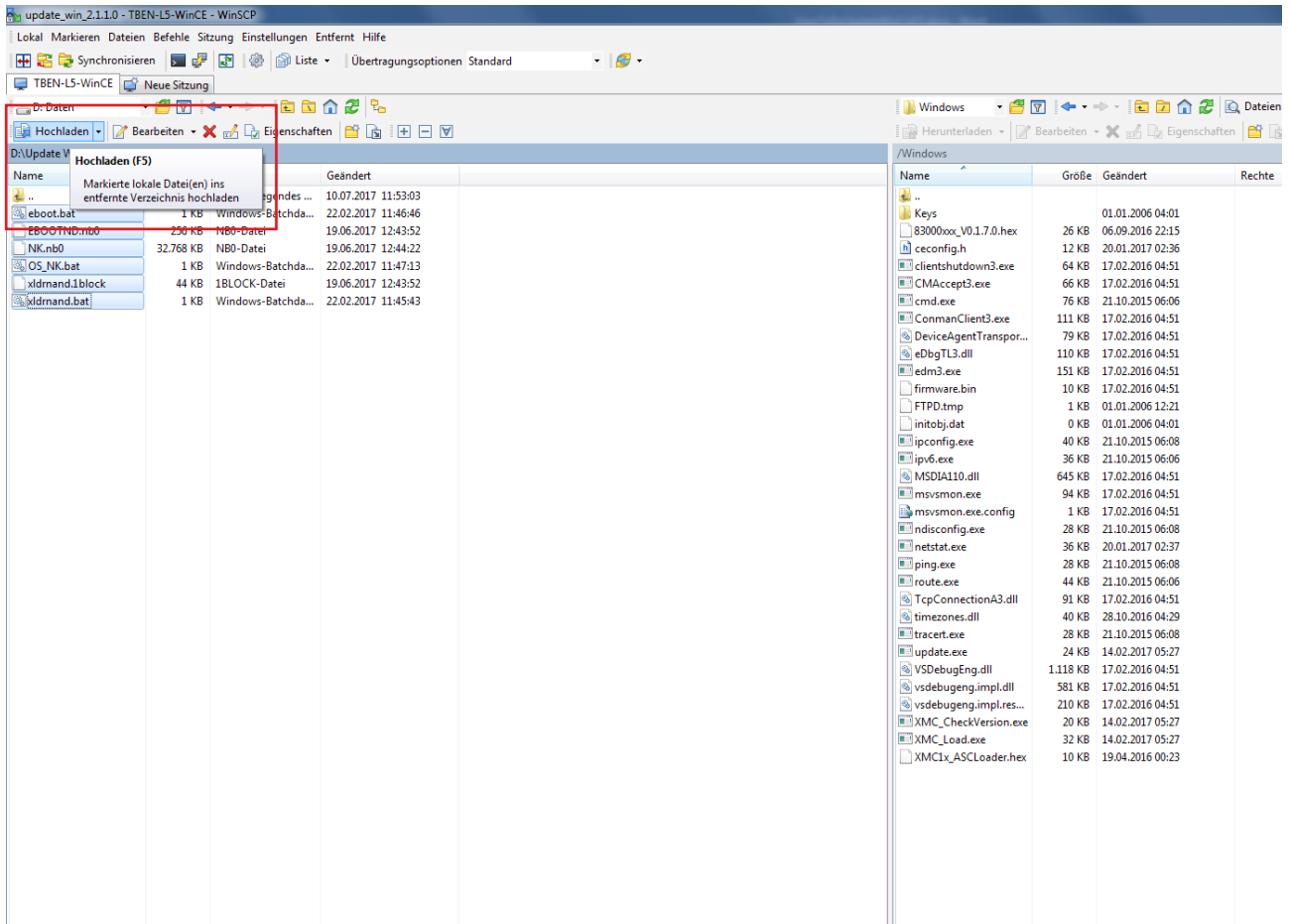


Abb. 30: Dateien an das Gerät übertragen

➤ Ansicht in PuTTY aktualisieren.

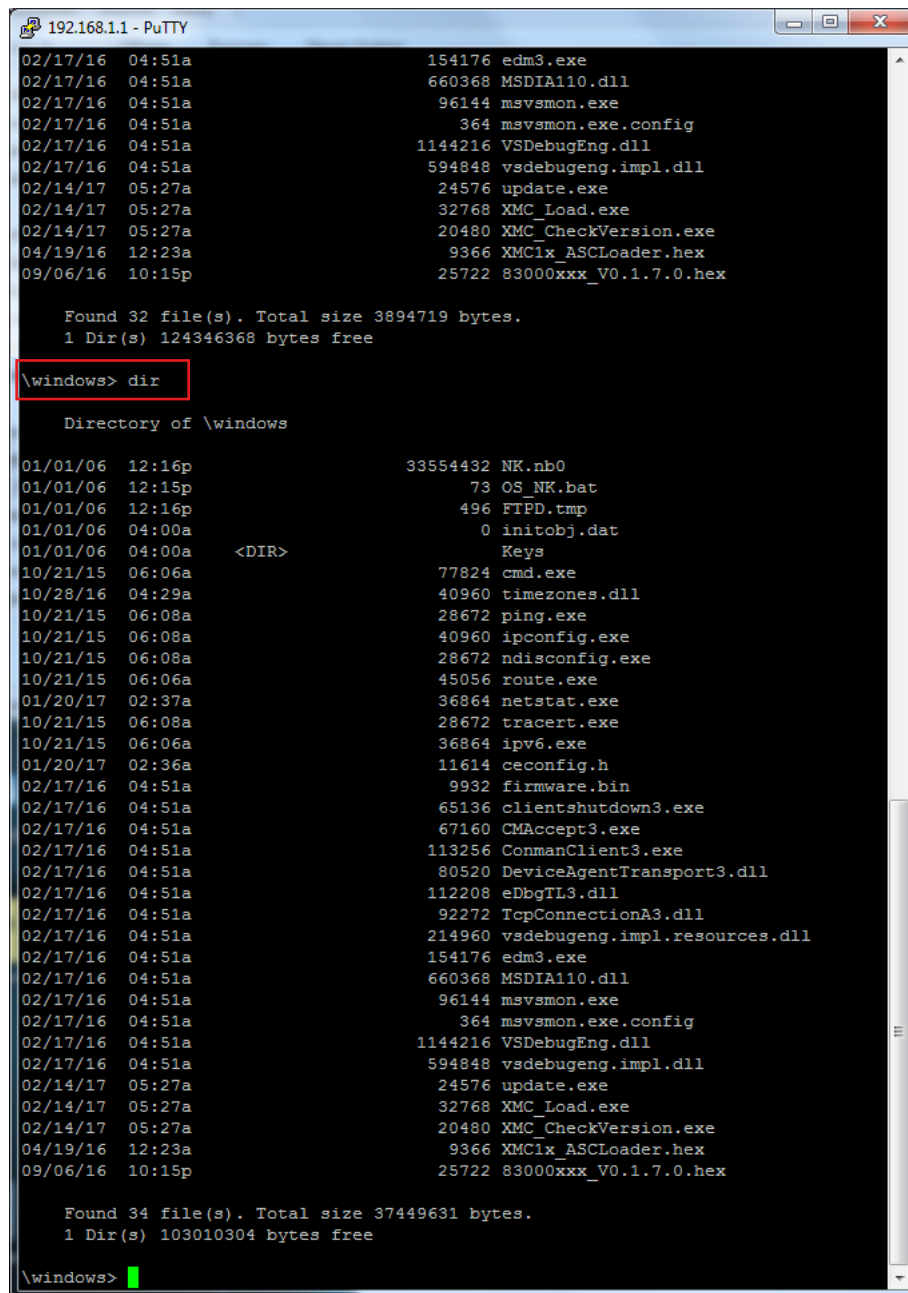


Abb. 31: PuTTY – Aktualisierte Ansicht

► Firmware-Update starten: OS\_NK.bat eingeben und mit ENTER bestätigen.

```

192.168.1.1 - PuTTY
02/17/16 04:51a          154176 edm3.exe
02/17/16 04:51a          660368 MSDIA110.dll
02/17/16 04:51a          96144  msvsmon.exe
02/17/16 04:51a             364  msvsmon.exe.config
02/17/16 04:51a          1144216 VSDebugEng.dll
02/17/16 04:51a          594848 vsdebugeng.impl.dll
02/14/17 05:27a          24576  update.exe
02/14/17 05:27a          32768  XMC_Load.exe
02/14/17 05:27a          20480  XMC_CheckVersion.exe
04/19/16 12:23a           9366  XMC1x_ASCLoader.hex
09/06/16 10:15p          25722  83000xxx_V0.1.7.0.hex

Found 32 file(s). Total size 3894719 bytes.
1 Dir(s) 124346368 bytes free

\windows> dir

Directory of \windows

01/01/06 12:16p          33554432 NK.nb0
01/01/06 12:15p             73  OS_NK.bat
01/01/06 12:16p           496  FTPD.tmp
01/01/06 04:00a             0  initobj.dat
01/01/06 04:00a          <DIR>  Keys
10/21/15 06:06a          77824  cmd.exe
10/28/16 04:29a          40960  timezones.dll
10/21/15 06:08a          28672  ping.exe
10/21/15 06:08a          40960  ipconfig.exe
10/21/15 06:08a          28672  ndisconfig.exe
10/21/15 06:06a          45056  route.exe
01/20/17 02:37a          36864  netstat.exe
10/21/15 06:08a          28672  tracert.exe
10/21/15 06:06a          36864  ipv6.exe
01/20/17 02:36a          11614  ceconfig.h
02/17/16 04:51a           9932  firmware.bin
02/17/16 04:51a          65136  clientshutdown3.exe
02/17/16 04:51a          67160  CMAccept3.exe
02/17/16 04:51a          113256  ConmanClient3.exe
02/17/16 04:51a          80520  DeviceAgentTransport3.dll
02/17/16 04:51a          112208  eDbgTL3.dll
02/17/16 04:51a          92272  TcpConnectionA3.dll
02/17/16 04:51a          214960  vsdebugeng.impl.resources.dll
02/17/16 04:51a          154176  edm3.exe
02/17/16 04:51a          660368  MSDIA110.dll
02/17/16 04:51a          96144   msvsmon.exe
02/17/16 04:51a             364   msvsmon.exe.config
02/17/16 04:51a          1144216  VSDebugEng.dll
02/17/16 04:51a          594848  vsdebugeng.impl.dll
02/14/17 05:27a          24576   update.exe
02/14/17 05:27a          32768   XMC_Load.exe
02/14/17 05:27a          20480   XMC_CheckVersion.exe
04/19/16 12:23a           9366   XMC1x_ASCLoader.hex
09/06/16 10:15p          25722   83000xxx_V0.1.7.0.hex

Found 34 file(s). Total size 37449631 bytes.
1 Dir(s) 103010304 bytes free

\windows> OS_NK.bat

```

Abb. 32: Firmware-Update starten

Das laufende Update wird mit „update NK...“ angezeigt.

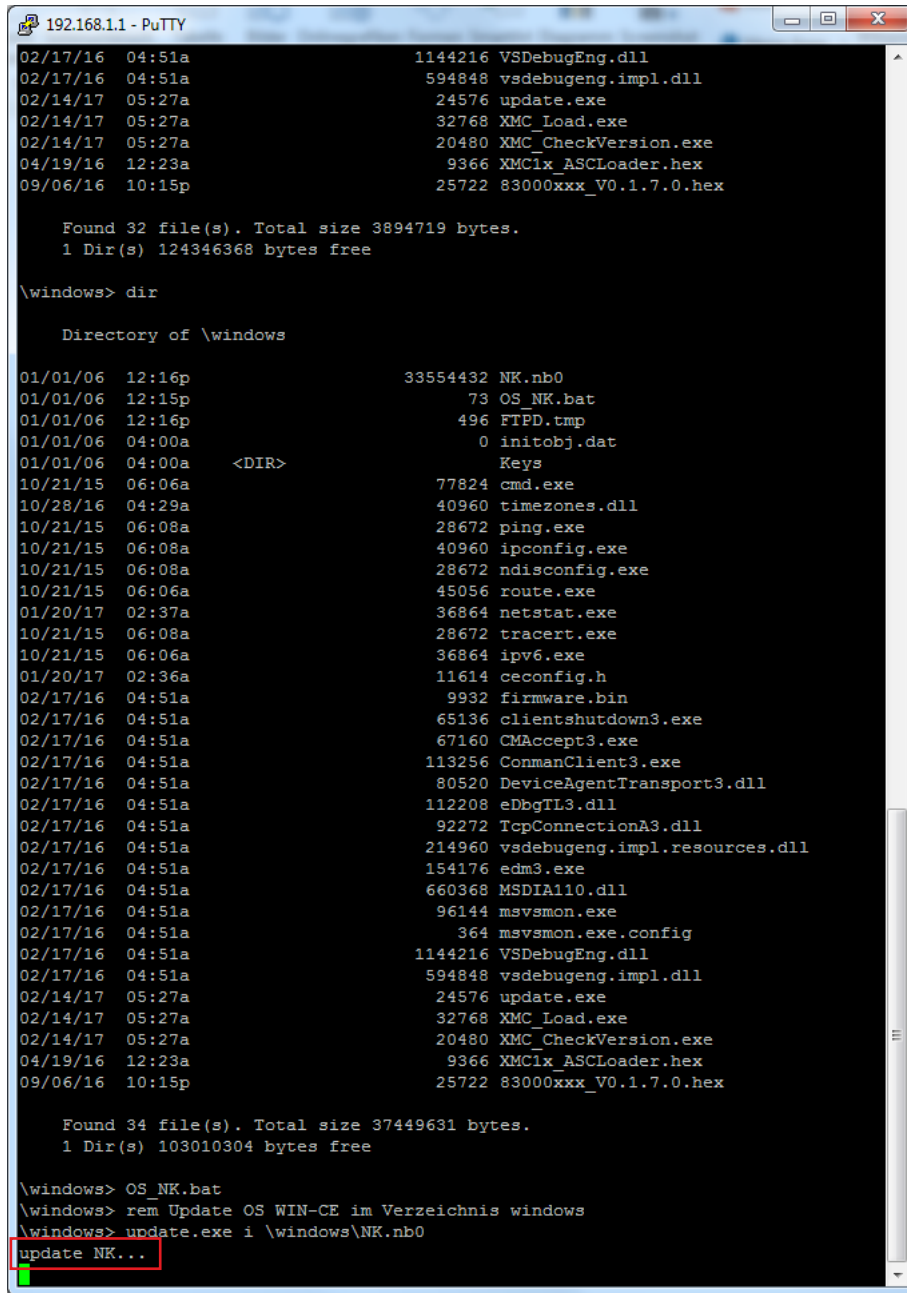


Abb. 33: Laufendes Firmware-Update

Das Firmware-Update ist abgeschlossen, wenn die Meldung „done“ erscheint.

```

192.168.1.1 - PuTTY
02/14/17 05:27a      24576 update.exe
02/14/17 05:27a      32768 XMC_Load.exe
02/14/17 05:27a      20480 XMC_CheckVersion.exe
04/19/16 12:23a       9366 XMCLx_ASCLoader.hex
09/06/16 10:15p      25722 83000xxx_V0.1.7.0.hex

Found 32 file(s). Total size 3894719 bytes.
1 Dir(s) 124346368 bytes free

\windows> dir

Directory of \windows

01/01/06 12:16p      33554432 NK.nb0
01/01/06 12:15p         73 OS_NK.bat
01/01/06 12:16p         496 FTPD.tmp
01/01/06 04:00a         0 initobj.dat
01/01/06 04:00a      <DIR>
                          Keys
10/21/15 06:06a      77824 cmd.exe
10/28/16 04:29a      40960 timezones.dll
10/21/15 06:08a      28672 ping.exe
10/21/15 06:08a      40960 ipconfig.exe
10/21/15 06:08a      28672 ndisconfig.exe
10/21/15 06:06a      45056 route.exe
01/20/17 02:37a      36864 netstat.exe
10/21/15 06:08a      28672 tracert.exe
10/21/15 06:06a      36864 ipv6.exe
01/20/17 02:36a      11614 ceconfig.h
02/17/16 04:51a         9932 firmware.bin
02/17/16 04:51a      65136 clientshutdown3.exe
02/17/16 04:51a      67160 CMAccept3.exe
02/17/16 04:51a      113256 ConmanClient3.exe
02/17/16 04:51a      80520 DeviceAgentTransport3.dll
02/17/16 04:51a      112208 eDbgTL3.dll
02/17/16 04:51a      92272 TcpConnectionA3.dll
02/17/16 04:51a      214960 vsdebugeng.impl.resources.dll
02/17/16 04:51a      154176 edm3.exe
02/17/16 04:51a      660368 MSDIA110.dll
02/17/16 04:51a      96144 msvsmon.exe
02/17/16 04:51a         364 msvsmon.exe.config
02/17/16 04:51a      1144216 VSDebugEng.dll
02/17/16 04:51a      594848 vsdebugeng.impl.dll
02/14/17 05:27a      24576 update.exe
02/14/17 05:27a      32768 XMC_Load.exe
02/14/17 05:27a      20480 XMC_CheckVersion.exe
04/19/16 12:23a       9366 XMCLx_ASCLoader.hex
09/06/16 10:15p      25722 83000xxx_V0.1.7.0.hex

Found 34 file(s). Total size 37449631 bytes.
1 Dir(s) 103010304 bytes free

\windows> OS_NK.bat
\windows> rem Update OS WIN-CE im Verzeichnis windows
\windows> update.exe i \windows\NK.nb0
update_NK...
done
\windows>
\windows>

```

Abb. 34: Firmware-Update abgeschlossen

- Firmware-Update abschließen: Spannungsreset des Geräts durchführen.

## Firmware-Stand überprüfen

- Ab Version 2.1.0.0 lässt sich der Firmware-Stand über das Turck Service Tool anzeigen.
- Turck Service Tool öffnen.
- „Suchen“ klicken.
- ➔ Das Gerät wird angezeigt. Der Firmware-Stand ist unter „Version“ aufgeführt.

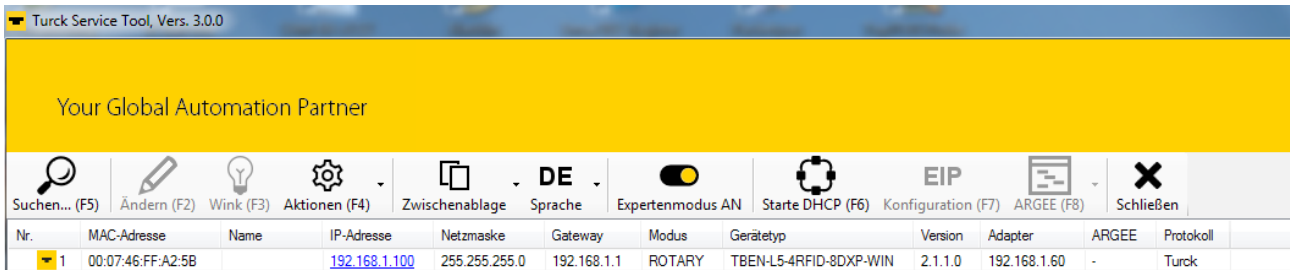


Abb. 35: TBEN-L5-4RFID-8DXP im Turck Service Tool

## 11 Reparieren

Das Gerät ist nicht zur Reparatur durch den Benutzer vorgesehen. Sollte das Gerät defekt sein, nehmen Sie es außer Betrieb. Bei Rücksendung an Turck beachten Sie bitte unsere Rücknahmebedingungen.

### 11.1 Geräte zurücksenden

Ist die Rücksendung eines Geräts erforderlich, so können nur Geräte entgegengenommen werden, die mit einer Dekontaminationserklärung versehen sind. Diese steht unter [http://www.turck.de/static/media/downloads/01\\_Dekontaminationserklaerung\\_DE.pdf](http://www.turck.de/static/media/downloads/01_Dekontaminationserklaerung_DE.pdf) zum Download zur Verfügung und muss vollständig ausgefüllt, wetter- und transportsicher an der Außenseite der Verpackung angebracht sein.

## 12 Entsorgen



Die Geräte müssen fachgerecht entsorgt werden und gehören nicht in den normalen Hausmüll.



## 13 Technische Daten

| <b>Technische Daten</b>    |   |
|----------------------------|---|
| <b>Versorgung</b>          |   |
| Versorgungsspannung        | 24 VDC  |
| Zulässiger Bereich         | 18...30 VDC   |
| Gesamtstrom                | V1 max. 8 A, V2 max. 9 A bei 70°C pro Modul         |
| RFID-Versorgung            | 2 A pro Kanal bei 70°C                              |
| Sensor-/Aktuatorversorgung | 2 A pro Steckplatz bei 70°C                         |
| Potenzialtrennung          | galvanische Trennung von V1- und V2-Spannungsgruppe |
| Spannungsfestigkeit        | bis 500 VDC V1 und V2 gegenüber Ethernet            |
| Verlustleistung            | typisch ≤ 5 W                                       |
| <b>Systembeschreibung</b>  |   |
| Prozessor                  | Cortex A8, 800 MHz                                  |
| Speicher                   | 256MB Flash ROM; 512MB DDR3 RAM                     |
| Echtzeituhr                | ja  |
| Betriebssystem             | Windows Embedded Compact 2013                       |
| <b>Systemdaten</b>         |   |
| Übertragungsrate           | Ethernet 10 Mbit/s / 100 Mbit/s                     |
| Anschlussstechnik          | 2 × M12, 4-polig, D-codiert                         |
| <b>RFID</b>                |   |
| Kanalanzahl                | 4   |
| Anschlussstechnik          | M12, 5-polig  |
| Versorgung                 | 2 A pro Kanal bei 70 °C, kurzschlussfest            |
| <b>Digitale Eingänge</b>   |   |
| Kanalanzahl                | 8   |
| Anschlussstechnik          | M12, 5-polig  |
| Eingangstyp                | PNP   |
| Art der Eingangsdiagnose   | Kanaldiagnosen                                      |
| Schaltswelle               | EN 61131-2 Typ 3, pnp                               |
| Signalspannung Low-Pegel   | < 5 V   |
| Signalspannung High-Pegel  | > 11 V  |
| Signalstrom Low-Pegel      | < 1,5 mA  |
| Signalstrom High-Pegel     | > 2 mA  |
| Potenzialtrennung          | galvanische Trennung zu P1/P2                       |
| Spannungsfestigkeit        | bis 500 VDC (V1 und V1 gegenüber Ethernet)          |
| Leitungslänge              | max. 50 m   |

| <b>Technische Daten</b>             |   |
|-------------------------------------|---|
| <b>Digitale Ausgänge</b>            |   |
| Kanalanzahl                         | 8   |
| Anschlusstechnik Ausgänge           | M12, 5-polig                                      |
| Ausgangstyp                         | PNP   |
| Art der Ausgangsdiagnose            | Kanaldiagnose                                     |
| Ausgangsspannung                    | 24 VDC aus Potenzialgruppe                        |
| Ausgangsstrom pro Kanal             | 2,0 A, kurzschlussfest, max. 4,0 A pro Steckplatz |
| Gleichzeitigkeitsfaktor             | 0,56  |
| Lastart                             | ohmsch, induktiv, Lampenlast                      |
| Kurzschlusschutz                    | ja  |
| Potenzialtrennung                   | galvanische Trennung zu P1/P2                     |
| Spannungsfestigkeit                 | bis 500 VDC (V1 und V1 gegenüber Ethernet)        |
| <b>Norm-/Richtlinienkonformität</b> |   |
| Schwingungsprüfung                  | gemäß EN 60068-2-6                                |
| Beschleunigung                      | bis 20 g  |
| Schockprüfung                       | gemäß EN 60068-2-27                               |
| Kippfallen und Umstürzen            | gemäß IEC 60068-2-31/IEC 60068-2-32               |
| Elektromagnetische Verträglichkeit  | gemäß EN 61131-2                                  |
| Zulassungen und Zertifikate         | CE  |
| UL Kond.                            | cULus LISTED 21 W2, Encl.Type 1 IND.CONT.EQ.      |
| <b>Allgemeine Information</b>       |   |
| Abmessungen (B×L×H)                 | 60.4 × 230.4 × 39 mm                              |
| Betriebstemperatur                  | -40...+70 °C                                      |
| Lagertemperatur                     | -40...+70 °C                                      |
| Einsatzhöhe                         | max. 5000 m                                       |
| Schutzart                           | IP65/IP67/IP69K                                   |
| MTTF                                |   |
| Gehäusematerial                     | PA6-GF30  |
| Gehäusefarbe                        | schwarz   |
| Material Fenster                    | Lexan   |
| Material Schraube                   | 303 Edelstahl                                     |
| Halogenfrei                         | ja  |
| Montage                             | 2 Befestigungslöcher, Ø 6,3 mm                    |



# TURCK

...with 28 subsidiaries and over  
60 representations worldwide!

D500061 | 2018/01



[www.turck.com](http://www.turck.com)