

SIEMENS

SIMATIC

PROFINET Leitfaden Evaluation Kit ERTEC 200P PN IO V4.0

Programmier- und Bedienhandbuch




Vorwort

Einleitung	1
Lieferumfang	2
Übersicht des Leitfadens	3
PROFINET IO-Device einrichten	4
Automatisierungssystem einrichten	5
Erste Schritte mit PROFINET IO-Device-Kommunikation	6
Applikationsbeispiele	7
Eigene Devices erstellen	8
Anhang	A

Rechtliche Hinweise

Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 GEFAHR
bedeutet, dass Tod oder schwere Körperverletzung eintreten wird , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
 WARNUNG
bedeutet, dass Tod oder schwere Körperverletzung eintreten kann , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
 VORSICHT
bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
ACHTUNG
bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.


Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 WARNUNG
Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Vorwort

Vorwort

Zweck des Leitfadens

Der Leitfaden beschreibt den schnellen Einstieg in die Handhabung des Evaluation Board ERTEC 200P mit dem Betriebssystem eCos.

Zielgruppe des Leitfadens

Dieser Leitfaden ist für Software-Entwickler gedacht, die den ERTEC 200P zusammen mit dem Betriebssystem eCos für neue Produkte einsetzen wollen. Folgende Grundkenntnisse sind dafür erforderlich:

- Programmiererfahrung in C/C++
- Programmier Techniken wie z. B. Multi-Threading und Callback-Routinen
- PROFINET IO-Systemkenntnisse
- Allgemeine Kenntnisse auf dem Gebiet der Automatisierungstechnik
- Basiskenntnisse für die Projektiersoftware STEP 7

Dem Entwickler wird ein Evaluation Kit, bestehend aus einem Evaluation Board ERTEC 200P und einem PROFINET IO-Softwarepaket zur Verfügung gestellt, mit dem er seine PROFINET IO-Device-Applikation mit dem Betriebssystem eCos testen kann.

Aufbau des Leitfadens

Der Leitfaden beschreibt das Evaluation Board ERTEC 200P und ist wie folgt aufgebaut:

- Kapitel 1 Einleitung
- Kapitel 2 Lieferumfang
- Kapitel 3 Übersicht des Leitfadens
- Kapitel 4 PROFINET IO-Device einrichten
- Kapitel 5 Automatisierungssystem einrichten
- Kapitel 6 Erste Schritte mit PROFINET IO-Device-Kommunikation
- Kapitel 7 Applikationsbeispiele
- Kapitel 8 Eigene Devices erstellen
- Anhang: Abkürzungen/Begriffsverzeichnis, Literaturverzeichnis

Dieser Leitfaden enthält die Beschreibung des PROFINET IO-Stacks für das Evaluation Board EB 200P, der bei Bedarf aktualisiert wird. Den jeweils aktuellen Stand finden Sie im Internet (<http://www.siemens.de/comdec>).

Wegweiser

Um Ihnen den schnellen Zugriff auf spezielle Informationen zu erleichtern, enthält das Handbuch folgende Zugriffshilfen:

- Am Anfang des Handbuchs finden Sie ein vollständiges Inhaltsverzeichnis und jeweils eine Liste der im Handbuch enthaltenen Abbildungen und Tabellen.
- Im Anhang finden Sie ein Abkürzungs- und Begriffsverzeichnis, in welchem wichtige Fachbegriffe definiert sind, die in diesem Handbuch verwendet werden.
- Hinweise auf weitere Dokumente sind mit Hilfe von Literaturnummern in Schrägstrichen ("Nr./") angegeben. Damit können Sie dem Literaturverzeichnis im Anhang des Handbuchs den genauen Titel der Dokumente entnehmen.

Konventionen

Beachten Sie auch die folgendermaßen gekennzeichneten Hinweise:

Hinweis

Ein Hinweis enthält wichtige Informationen zum beschriebenen Produkt, zur Handhabung des Produkts oder zu dem Teil der Dokumentation, auf den besonders aufmerksam gemacht wird.

Weitere Unterstützung

Bei Fragen zur Nutzung des beschriebenen Bausteines, die Sie nicht in der Dokumentation beantwortet finden, wenden Sie sich bitte an Ihre Siemens Ansprechpartner in den für Sie zuständigen Vertretungen oder Geschäftsstellen.

Fragen, Anmerkungen und Verbesserungen zum vorliegenden Handbuch schicken Sie bitte schriftlich an die angegebene E-Mail-Adresse.

Zusätzlich erhalten Sie allgemeine Informationen, aktuelle Produkt-Informationen, FAQs und Downloads, die beim Einsatz nützlich sein können, im Internet (<http://www.siemens.de/comdec>).

Technischer Ansprechpartner für Deutschland/weltweit

Siemens AG
ComDeC
(<http://www.siemens.de/comdec>)

Hausadresse:
Würzburger Str. 121
D-90766 Fürth

Tel.: +49 911 750 2080
Tel.: +49 911 750 4384
Tel.: +49 911 750 2078
Fax: +49 911 750 2100
E-Mail: (<mailto:ComDeC@siemens.com>)

Briefadresse:
Postfach 2355
D-90713 Fürth

Technischer Ansprechpartner für USA

PROFI Interface Center
(<http://www.profiinterfacecenter.com>)
One Internet Plaza
Johnson City, TN 37604

Tel: +1 (423) 262- 2576
Fax: +1 (678) 297- 7289
E-Mail: (<mailto:PIC.industry@siemens.com>)

Inhaltsverzeichnis

	Vorwort	3
1	Einleitung	11
2	Lieferumfang	13
	2.1 Hardware-Komponenten.....	13
	2.2 Software-Komponenten.....	13
	2.3 Inhalt und Zielgruppe der Anwenderbeschreibung.....	13
	2.4 Sonstige Hinweise.....	14
	2.5 Notwendige Tools und Komponenten.....	14
3	Übersicht des Leitfadens	15
4	PROFINET IO-Device einrichten	17
	4.1 Betriebssystem eCos und PROFINET IO-Stack.....	17
	4.1.1 Installationshinweise.....	17
	4.1.2 Installation durchführen.....	18
	4.2 eCos mit Entwicklungsumgebung Eclipse.....	20
	4.2.1 Entwicklungsplattform Eclipse installieren.....	20
	4.2.2 Entwicklungsplattform Eclipse einrichten.....	21
	4.2.3 Eclipse: eCos-Kernel generieren.....	23
	4.2.4 Eclipse: Build-Konfiguration einstellen.....	25
	4.2.5 Eclipse: Build der PROFINET IO-Applikation durchführen.....	26
	4.2.6 Applikationsbeispiel bearbeiten.....	29
	4.3 Konfiguration des Terminalprogrammes für Meldungsangaben.....	30
	4.4 EB 200P in Betrieb nehmen.....	31
	4.4.1 Übersicht Evaluation Board EB 200P.....	31
	4.4.2 Verbindungen durchführen.....	32
	4.4.3 MAC-Adresse einstellen.....	35
5	Automatisierungssystem einrichten	37
	5.1 Automatisierungssystem einrichten.....	37
	5.2 Projektbeispiel für PROFINET IO-Device bearbeiten.....	37
	5.3 PROFINET IO-Projekt mit STEP 7 bearbeiten und in CPU 317 laden.....	40
6	Erste Schritte mit PROFINET IO-Device-Kommunikation	45
	6.1 PROFINET IO-Device-Kommunikation starten.....	45
	6.2 PROFINET IO-Device-Kommunikation testen.....	47

7	Applikationsbeispiele	49
7.1	Beispiele für Device-Applikationen	50
7.1.1	Device-Applikationsbeispiel für das Standard-Interface (SI)	52
7.1.2	Device-Applikationsbeispiel für das Direct Buffer Access Interface (DBAI)	53
7.1.3	Device-Applikationsbeispiel für eine taktsynchrone Applikation mit IRT	54
7.2	Funktionalitäten aus Anwendersicht (Use Cases)	55
7.2.1	Use Case RT	55
7.2.2	Use Case IRT (ohne taktsynchrone Applikation)	55
7.2.3	Use Case IRT mit taktsynchroner Applikation	56
7.2.4	Use Case Tool Calling Interface (TCI)	57
7.2.5	Use Case PROFInergy	58
7.2.6	Use Case Medienredundanz (MRP)	60
7.2.7	Use Case Shared Device	61
7.3	GSD-Dateien	62
7.4	Controller-Applikationsbeispiele	63
7.4.1	Controller-Applikationsbeispiele für SIMATIC S7-300 CPUs	63
7.5	Laden einer Applikationsfirmware in den Flash	63
8	Eigene Devices erstellen	65
A	Anhang	67
A.1	Abkürzungen/Begriffsverzeichnis	67
A.2	Literaturverzeichnis	69

Tabellen

Tabelle 7- 1 Controller-Beispiele mit Device-Applikation 63

Bilder

Bild 3-1 Übersicht: Notwendige Schritte..... 16

Bild 4-1 Lizenzvereinbarungen für Installation von Betriebssystem eCos 18

Bild 4-2 Installationsfenster der Evaluation Kit-Software für Betriebssystem eCos..... 18

Bild 4-3 Installationsfenster zu Software Requirements 19

Bild 4-4 Installationsverzeichnis eCos..... 19

Bild 4-5 Eclipse einrichten 21

Bild 4-6 Entwicklungsplattform Eclipse mit Perspektive "C/C++"..... 22

Bild 4-7 Build-Konfiguration einstellen 25

Bild 4-8 Projekt-Build durchführen 26

Bild 4-9 Fortschrittsbalken während des Build-Vorganges 27

Bild 4-10 Projektordner mit Bin-Datei und ladbarer Datei 28

Bild 4-11 Serial Port Setup Parameter für Terminal..... 30

Bild 4-12 Steckerübersicht für Evaluation Board EB 200P 31

Bild 4-13 Verbindung des JTAG Debugger mit dem Evaluation Board 33

Bild 4-14 Aufbau PROFINET IO-Controller/Device für EK-ERTEC 200P mit einem beliebigen
PROFINET IO-Controller 34

Bild 4-15 MAC-Adresse für Evaluation Board 35

Bild 5-1 RT-Projekt öffnen und bearbeiten..... 38

Bild 5-2 Hardware-Katalog aktualisieren..... 39

Bild 5-3 Projektieren eines PROFINET Device mit EB 200P in HW-KONFIG 40

Bild 5-4 Alle Ethernet Teilnehmer überprüfen 42

Bild 5-5 Alle Teilnehmer im Netz anzeigen 43

Bild 5-6 Device-Name und IP-Parameter zuweisen..... 44

Bild 7-1 Konfigurationsdatei "usriod_app.h" 50

Bild 7-2 Dateien des Standard-Interface-Beispiels (SI) 52

Bild 7-3 Dateien des DBAI-Beispiels..... 53

Bild 7-4 Dateien des IsoApp-Beispiels 54

Bild 7-5 Applikationsdatei für PROFInergy-Beispiel..... 59

PROFINET ist ein Automatisierungskonzept für die Realisierung modularer, dezentraler Applikationen. Mit PROFINET erstellen Sie Automatisierungslösungen, wie sie Ihnen von PROFIBUS her bekannt und vertraut sind. Die Umsetzung von PROFINET wird einerseits durch den PROFINET-Standard für Automatisierungsgeräte und andererseits durch das Engineeringtool (STEP 7, TIA-Portal) realisiert. Das bedeutet, dass Sie im Engineering nahezu die gleiche Applikationssicht haben – unabhängig davon, ob Sie PROFINET-Geräte oder PROFIBUS-Geräte projektieren. Die Programmierung Ihres Anwenderprogramms ist damit für PROFINET und PROFIBUS nahezu identisch.

Für PROFINET stellt die Siemens AG Entwicklungspakete auf Basis ERTEC 200P zur Verfügung, mit denen eine schnelle und kostengünstige Realisierung eines PROFINET IO-Devices durchgeführt werden kann. Das Entwicklungspaket entlastet den Anwender bei der Entwicklung seiner Hard- und Software.

Für die Entwicklung einer PROFINET IO-Device Hardware sind folgende Schaltpläne auf CD abgelegt:

- Kompletter Schaltplan des EB 200P
- Minimalkonfiguration für ein PROFINET IO-Device

Für die Entwicklung einer PROFINET IO-Device Software ist ein PROFINET IO-Stack mit Anwenderbeispiel auf CD abgelegt, der folgende Funktionalität beinhaltet:

- zyklischer Datenaustausch RT und IRT mit einem PROFINET IO-Controller
- Senden und Empfangen von Diagnose- und Prozessalarmen, Plug- und Pull-Alarmen
- Behandlung I&M0...4-Daten
- Vergabe von IP-Adressen und Gerätenamen über PROFINET
- Fast Startup-Funktionalität
- Shared Device
- Medienredundanz (MRP)

Um den Firmware-Stack implementieren zu können, werden gute PROFINET IO-Kenntnisse vorausgesetzt.

Lieferumfang

2.1 Hardware-Komponenten

Die Hardware besteht aus folgenden Komponenten:

- EB 200P
 - Evaluation Board zur Realisierung eines PROFINET IO-Devices

2.2 Software-Komponenten

Die Software besteht aus einer CD mit folgendem Inhalt:

- EK-ERTEC 200P
 - PROFINET IO-Protokoll-Stack und Applikationsbeispiel zur Realisierung eines PROFINET IO-Devices mit Betriebssystem eCos
 - eCos V3.0 mit Board Support Package für EB 200P
 - GSDML V2.3-Beispieldatei zur Integration in STEP 7 HW-Konfig
 - Dokumentation zum Evaluation Kit
 - Eclipse 3.4.1 inklusive GDB Hardware Debugger Support, z. B. für die Nutzung eines OpenOCD JTAG Debuggers.

PROFINET IO-Stack, Applikationsbeispiel und eCos liegen als C-Quellcode für ein PROFINET IO-Device vor.

2.3 Inhalt und Zielgruppe der Anwenderbeschreibung

Die vorliegende Dokumentation ist gedacht für Entwickler von PROFINET IO-Devices. Sie beinhaltet:

- Übersicht über den Aufbau des EK-ERTEC 200P
- Beschreibung und Konfiguration der notwendigen Tools
- Beschreibung und Generierung des Anwenderbeispiels
- Beschreibung eines ablauffähigen PROFINET-Gesamtsystems

Diese Dokumentation beinhaltet **keine**:

- Übersicht über PROFINET
- Beschreibung des PROFINET-Protokolls
- detaillierte Beschreibung über Aufbau und Abläufe im PROFINET IO-Stack

2.4 Sonstige Hinweise

Das mitgelieferte Applikationsbeispiel wurde auf einem Evaluation Board EB 200P getestet.

2.5 Notwendige Tools und Komponenten

Notwendige Tools und Komponenten:

- "Wireshark Network Protocol Analyzer" zur Aufzeichnung von Ethernet-Datenpaketen
- USB-Kabel Mini-B zur Anschaltung eines Terminalprogramms auf dem PC (z. B. TeraTerm, RS232-USB-Konverter ist auf dem EB 200P integriert)
- USB-RS232-Treiber, enthalten auf der CD im Verzeichnis "\Tools\USB to RS232 driver"
- Ethernet-Kabel
- JTAG-Debugger: Dokumentation und Script-Dateien für einen preiswerten OpenOCD-basierten JTAG-Debugger finden Sie nach Installation der Evaluation Kit Produkt-CD auf Ihrem PC im Unterverzeichnis "...\\PNIODevkit\\OpenOcdDebug" oder auf der ComDeC-Webseite (<http://www.siemens.de/comdec>).

PROFINET IO-Controller und Engineering Tool

- Ein PROFINET IO-Controller ist nicht im Evaluation Kit enthalten, er muss gesondert bezogen werden. Dies könnte z. B. eine SIMATIC S7 CPU 317-2 PN/DP V3.2 oder höher sein. Ein passendes Anwenderbeispiel dazu ist auf der Produkt-CD enthalten und kann bei Bedarf an andere S7-CPU's angepasst werden.

Übersicht des Leitfadens

Die nachfolgende Abbildung zeigt alle Schritte, die notwendig sind:

- Hard- und Software installieren
- Terminal einrichten (z. B. TeraTerm) und USB-RS232-Treiber von CD installieren
- EB 200P in Betrieb nehmen und ggf. voreingestellte MAC-Adresse ändern
- PROFINET IO-Controller und passendes Engineering System in Betrieb nehmen (nicht Bestandteil des Evaluation Kit)
- GSD-Datei in Engineering System importieren
- Projektierungsbeispiel erstellen und in den PROFINET-Controller laden
- Eine lauffähige PROFINET IO-Device-Applikation erstellen und auf das Evaluation Board laden (Hinweis: Ein lauffähiges Applikationsbeispiel ist bereits im Auslieferungszustand auf dem EB 200P geladen und wird automatisch nach Herstellung der Versorgungsspannung am EB gestartet).

- Das PROFINET IO-Device mit einem PROFINET IO-Controller verbinden und testen
- Eigene Device-Applikation definieren und Anpassungen bei GSD-Datei, Projektierung und Device-Applikation durchführen

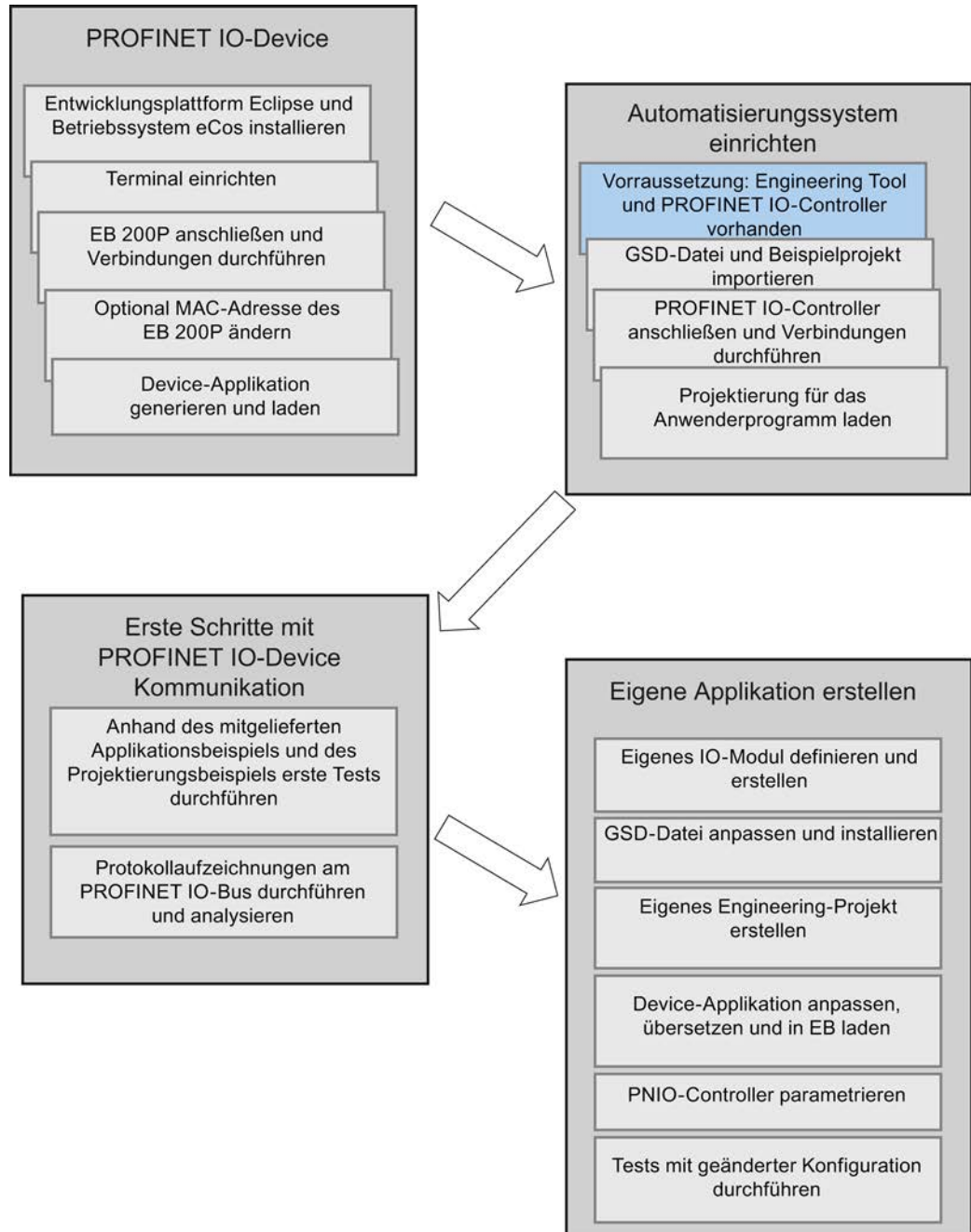


Bild 3-1 Übersicht: Notwendige Schritte

PROFINET IO-Device einrichten

Zur Erstellung eines PROFINET IO-Devices muss zuerst das Betriebssystem eCos zusammen mit dem PROFINET IO-Devicestack und der Device-Applikation auf einem Windows-PC installiert werden.

Wir empfehlen, die Entwicklungsplattform Eclipse zu verwenden, die in diesem Leitfaden auch beschrieben ist. Sollten Sie eine andere Entwicklungsplattform verwenden, so müssen Sie die notwendigen Anpassungen selbst durchführen.

Hinweis zur Installation

Hinweis

Alle Einstellungs- und Installationshinweise in den nachfolgenden Kapiteln beziehen sich auf das Evaluation Board EB 200P.

4.1 Betriebssystem eCos und PROFINET IO-Stack

4.1.1 Installationshinweise

Das Betriebssystem eCos ist eine lizenzgebührenfreie Open Source Software.

Hinweis

eCos steht u. a. unter der GNU General Public License V2 (oder höher), mit einer speziellen Ausnahme bzgl. Copyleft Effekt:

"As a special exception, if other files instantiate templates or use macros or inline functions from this file, or you compile this file and link it with other works to produce a work based on this file, this file does not by itself cause the resulting work to be covered by the GNU General Public License. However the source code for this file must still be made available in accordance with section (3) of the GNU General Public License.

This exception does not invalidate any other reasons why a work based on this file might be covered by the GNU General Public License."

Bei Nutzung und Weitergabe von eCos sind die Bedingungen dieser und weiterer Lizenzbedingungen zu erfüllen. In der Datei "Readme_OSS" auf dem EvalKit-Datenträger finden Sie im Kapitel zu eCos eine Zusammenstellung von Lizenzbedingungen, die sich in eCos finden.

4.1.2 Installation durchführen

1. Legen Sie die Evaluation Kit-CD in das CD/DVD-Laufwerk Ihres Windows PCs ein.
2. Starten Sie die Installation durch Aufruf der Batch-Datei "Install.bat" auf dem Root-Verzeichnis der CD.
Folgende Lizenzvereinbarungen werden vor der Installation angezeigt:

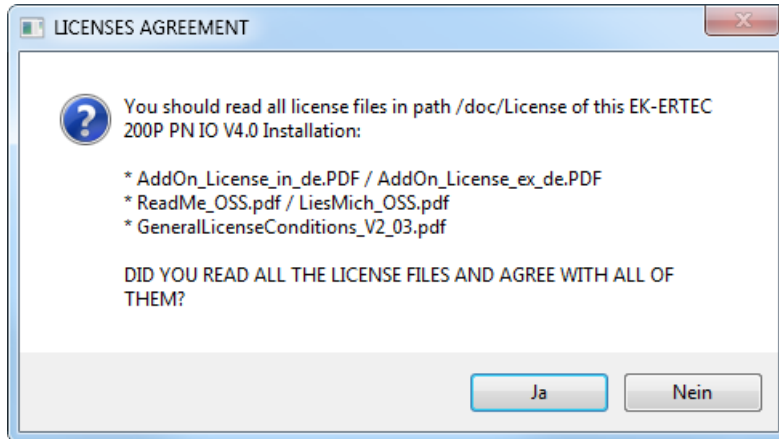


Bild 4-1 Lizenzvereinbarungen für Installation von Betriebssystem eCos

3. Klicken Sie auf **Ja**, wenn Sie die Lizenzvereinbarungen akzeptieren. Die Installation wird gestartet.
(Klicken Sie auf **Nein**, wenn Sie die Lizenzvereinbarungen nicht akzeptieren. Die Installation wird abgebrochen.)

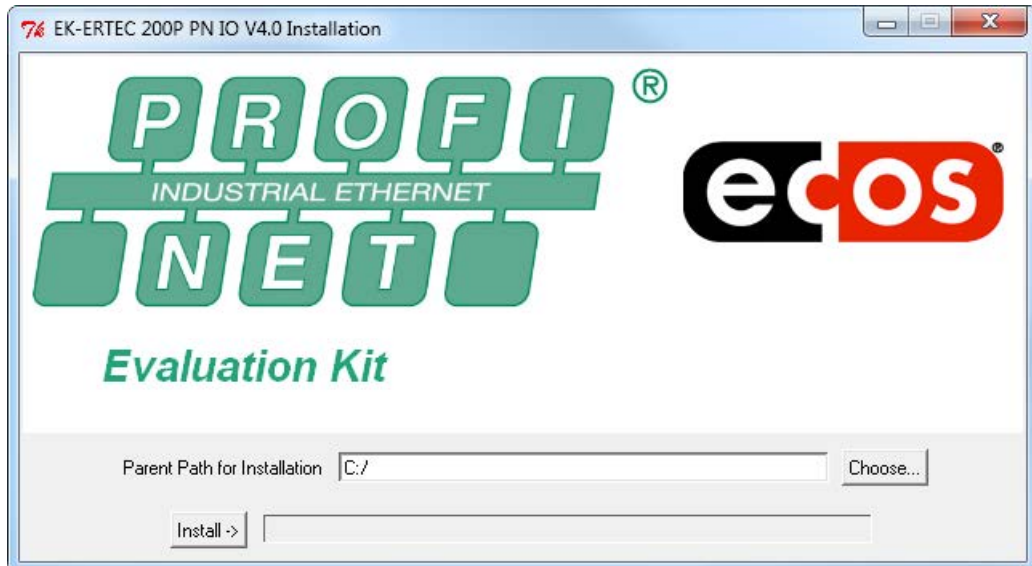


Bild 4-2 Installationsfenster der Evaluation Kit-Software für Betriebssystem eCos

4. Wählen Sie mit der Schaltfläche **Choose** das Laufwerk aus, auf dem die Installation durchgeführt werden soll. Als **Default** ist Laufwerk **C:/** eingestellt.

5. Starten Sie die Installation mit der Schaltfläche **Install**.
Es erscheint folgende Meldung am Bildschirm:

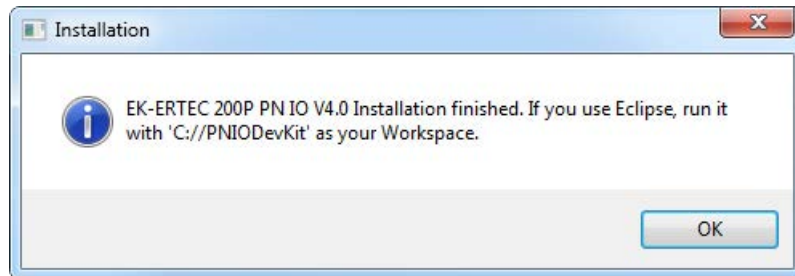


Bild 4-3 Installationsfenster zu Software Requirements

6. Bestätigen Sie mit **OK**, um die Installation fortzusetzen. Der Installationsfortschritt wird mit dem Fortschrittsbalken angezeigt.
7. Gegen Ende der Installation können Sie wählen, ob auch Eclipse von der CD installiert werden soll. Wenn Sie dies mit der Schaltfläche **Ja** bestätigen, wird eine selbstentpackende Zip-Datei "eclipseInstall.exe" von der CD gestartet, andernfalls wird die Eclipse-Installation übersprungen.
Wählen Sie im WinZip Self-Extractor ein beliebiges Verzeichnis aus, in das Eclipse installiert werden soll und klicken Sie auf "Extrahieren", um die Installation fortzusetzen.
Nach Beendigung der Installation verlassen Sie den WinZip Self-Extractor mittels der Schaltfläche **Schließen**.

Ergebnis: Die Installation erfolgt, wie im nachfolgenden Bild dargestellt, in dem ausgewählten Pfad "LW:/PNIODevKit".

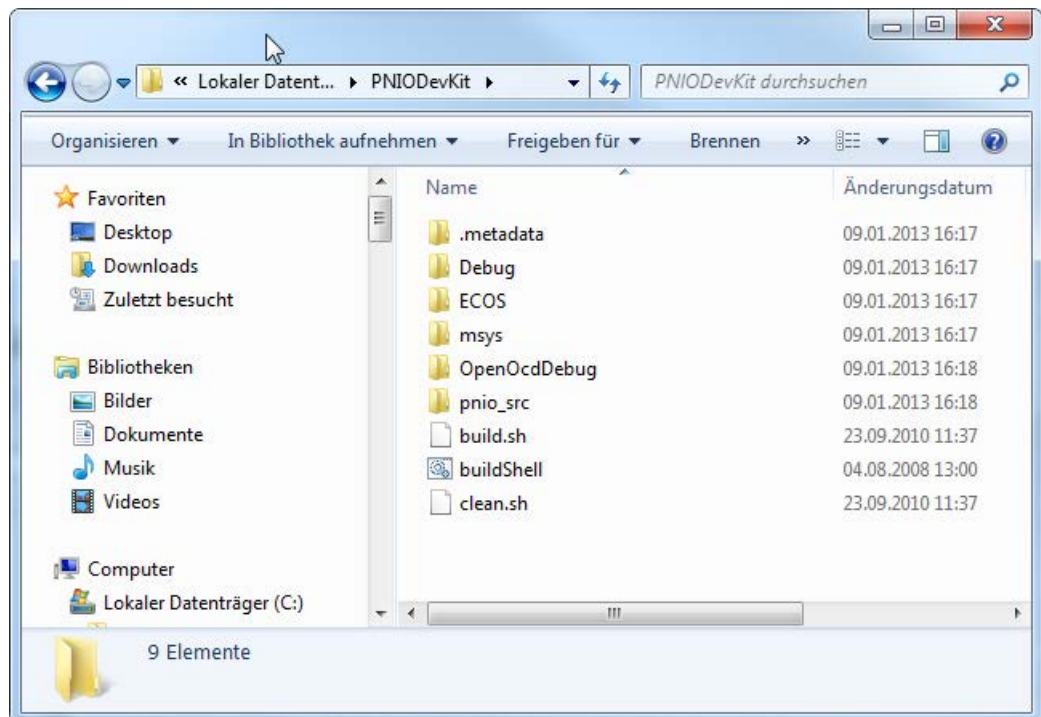


Bild 4-4 Installationsverzeichnis eCos

4.2 eCos mit Entwicklungsumgebung Eclipse

4.2.1 Entwicklungsplattform Eclipse installieren

Wir empfehlen, für die Entwicklung eines eigenen PROFINET IO-Device mit dem Betriebssystem eCos die Entwicklungsumgebung Eclipse zu verwenden.

Eine Eclipse Version 3.4.1 mit GDB Hardware-Debugger Support (notwendig für die Verwendung eines OpenOCD JTAG-Debuggers) ist auf der Produkt-CD im Verzeichnis "(...)\tools\eclipse" vorhanden.

Eclipse kann bereits beim Installieren des Evalkit von der CD mitinstalliert werden, das Eclipse-Installationsverzeichnis kann frei gewählt werden.

Da Eclipse keine Einträge in der Windows Registry vornimmt, kann das Eclipse-Installationsverzeichnis nachträglich beliebig kopiert oder verschoben werden.

Eclipse ist eine lizenzgebührenfreie Open Source Software, die alternativ auch von der Webseite (<http://www.eclipse.org/downloads>) heruntergeladen werden kann. Für Eclipse gelten gesonderte Open Source-Lizenzbedingungen.

Hinweis

Nutzer von Windows 7

Im Zusammenspiel mit bestimmten Virenscannern kann es zu deutlich längeren Generierzeiten kommen. Es wird empfohlen, in solchen Fällen durch entsprechende Exclude-Anweisungen im Virenscanner die PROFINET-Generierverzeichnisse von der Überprüfung auszuschließen.

Hinweise zu Referenzplattformen, auf denen Eclipse getestet wurde, finden Sie nach der Installation von Eclipse in den **Eclipse Release Notes** in Datei "(...)\eclipse\readme\readme_eclipse.html".

4.2.2 Entwicklungsplattform Eclipse einrichten

Eclipse einrichten

Hinweis

Voraussetzung für den Start von Eclipse

Damit Sie Eclipse IDE auf Ihrem Windows PC starten können, muss Java Runtime Environment JRE1.5 oder höher installiert sein. Falls die Software auf Ihrem Rechner noch nicht installiert ist, laden Sie diese von der Webseite (<http://www.java.com>) herunter und führen die Installation durch. Die Installationsanweisung dazu finden Sie auf der gleichen Webseite.

1. Wechseln Sie in Ihr "Eclipse"-Verzeichnis und starten Sie "eclipse.exe".
Eclipse startet mit folgendem Fenster:

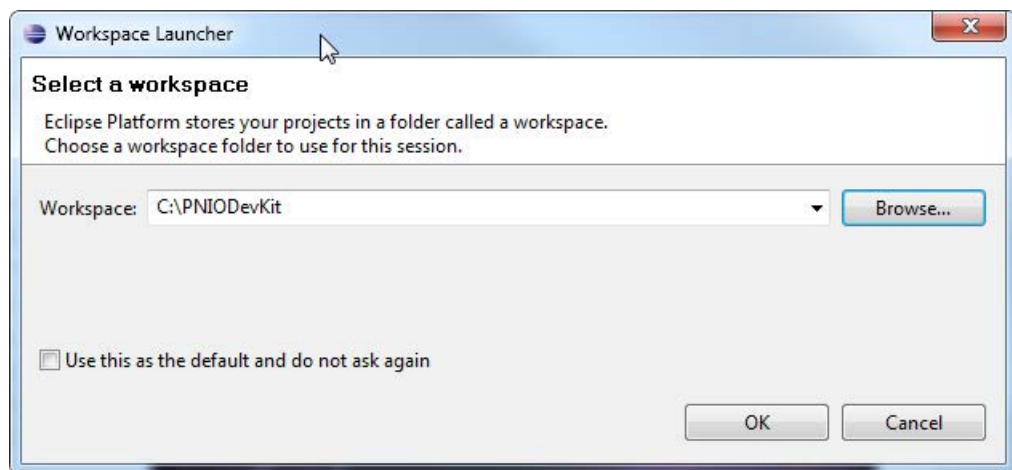


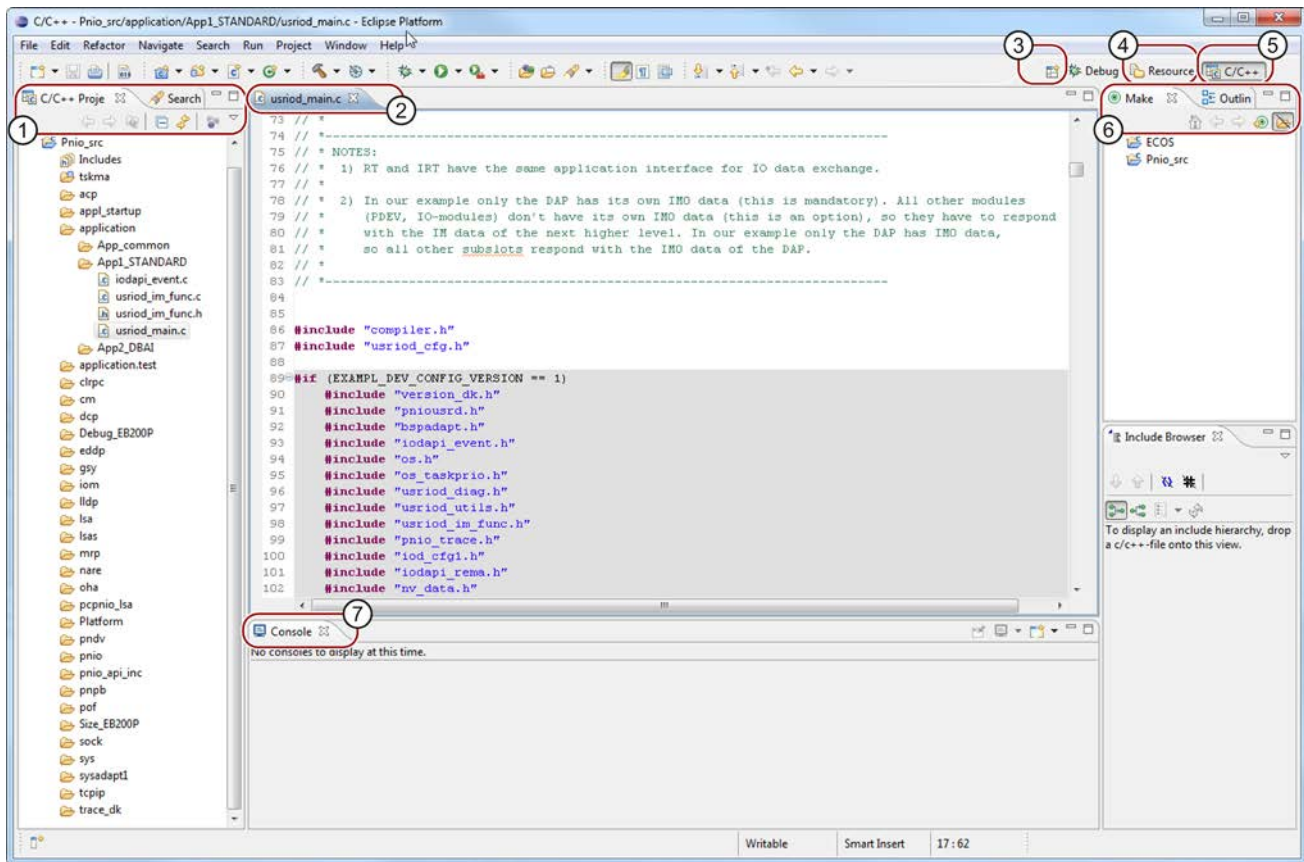
Bild 4-5 Eclipse einrichten

2. Geben Sie das Arbeitsverzeichnis (z. B. "LW:\PNIODevKit"), in dem eCos und der PROFINET IO-Stack installiert wurden, in der Zeile "Workspace" an und bestätigen Sie mit **OK**.

Hinweis

Dieses Fenster wird bei jedem Start von Eclipse geöffnet. Wenn Sie immer mit den gleichen Einstellungen starten wollen, aktivieren Sie das Kontrollkästchen "Use this as the default ...". Danach wird das Fenster nicht mehr angezeigt.

Nachdem der Workspace eingerichtet wurde, startet Eclipse mit folgenden Dialogfenstern:



- ① Projekt Explorer mit den Projekt-Workspaces C/C++ Projects, Include Browser
- ② C/C++ Editor-Fenster mit allen geöffneten C- und H-Dateien, z. B. usriod_main.c
- ③ Perspektive öffnen
- ④ Perspektive: Resource
- ⑤ Perspektive: C/C++ (Default)
- ⑥ Dialog-Fenster. Default sind folgende Ansichten eingestellt: Make Targets, Outline
- ⑦ Dialog-Fenster. Default ist folgende Ansicht eingestellt: Console

Bild 4-6 Entwicklungsplattform Eclipse mit Perspektive "C/C++"

Sie können die Standardeinstellung und -perspektiven jederzeit ändern oder Ansichten und Perspektiven hinzufügen.

Ansichten hinzufügen	Wählen Sie den Menübefehl Window <Show View <Others . Selektieren Sie die gewünschte Ansicht und bestätigen Sie mit OK . Die Ansicht erscheint nun in einem der Dialogfenster. Nun können Sie die Ansicht, wie unten beschrieben, verschieben.
Ansichten verschieben	Selektieren Sie die Ansicht, die Sie verschieben möchten, und ziehen Sie diese in das gewünschte Dialogfenster.
Perspektiven hinzufügen	Klicken Sie auf das Symbol Perspektive öffnen und selektieren Sie die Perspektive, die Sie hinzufügen möchten. Die ausgewählte Perspektive wird danach rechts oben als Symbol angezeigt ("Perspektiven").

Weitere Schritte

Nach der Neuinstallation von eCos und Eclipse sind **einmalig** folgende Schritte durchzuführen:

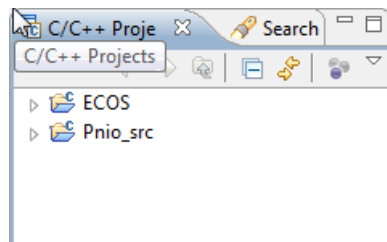
- eCos-Kernel für EB 200P generieren (Seite 23)
- Build-Konfiguration einstellen (Seite 25)
- Build-Vorgang für den PROFINET IO-Stack EB 200P durchführen (Seite 26)

4.2.3 Eclipse: eCos-Kernel generieren

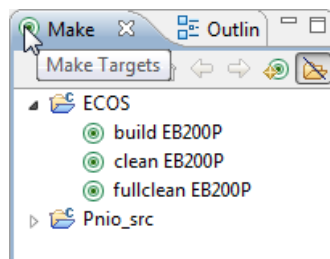
eCos-Kernel für EB 200P generieren

1. Wählen Sie dazu das Dialogfenster "Make Targets" aus.
2. Rufen Sie "build EB200P" auf (siehe Bild unten), entweder mit Doppelklick auf "build EB200P" oder rechte Maustaste auf "build EB200P" und aus dem Kontextmenü **Build Make Target** aufrufen.

Ergebnis: Der eCos-Kernel und die Libraries für EB 200P werden generiert.



Ansicht "C/C++ Projects"



Ansicht "Make Targets"

Mit "clean EB200P" werden die Object-Verzeichnisse im ausgewählten Kernel gelöscht.

Mit "fullclean EB200P" werden die Build- und Include-Verzeichnisse im ausgewählten Kernel gelöscht.

Hinweis

Ein erneutes Building des eCos-Kernel ist nur notwendig, wenn:

- Änderungen im eCos Kernel durchgeführt werden
 - "clean EB200P" aufgerufen wurde
 - "fullclean EB200P" aufgerufen wurde
-

4.2.4 Eclipse: Build-Konfiguration einstellen

Build-Konfiguration einstellen

1. Selektieren Sie den Projektfolder "Pnio_src" und drücken Sie die rechte Maustaste.
2. Wählen Sie eine Build-Konfiguration aus (z. B. **Build Configurations > Set Active > Size EB200P**)

Ergebnis: Die Build-Konfiguration für alle nachfolgenden Aktionen ist nun eingestellt. Die aktive Build-Konfiguration wird mit einem Haken angezeigt (siehe folgendes Bild).

Hinweis

Folgende Build-Konfigurationen sind möglich:

- Debug_EB200P: "debug" für EB 200P
- Size_EB200P: "optimized Size" für EB 200P

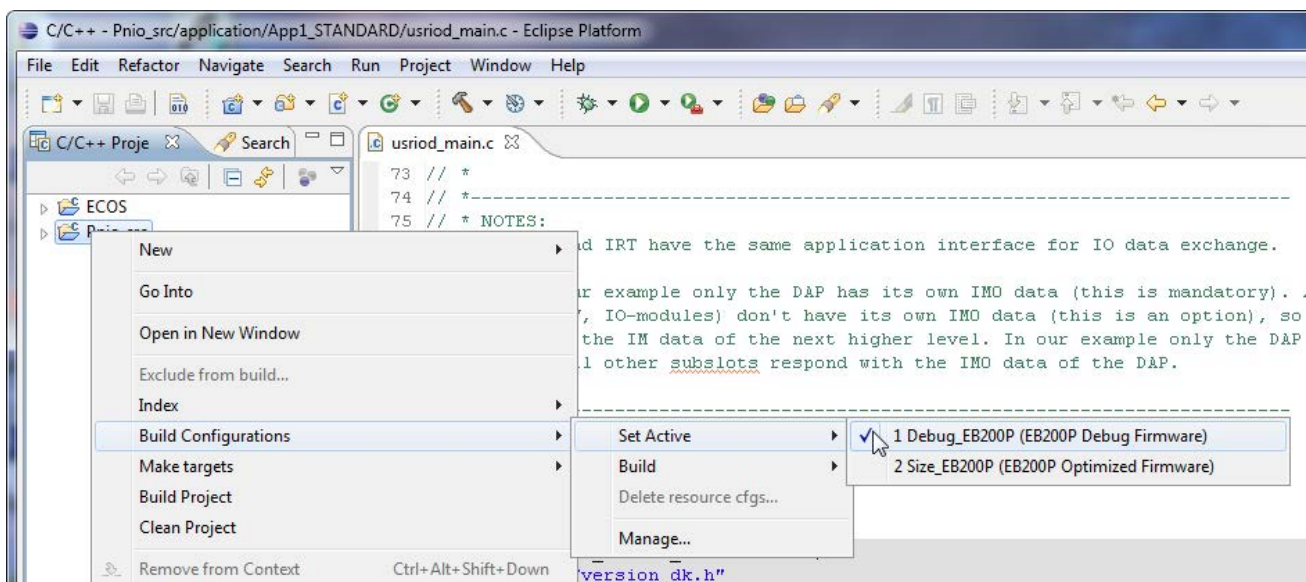


Bild 4-7 Build-Konfiguration einstellen

4.2.5 Eclipse: Build der PROFINET IO-Applikation durchführen

Build-Vorgang für den PROFINET IO-Stack inklusive der Applikation für EB 200P durchführen

1. Drücken Sie die rechte Maustaste im Projektfolder "Pnio_src".
2. Wählen Sie das Kommando **Build Project** ①

Ergebnis: Der Build-Vorgang wird gestartet und ein Projektordner für die entsprechende Build-Konfiguration erzeugt.

- Während des Build-Vorganges wird der Build-Fortschritt im Fenster "Progress" ③ und Ausgaben im Fenster "Console" ② angezeigt.

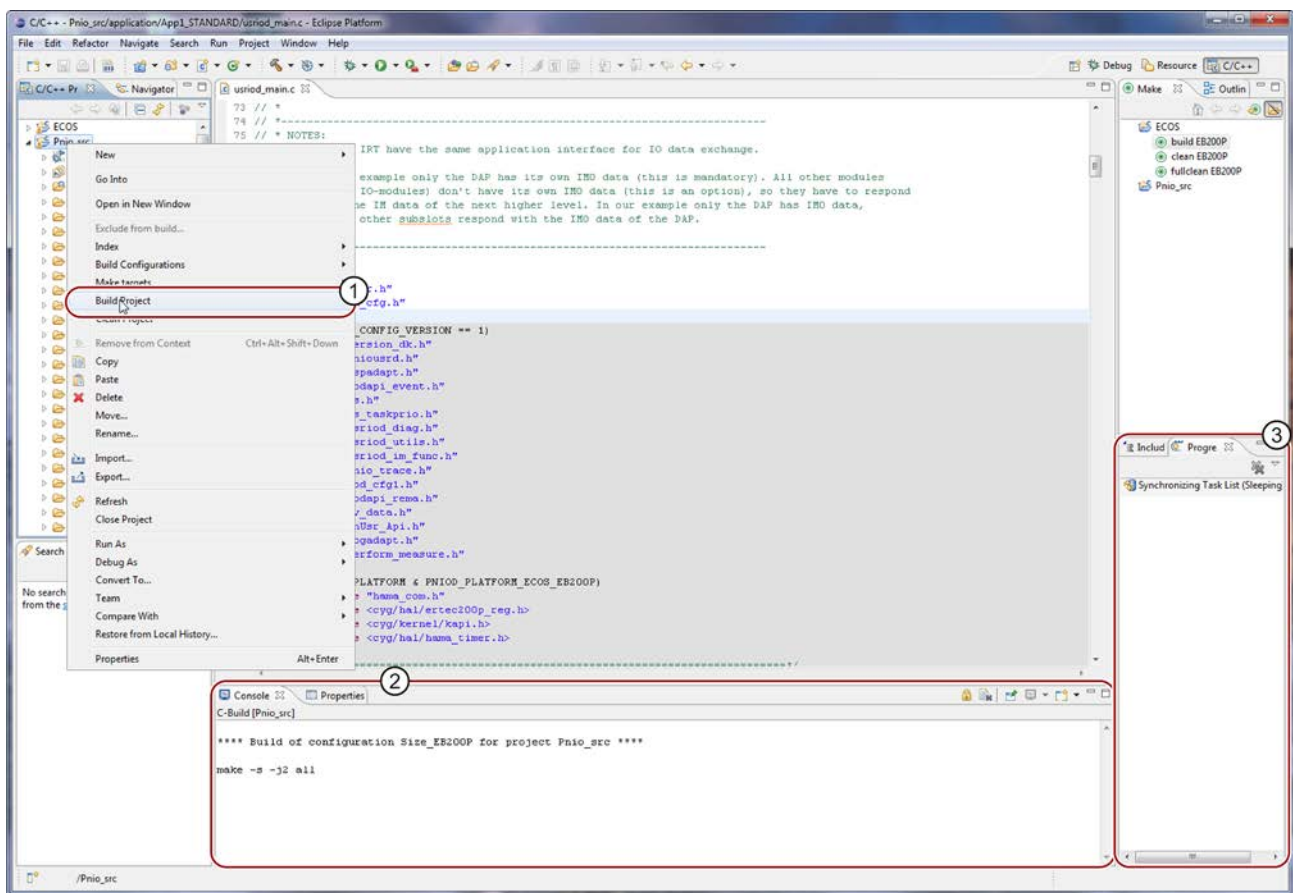


Bild 4-8 Projekt-Build durchführen

- Der Build-Vorgang ist beendet, wenn der Fortschrittsbalken, der während des Build-Vorganges angezeigt wird, geschlossen wird (siehe Screenshot).

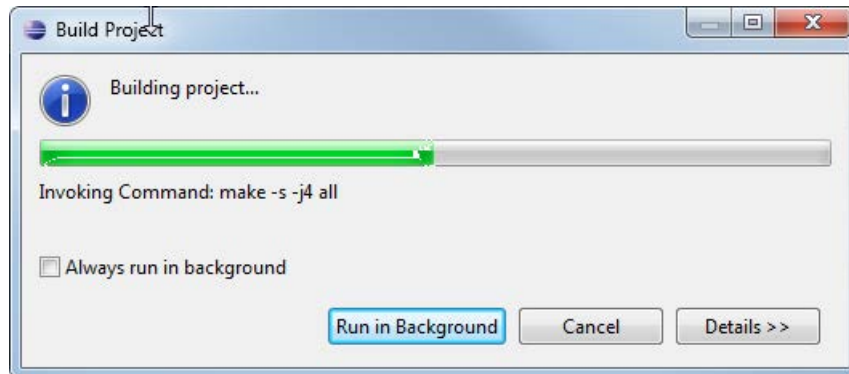


Bild 4-9 Fortschrittsbalken während des Build-Vorganges

- Nach erfolgreichem Build-Vorgang wird im Projektordner "Pnio_src" ein Ordner, wie in der Build-Konfiguration ausgewählt, angelegt. In unserem Beispiel ist dies der Ordner "Size_EB200P" (siehe folgendes Bild).

Der Ordner enthält neben den Objekt-Ordern die beiden erzeugten ablauffähigen Dateien:

- PNIO4ECOS: ablauffähiges PROFINET IO-Beispiel (ELF-Datei) für Test mit einem JTAG-Debugger (Start der Firmware aus dem RAM)
- ecos.bin: binäre Imagedatei mit ablauffähigem PROFINET IO-Beispiel für Flashprogrammierung (BIN-Datei)

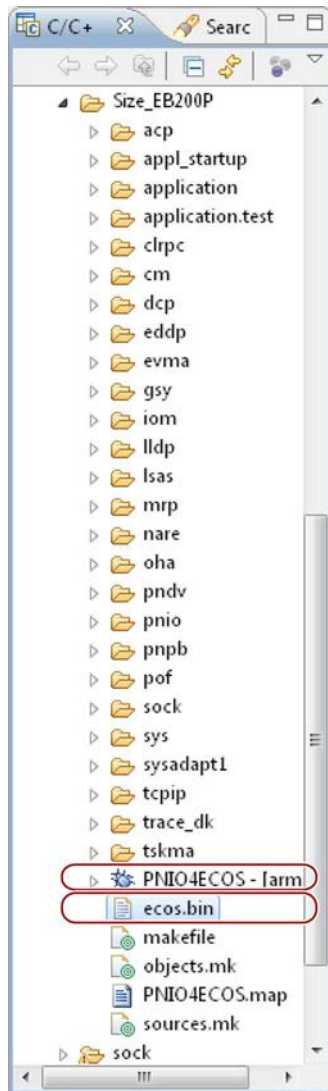


Bild 4-10 Projektordner mit Bin-Datei und ladbarer Datei

4.2.6 Applikationsbeispiel bearbeiten

Im Ordner "Pnio_src" ist der Ordner "application" mit verschiedenen Beispielen für ein PROFINET IO-Device enthalten. Folgende PROFINET IO-Beispiele können ausgewählt werden:

Beispiel	Wert	Applikationsdatei
RT Class 1, IRT Class 3 Beispiel für ERTEC 200P	1	Usriod_main.c
DBAI-Beispiel (Direct Buffer Access Interface)	2	Usriod_main_dbai.c

Die Auswahl einer Beispielapplikation erfolgt in der Datei "**usrapp_cfg.h**" im Ordner "**application\app_common**".

Defaultmäßig eingestellt ist das RT/IRT-Beispiel mit:

```
#define EXAMPL_DEV_CONFIG_VERSION 1
```

Das RT/IRT-Beispiel beinhaltet folgende PROFINET IO-Device-Funktionen:

- Slot 1 → 64 byte Input-Modul
- Slot 2 → 64 byte Output-Modul
- Verschiedene modulbezogene Diagnosen
- Prozessalarm
- I&M0...4-Funktionen

Wenn Sie Änderungen für Ihre PROFINET IO-Device-Funktionen im Applikationsordner vornehmen, müssen Sie folgende Schritte durchführen:

1. Führen Sie alle notwendigen Änderungen in den Applikationsdateien durch und speichern Sie die geänderten Dateien.
2. Selektieren Sie den Projektordner "Pnio_src" und drücken Sie die rechte Maustaste.
3. Aktivieren Sie das Kommando **Build Project**, siehe Bild 4-8 Projekt-Build durchführen (Seite 26).

Ergebnis: Der Build-Vorgang wird gestartet und alle geänderten Dateien werden übersetzt. Nach Beendigung des Build-Vorgangs werden das ladbare PROFINET IO-Device-Beispiel ("PNIO4ECOS") und die Objektdatei ("ecos.bin") neu erzeugt, siehe Bild 4-10 Projektordner mit Bin-Datei und ladbarer Datei (Seite 28).

Siehe auch

Eclipse: Build der PROFINET IO-Applikation durchführen (Seite 26)

4.3 Konfiguration des Terminalprogrammes für Meldungsoutputs

In diesem Beispiel wird das Terminalprogramm "TeraTerm" benutzt, das als Freeware aus dem Internet geladen werden kann. Es kann aber auch "Microsoft HyperTerminal" verwendet werden.

1. Starten Sie das Terminalprogramm.
2. Führen Sie das Setup der seriellen Schnittstelle durch. Nehmen Sie dazu folgende Einstellungen vor:

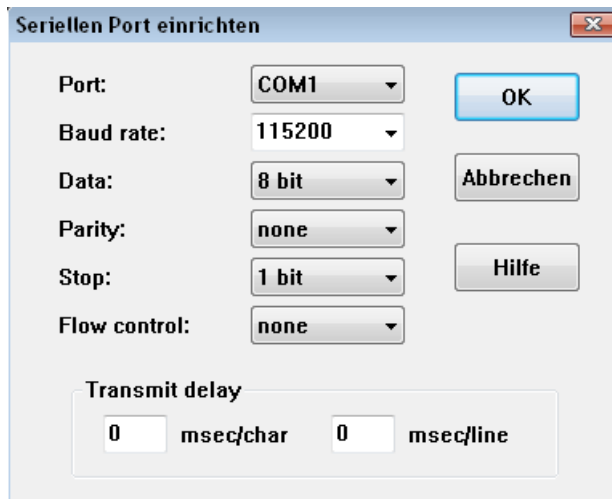


Bild 4-11 Serial Port Setup Parameter für Terminal

3. Klicken Sie zum Aktivieren auf **OK**.
4. Speichern Sie das Setup.

4.4 EB 200P in Betrieb nehmen

4.4.1 Übersicht Evaluation Board EB 200P

Steckerübersicht

Hinweis

Für den ordnungsgemäßen Betrieb muss das EB 200P im PC eingebaut sein.

Hinweis

Steckverbindungen (außer X1, X2 und X11) dürfen nur im ausgeschalteten Zustand gesteckt/gezogen werden!

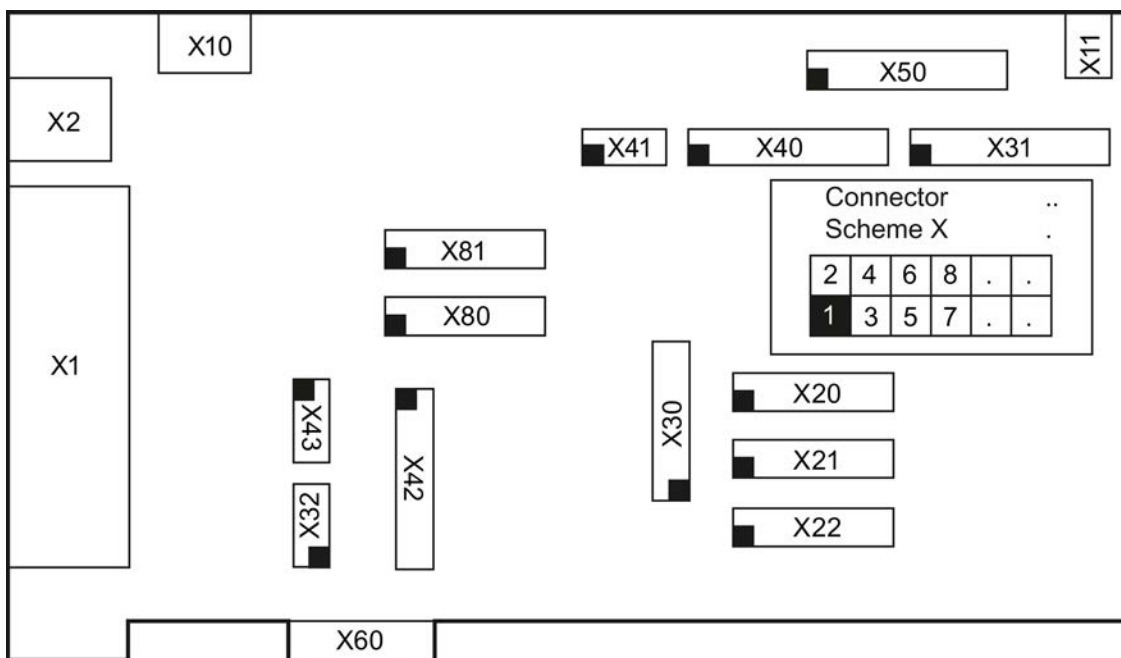


Bild 4-12 Steckerübersicht für Evaluation Board EB 200P

Verweis

Detaillierte Beschreibung des Evaluation Board EB 200P siehe /2/.

4.4.2 Verbindungen durchführen

Für den Betrieb des Evaluation Board EB 200P sind folgende Adaptionen durchzuführen:

1. An Stecker X10 24 VDC-Spannungsversorgung anschließen, aber noch nicht einschalten.
2. Installieren Sie den "RS232 via USB"-Treiber des PCs von der CD. Dieser befindet sich im Unterverzeichnis "\Tools\RS232 via USB driver". Folgen Sie dabei den Anweisungen des Installationsprogramms.
3. Verbinden Sie eine freie USB-Schnittstelle ihres PCs mit dem USB-Port (RS232 via USB) des EB 200P, siehe Terminal-Verbindung in Bild 4-12 Steckerübersicht für Evaluation Board EB 200P (Seite 31).

Hinweis

Verwenden Sie für die serielle Verbindung ein USB-Mini-B Kabel.

4. Verbinden Sie Ihren JTAG Debugger mit der JTAG-Schnittstelle (X31) des Evaluation Board, wie im folgenden Bild dargestellt.

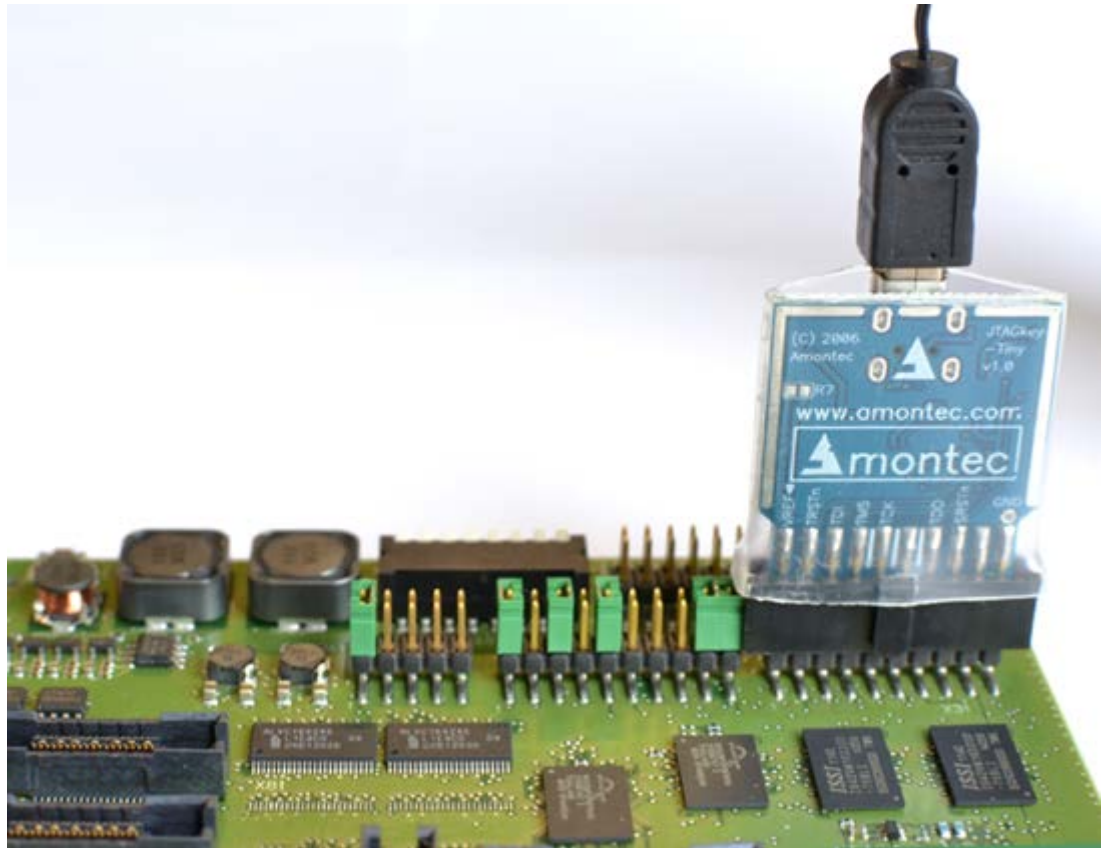


Bild 4-13 Verbindung des JTAG Debugger mit dem Evaluation Board

Beachten Sie: Die Jumper-Einstellungen in obiger Darstellung entsprechen **nicht** dem Auslieferungszustand!

Hinweis

Im Auslieferungszustand ist auf dem Evaluation Board bereits die Beispielapplikation auf dem Flash vorhanden und startet nach Spannungswiederkehr automatisch. Ein Update der geflashten Applikation ist über Ethernet möglich.

Dokumentation und Script-Dateien für einen OpenOCD-basierten JTAG Debugger finden Sie nach Installation der Produkt-CD auf Ihrem PC im Unterverzeichnis "PNIODevkit\OpenOcdDebug" in der Datei "Howto_use_Amontec_JTAGkey-Tiny_on_EB200P.pdf" oder auf der ComDeC Webseite (<http://www.siemens.de/comdec>).

5. Verbinden Sie eine freie Ethernet-Schnittstelle ihres PCs mit einem Switch Port des EB 200P (PN Port 1 oder PN Port 2).

Hinweis

Für die LAN-Verbindung vom/zum Evaluation Board dürfen nur Ethernet-Leitungen verwendet werden, die nicht länger als 100 m sind.

- Optional: Zur Aufzeichnung von Ethernet-Telegrammen verbinden Sie die beiden TAP-Ports mit einem Aufzeichnungssystem. Das EB 200P hat einen integrierten TAP, der optional zum Mitschneiden des Ethernet-Telegrammverkehrs dienen kann. Verbinden Sie die beiden TAP-Ports (Port 1-A und Port 2-B) mit je einer freien Schnittstelle eines Aufzeichnungssystems für Ethernet-Telegramme. Dies kann z. B. ein PC mit 2 freien Ethernet-Schnittstellen oder ein spezielles Hardwaresystem für diesen Zweck sein (z. B. CP1616 NetPROFI Software).

Hinweis

Für die LAN-Verbindung vom/zum Evaluation Board dürfen nur Ethernet-Leitungen verwendet werden, die nicht länger als 100 m sind.

- Wenn alle Verbindungen angeschlossen wurden, schalten Sie die Spannungsversorgung ein.

Evaluation Kit EK-ERTEC 200P mit PROFINET IO-Controller

Das folgende Bild zeigt die Verdrahtung für das PROFINET IO-Beispiel.

Im EB 200P ist ein Ethernet-TAP integriert, der im Kommunikationszweig zwischen PROFINET IO-Controller und PROFINET IO-Device geschaltet ist. Dieser TAP dient zur passiven Auskopplung von Ethernet-Telegrammen in einem geschalteten Netzwerk (nur Port 1). Mit dem Wireshark Network Protocol Analyzer können die PROFINET IO-Datenpakete aufgezeichnet und analysiert werden.

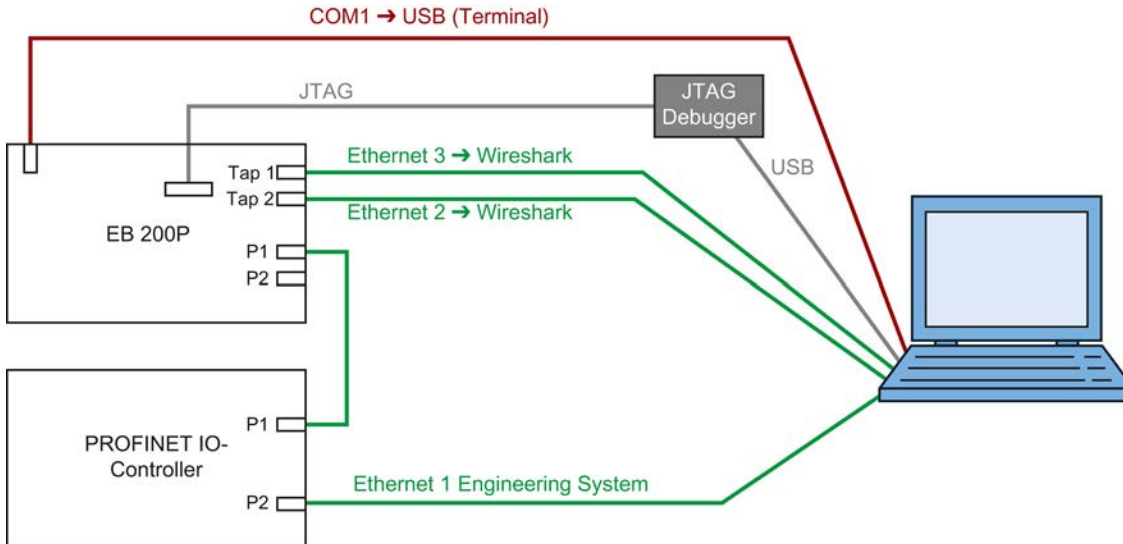


Bild 4-14 Aufbau PROFINET IO-Controller/Device für EK-ERTEC 200P mit einem beliebigen PROFINET IO-Controller

Siehe auch

Übersicht Evaluation Board EB 200P (Seite 31)

4.4.3 MAC-Adresse einstellen

MAC-Adresse

Im Auslieferungszustand des EB 200P ist eine Default-MAC-Adresse im Flash abgelegt. Dies ist bei allen EB 200P-Baugruppen die gleiche MAC-Adresse. Auf der Baugruppe ist ein Typenschild mit einer eindeutigen MAC-Adresse angebracht.



Bild 4-15 MAC-Adresse für Evaluation Board

Diese eindeutige MAC-Adresse muss auf der EB 200P-Baugruppe im Flash abgelegt werden. Das Ändern der MAC-Adresse kann aus dem Anwenderbeispiel heraus gestartet werden.

Führen Sie die folgenden Schritte durch.

MAC-Adresse ändern

1. Drücken Sie am Terminal die Taste "N" , dann <Enter>.
Auf dem Terminal erscheint folgender Text:

```
input mac address
***Bsp_nv_data_restore:NvDataType=0***
*** done Bsp_nv_data_restore ***
Current Ethernet address is: 00:0e:8c:b3:24:24
Modify all 6 bytes (board unique portion) of Ethernet Address
The first 3 bytes are manufacturer's default address block
00 - █
```

2. Ändern Sie die MAC-Adresse, wie auf dem Typenschild angegeben:
Byte 1 bis Byte 6 der MAC-Adresse nacheinander, wie auf dem Typenschild angegeben, eingeben. Nach der Eingabe des 6. Bytes und RETURN werden die Daten im Flash gespeichert.
Auf dem Terminal erscheint folgender Text:

```
00 - 00
0e - 0e
8c - 8c
b3 - b3
24 - 24
24 - 24
store new Ethernet address 00:0e:8c:b3:24:24
***Bsp_nv_data_store:NvDataType=0***
***Bsp_nv_data_store:NvDataType=0:Data already in Flash***
***Bsp_nv_data_memfree***
*-----*
* to activate the new mac address, perform a system restart *
*-----*
```

Automatisierungssystem einrichten

5.1 Automatisierungssystem einrichten

Das EB 200P-Device kann im Prinzip an jedem zertifizierten PROFINET IO-Controller betrieben werden. Auf der CD finden Sie dazu ein Beispielprojekt für eine SIMATIC CPU 317-2 PN/DP. Dieses können Sie bei Bedarf als Vorlage für die Erstellung einer Projektierung für einen anderen PROFINET IO-Controller verwenden.

Falls Sie eine SIMATIC CPU als Controller verwenden, müssen Sie STEP 7 als Engineering Tool installieren. PROFINET IO-Controller und Engineering System selbst sind nicht Bestandteil des Evaluation Kit.

Dieses Kapitel beschreibt die wesentlichen Schritte, um das auf der CD enthaltene Beispielprojekt für eine SIMATIC S7 CPU 317-2 PN/DP in das Engineering System STEP 7 zu importieren, die GSD-Datei zu installieren und die Projektierung auf den Controller zu laden.

Diese Dokumentation kann dabei nur die wesentlichen Schritte beschreiben. Weitere detaillierte Informationen zum PROFINET IO-Controller oder dem Engineering System sowie deren Bedienung finden Sie in der Anwenderdokumentation des jeweiligen Produktes.

5.2 Projektbeispiel für PROFINET IO-Device bearbeiten

Projektbeispiel

Auf der CD EK-ERTEC 200P PN IO sind im Verzeichnis "SIMATIC_STEP7" fertige und zur Device-Applikation passende STEP 7-Projekte für verschiedene Anwendungsfälle (Realtime, IRT, MRP, Shared Device, takt synchrone IRT-Applikation) enthalten. Diese beinhalten:

- die Busprojektierung mit SIMATIC S7 CPU 317-2 PN/DP-Controller und PNIO-Device
- Projektierung der Eingänge und Ausgänge des PNIO-Device

Tipp: Es wird empfohlen, für die Erstinbetriebnahme diese Beispiele zunächst unverändert zu verwenden.

Projektbeispiel bearbeiten

1. Projektdatei in STEP 7 importieren

Öffnen Sie STEP 7 und importieren Sie die Datei "Cpu317-2_EB200P_RT.zip". Öffnen Sie dazu in STEP 7 das Menü **Datei** und wählen Sie dort die Option **Dearchivieren**.

2. RT-Projekt öffnen und bearbeiten

In STEP 7 das Menü **Datei > Öffnen** auswählen.

Im Fenster "Öffnen" wählen Sie das gesuchte Projekt aus und öffnen dieses mit **OK**.

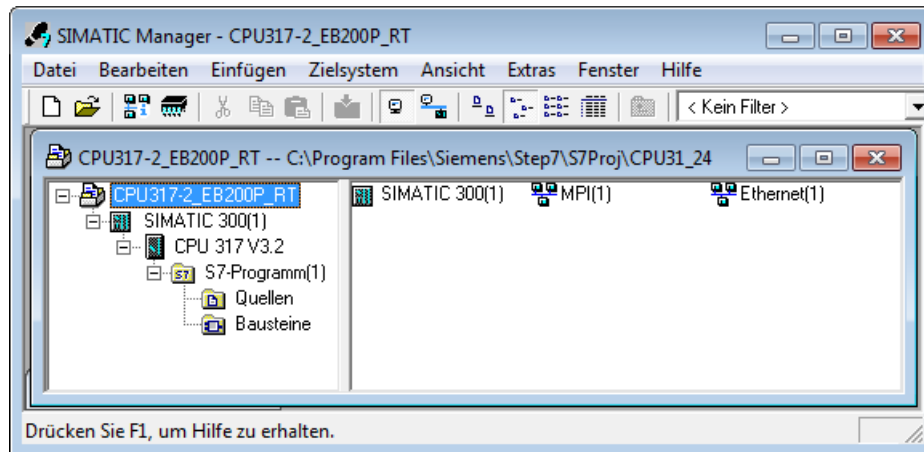


Bild 5-1 RT-Projekt öffnen und bearbeiten

3. Hardware-Katalog aktualisieren

Im geöffneten Projekt "SIMATIC 300(1)" auswählen und mittels rechter Maustaste die Option "Öffnen..." auswählen. Damit wird das Hardwarekonfigurationstool HW-Konfig geöffnet.

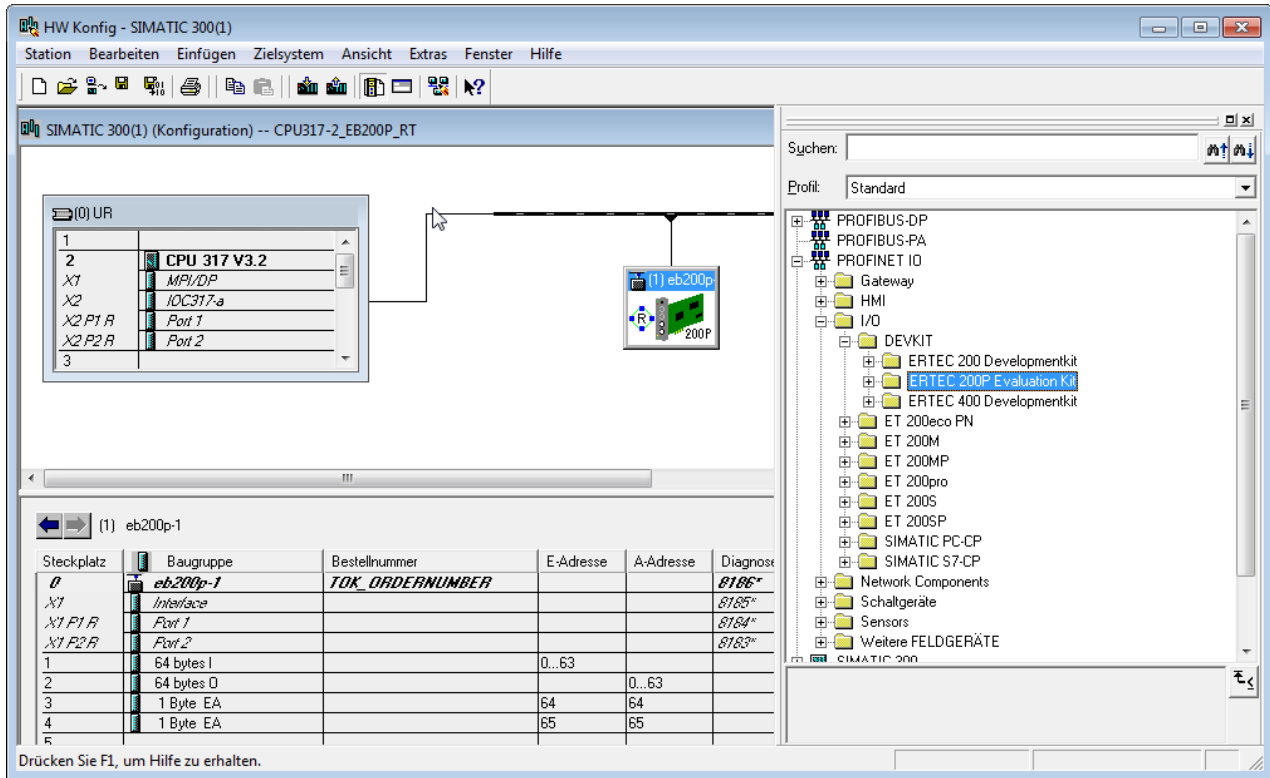


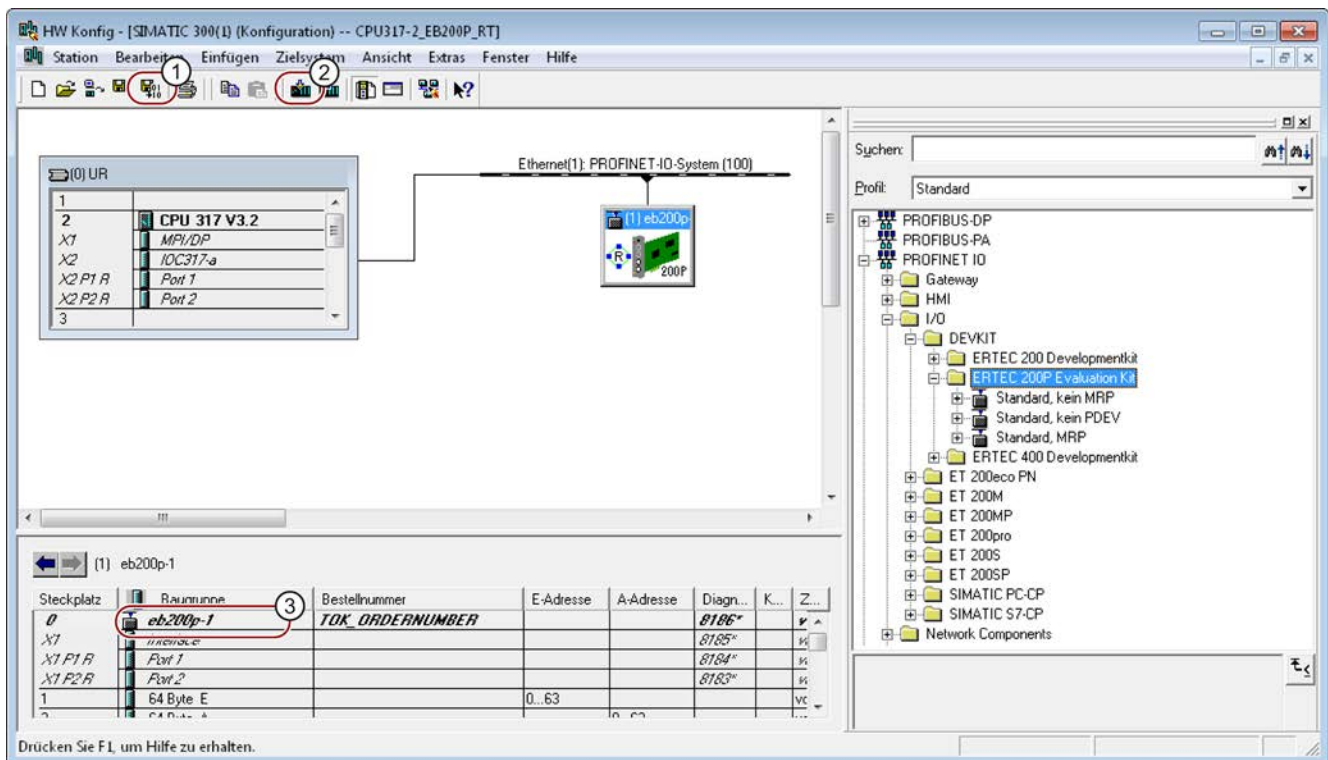
Bild 5-2 Hardware-Katalog aktualisieren

- Im Menü **Extras > GSD-Dateien installieren** auswählen.
- Im Fenster "GSD-Dateien installieren > Durchsuchen" das Verzeichnis "GSDML" auf der CD "EK-ERTEC200P_PNIO" auswählen und die angezeigte GSD-Datei installieren mittels Schaltfläche **Installieren**.

5.3 PROFINET IO-Projekt mit STEP 7 bearbeiten und in CPU 317 laden

Das HW-Konfig-Fenster zeigt die PROFINET IO-Buskonfiguration mit folgenden Komponenten:

Komponente	Gerätename	Typ	Eingestellte Parameter
CPU 317 V3.2	IOC317-a	PROFINET IO-Controller	IP: 192.168.20.171
EB 200P	eb200p-1	PROFINET IO-Device	IP: 192.168.20.182
		Slot 1 → 64 byte Input-Modul Slot 2 → 64 byte Output-Modul	Eingangsadresse 0 - 63 Ausgangsadresse 0 - 63



- ① Speichern und Übersetzen
- ② Konfiguration in Controller laden
- ③ Gerätename "eb200p-1" für das Device in diesem Beispiel

Bild 5-3 Projektieren eines PROFINET Device mit EB 200P in HW-KONFIG

Falls Sie Änderungen an der Projektierung vornehmen wollen, gehen Sie folgendermaßen vor:

1. Änderung durchführen

- Selektieren Sie die Komponente, die Sie ändern möchten, und drücken Sie die rechte Maustaste. Wählen Sie "Objekteigenschaften".
- Ein Fenster wird geöffnet, in dem Sie Ihre Änderungen durchführen können.
- Nachdem alle Änderungen durchgeführt wurden, schließen Sie die Projektierung im Menü "Station" mit der Option **Speichern und Übersetzen** ab (siehe Bild oben).

2. Download der Projektierung in den PROFINET IO-Controller

- Schließen Sie alle Komponenten an, wie beschrieben in Kapitel Verbindungen durchführen (Seite 32).
- Laden Sie die Buskonfiguration in den PROFINET IO-Controller im Menü **Zielsystem** durch Anwahl der Option **Laden**.

3. Überprüfen der angeschlossenen PROFINET IO-Teilnehmer

- Wählen Sie in HW-Konfig das Menü **Zielsystem > Ethernet-Teilnehmer bearbeiten** aus.

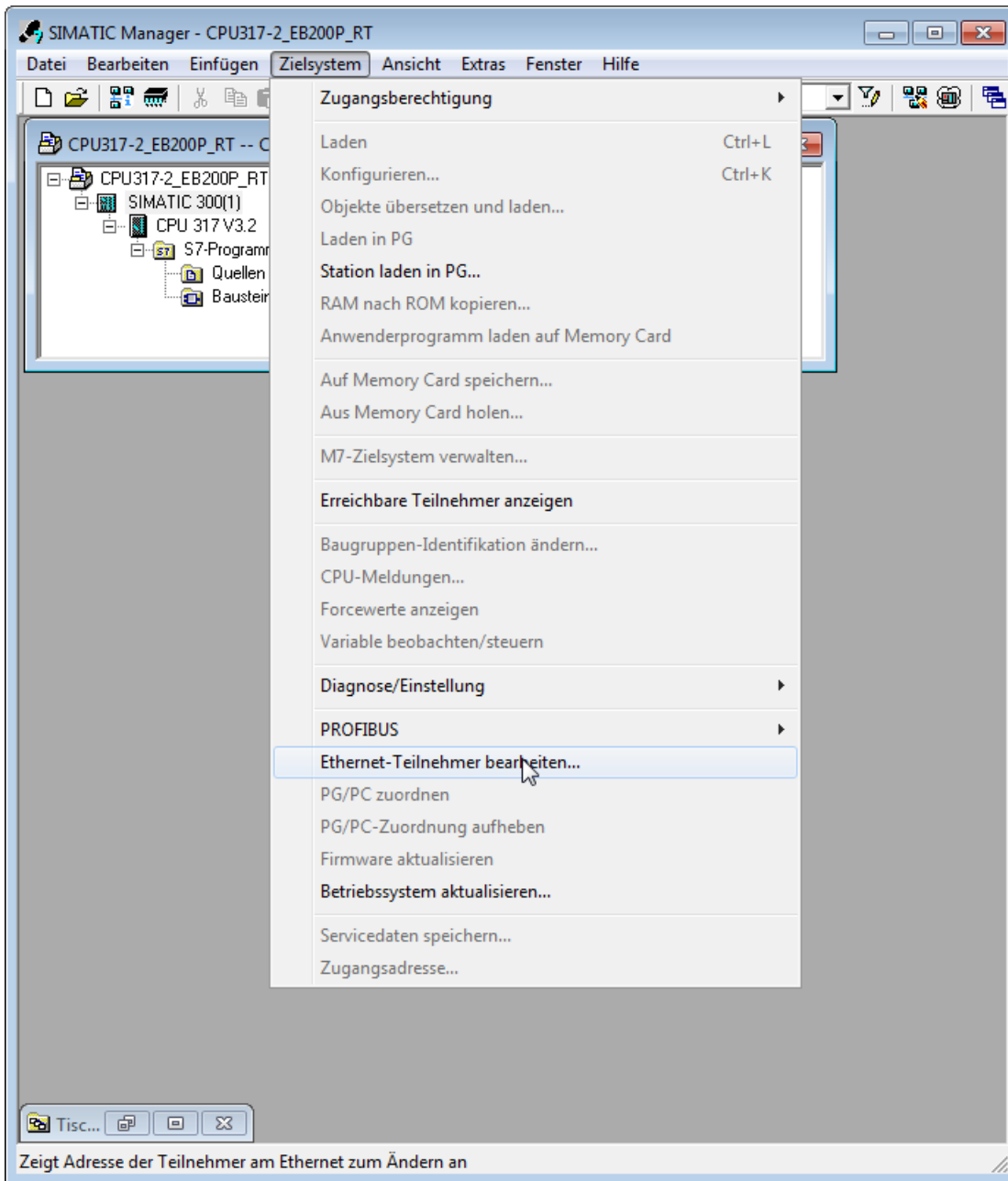


Bild 5-4 Alle Ethernet Teilnehmer überprüfen

- Klicken Sie anschließend auf die Schaltfläche **Durchsuchen....** Der Konfigurator scannt daraufhin einige Sekunden lang den Ethernet Bus und zeigt im Dialogfenster "Ethernet-Teilnehmer bearbeiten" alle online erreichbaren PROFINET IO-Teilnehmer mit MAC-Adresse, IP-Adresse (falls vorhanden), Gerätenamen und Gerätetyp an.

- Selektieren Sie mit der Maus den Teilnehmer, der bearbeitet werden soll, und bestätigen Sie mit **OK** (siehe folgendes Bild).

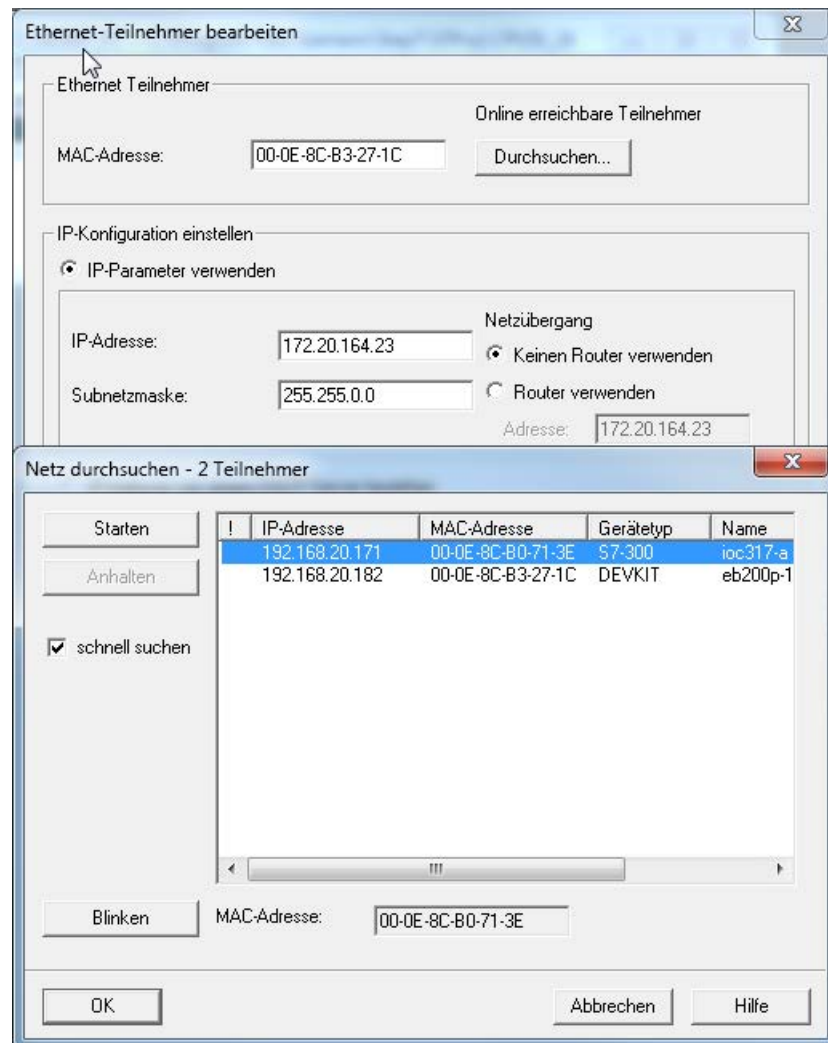


Bild 5-5 Alle Teilnehmer im Netz anzeigen

- Zuweisen von Device-Name/IP-Adresse eines ausgewählten PROFINET IO-Teilnehmers
 - Im Feld "Gerätename vergeben" den gewünschten Device-Namen eintragen und zuweisen mit der Schaltfläche **Name zuweisen** (siehe folgendes Bild).

Hinweis

Bei einer Änderung des Gerätenamens ist Folgendes zu beachten:

- Der Gerätename muss innerhalb der Anlage eindeutig sein.
- Der Gerätename auf dem Device muss mit der HW-Konfig-Projektierung übereinstimmen.
- Im Anwenderbeispiel wird auf dem Device bei einer Änderung des Gerätenamens die Device-Applikation benachrichtigt und der neue Name im Flash gespeichert.

- Um die IP-Adresse zu ändern, muss im Feld "IP-Konfiguration einstellen" die Option "IP-Parameter verwenden" gesetzt werden. Danach kann die gewünschte IP-Adresse und die Subnetz-Maske eingetragen und mit Schaltfläche **IP-Konfiguration zuweisen** zugewiesen werden (siehe folgendes Bild).

Hinweis

Normalerweise setzt der Controller die projektierte IP Adresse im Device, wenn der Device-Name in der Projektierung und im Device übereinstimmt. Deshalb ist ein Setzen der IP-Adresse hier nicht unbedingt erforderlich.

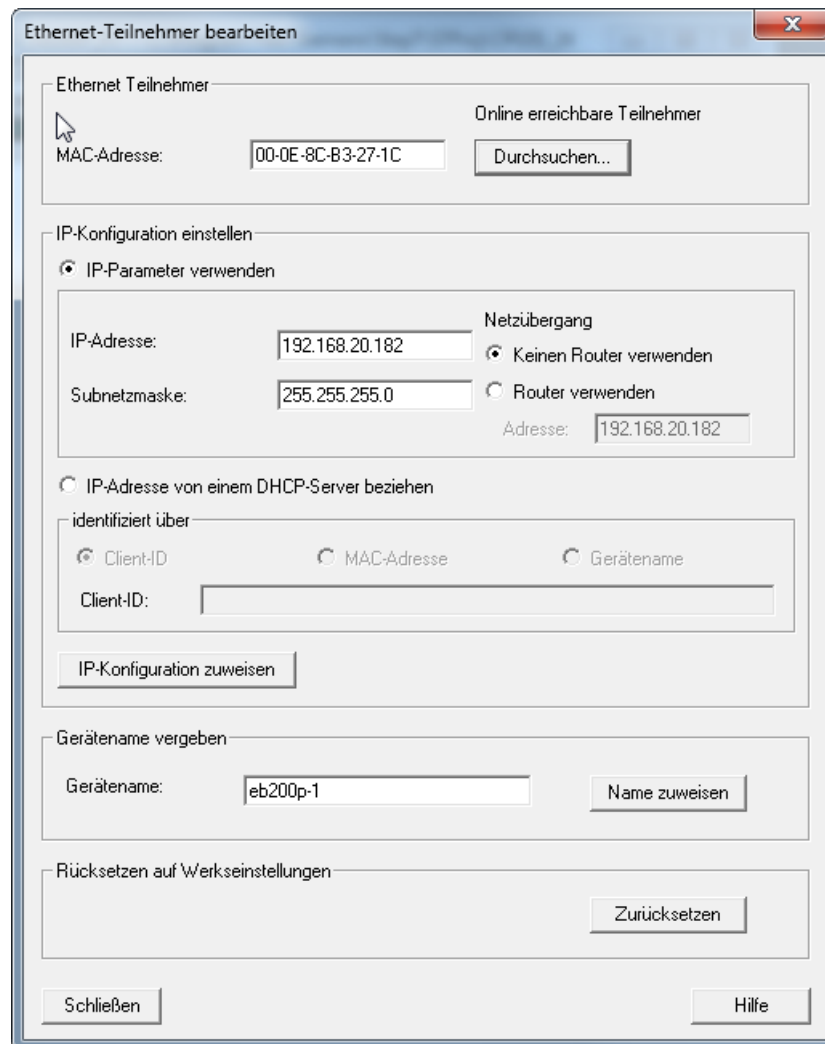


Bild 5-6 Device-Name und IP-Parameter zuweisen

Erste Schritte mit PROFINET IO-Device-Kommunikation

6

Dieses Kapitel beschreibt die ersten Schritte zum eigenen Device. Wir empfehlen, das mitgelieferte Device-Beispiel und die Projektierung zunächst unverändert zu lassen und damit die ersten Versuche zu starten. Das Beispiel ist auf der Hardware-Plattform EB 200P getestet. Nachdem Sie sich mit den allgemeinen PROFINET IO-Features vertraut gemacht haben, kann Schritt für Schritt die Anpassung des eigenen Devices durchgeführt werden.

6.1 PROFINET IO-Device-Kommunikation starten

Folgende Voraussetzungen müssen erfüllt sein:

- Die Verkabelung wurde durchgeführt und die Stromversorgung an den Komponenten ist eingeschaltet, siehe Kapitel Verbindungen durchführen (Seite 32).
- Ein Terminalprogramm (z. B. TeraTerm) auf dem PC wurde gestartet und auf die passende RS232-Schnittstelle (COM-Schnittstelle) adaptiert (Einstellungen siehe Konfiguration des Terminalprogrammes für Meldungsangaben (Seite 30)).
- Die Controller-Projektierung wurde geladen und die Controller-Beispielapplikation wurde gestartet.

Am Terminal werden folgende Textzeilen ausgegeben, die Auskunft über den Ablauf geben, z. B.:

- Restore Daten aus NV-Memory
- Initialisierung abgeschlossen und Applikation gestartet

```
***Bsp_nv_data_restore:NvDataType=2***
*** done Bsp_nv_data_restore ***
***Bsp_nv_data_restore:NvDataType=0***
*** done Bsp_nv_data_restore ***
***Bsp_nv_data_memfree***
***Bsp_nv_data_restore:NvDataType=0***
*** done Bsp_nv_data_restore ***
***Bsp_nv_data_memfree***
***Bsp_nv_data_restore:NvDataType=0***
*** done Bsp_nv_data_restore ***
***Bsp_nv_data_memfree***
***Bsp_nv_data_restore:NvDataType=3***
*** done Bsp_nv_data_restore ***
##REMA DATA STORE total Bufsize=700
SubPlugResponse(0) = OK
SubPlugResponse(1) = OK
SubPlugResponse(2) = OK
SubPlugResponse(3) = OK
SubPlugResponse(4) = OK
SubPlugResponse(5) = OK
SubPlugResponse(6) = OK
SubPlugResponse(7) = OK
***Bsp_nv_data_memfree***
***Bsp_nv_data_memfree***
AutoFlash_Task started...
TCP interface initialized
TCP interface wait on connection...
PNIO DEVKIT appl.-example no.9, V4.0.100.1, 2012-12-21 10:57
SYSTEM STARTUP FINISHED, OK
```

Zu diesem Zeitpunkt ist der Hochlauf des PROFINET IO-Device Stacks abgeschlossen und die Kommunikation zwischen Controller und Device wird nun automatisch gestartet.

6.1 PROFINET IO-Device-Kommunikation starten

Folgende Ausgaben werden beim Verbindungsaufbau am Terminal angezeigt, z. B.:

- Anzahl der Module und Modulinformationen
- Ende der Parametrierung (submodul-granular)
- erste gültige Daten bzw. IOxS vom PNIO Controller erhalten (submodul-granular)

```
##CONNECT_IND HostIp = 0x0 AR = 0 AR type = 16
##OWNERSHIP_IND AR = 1 number of submodules = 5
  Api=0 Slot=0 Sub=1 ModID=3 SubID=1 OwnSessKey=1025 isHrongs=0
  Api=0 Slot=0 Sub=32768 ModID=3 SubID=2 OwnSessKey=1025 isHrongs=0
  Api=0 Slot=0 Sub=32769 ModID=3 SubID=3 OwnSessKey=1025 isHrongs=0
  Api=0 Slot=0 Sub=32770 ModID=3 SubID=3 OwnSessKey=1025 isHrongs=0
  Api=0 Slot=2 Sub=1 ModID=49 SubID=1 OwnSessKey=1025 isHrongs=0
**@sp_nv_data_store:NvDataType=2:Data already in Flash***
##PARAM END for Api=0, Slot=0, Sub=1, ArHnd1=1, Session=1025
##PARAM END for Api=0, Slot=0, Sub=32768, ArHnd1=1, Session=1025
##PARAM END for Api=0, Slot=0, Sub=32769, ArHnd1=1, Session=1025
##PARAM END for Api=0, Slot=0, Sub=32770, ArHnd1=1, Session=1025
##PARAM END for Api=0, Slot=2, Sub=1, ArHnd1=1, Session=1025
##PARAM SHADOW MEM STORE total memsize=700
**@sp_nv_data_store:NvDataType=3***
##AR IN-Data event indication received, ArHnd1 = 0h, Session = 0h
neu IO controller provider status = 0x80 in slot 2, subslot 1
**@sp_nv_data_store:NvDataType=3:Data already in Flash***
##CONNECT_IND HostIp = 0x0 AR = 1 AR type = 1
##OWNERSHIP_IND AR = 2 number of submodules = 1
  Api=0 Slot=1 Sub=1 ModID=48 SubID=1 OwnSessKey=1026 isHrongs=0
neu IO controller provider status = 0x80 in slot 2, subslot 1
##PARAM END for Api=0, Slot=1, Sub=1, ArHnd1=2, Session=1026
##AR IN-Data event indication received, ArHnd1 = 1h, Session = 0h
neu IO controller consumer status = 0x80 in slot 1, subslot 1
```

Für den Hochlauf wurde eine PROFINET IO-Busaufzeichnung durchgeführt. Diese Datei kann mit dem Wireshark Network Protocol Analyzer geöffnet und analysiert werden.

Busaufzeichnung siehe Dateianlage "Start_Up_RT.pcap".

6.2 PROFINET IO-Device-Kommunikation testen

Verschiedene PROFINET IO-Features können mit Hilfe von Tastatureingaben aktiviert/deaktiviert werden. Um das Hilfsmenü anzuzeigen, geben Sie ? am Terminal ein.

Es erscheint folgende Liste mit (z. B.) folgenden Angaben:

- Start des Hilfsmenüs
- PN IO-Prozessalarm senden
- PN IO-Diagnosen aktivieren/deaktivieren
- IO-Daten anzeigen
- LEDs schalten
- MAC Adresse einstellen
- Module/Submodule ziehen/stecken

```

COMMAND LIST   PNIO DEVKIT appl.-example no.9, V4.0.100.1, 2012-12-21 10:57

CONTROL:
'a' send process alarm on slot1, subslot1
'B / b' send channel diag-alarm 'line break' appear/disapp. on slot1, subsl.1
'C / c' send generic diag-alarm appear/disapp. on slot2, subslot1
'E / e' send extended channel diag-alarm appear/disapp. on slot2, subsl.1
'Q / q' increment/decrement input data values
'f' download firmware via TCP and store into flash
'G' send upload alarm (backup) to ipar-server
'g' send retrieval alarm (restore) to ipar-server
'j' print cyclic io data
'K' set RS232 output trace level
'k' set MEH output trace level
'L' set LED (1..N, 0:all) on
'l' set LED (1..N, 0:all) off
'N' set MAC address and store in flash
'p' enable user printf output on serial console
'p' disable user printf output
'S' plug a submodule
's' pull a submodule
'T' set trace level for a package
't' set trace level for a single subsystem
'z' measure system performance
'D' Perform System Reboot

```

Diese Liste kann vom Anwender selbständig erweitert werden, siehe Dateien im Ordner "application":

- "\App1_Standard\usriod_main.c "
- "\App2_DBA\usriod_main_dbai.c"

Machen Sie sich mit dem PROFINET IO-Protokoll vertraut:

1. Aktivieren Sie eine Auswahl an Kommandos in der Kommandoliste und zeichnen diese mit dem Wireshark Network Protocol Analyzer auf.
2. Führen Sie anschließend die Analyse der Aufzeichnung durch.

Applikationsbeispiele

Im Evaluation Kit sind verschiedene Applikationsbeispiele enthalten, um eine PROFINET IO-Kommunikation zwischen einem Controller und einem Device (EB 200P) herstellen zu können. Die enthaltenen Beispiele sind aufeinander abgestimmt ("ready to use") und betreffen folgende Funktionalitäten:

- RT-Kommunikation
- IRT-Kommunikation
- IRT-Kommunikation mit taktsynchroner Applikation
- Medienredundanz (MRP)
- Shared Device, d. h. Zugriff von zwei Controllern auf ein Device
- TCI
- PROFIenergy

Zu diesen Funktionen aus Anwendersicht (Use Cases) ist auf der CD jeweils ein passendes Device-Applikationsbeispiel (siehe Kapitel Beispiele für Device-Applikationen (Seite 50)), eine GSD-Datei (siehe Kapitel GSD-Dateien (Seite 62)) und eine Projektierung (siehe Kapitel Controller-Applikationsbeispiele (Seite 63)) enthalten. Die Kombination der passenden Device-Anwenderbeispiele und Projektierung (z. B. für SIMATIC CPU 3xx) ist jeweils bei den Use Cases dokumentiert.

7.1 Beispiele für Device-Applikationen

Im Sourcecode im Verzeichnis "Application" sind drei verschiedene Anwenderbeispiele implementiert, die sich im Wesentlichen in der Art des Zugriffs auf die IO-Daten unterscheiden.

- Im Unterverzeichnis "App1_Standard" ist der Beispielcode für den Zugriff über das Standard-Interface (SI) hinterlegt,
- im Verzeichnis "App2_DBAI" hingegen der Code für den IO-Datenzugriff über das Direct Buffer Access Interface (DBAI).
- In "App3_IsoApp" liegt ein Anwenderbeispiel für IRT mit taktischer Applikation inklusive der T_IO_Input- und T_IO_Output-Behandlung.

In "App_common" liegen alle Dateien, die von den Anwenderbeispielen gemeinsam verwendet werden.

Alle Funktionalitäten, wie z. B. RT/IRT-Betrieb, PROFIenergy, Redundanz etc., sind in diese Beispiele integriert. Gegebenenfalls müssen bestimmte Compilerschalter in Header-Dateien geändert werden, die entweder im Applikationsverzeichnis selbst oder in der Systemadaption "(...)\sysadapt1\cfg" abgelegt sind.

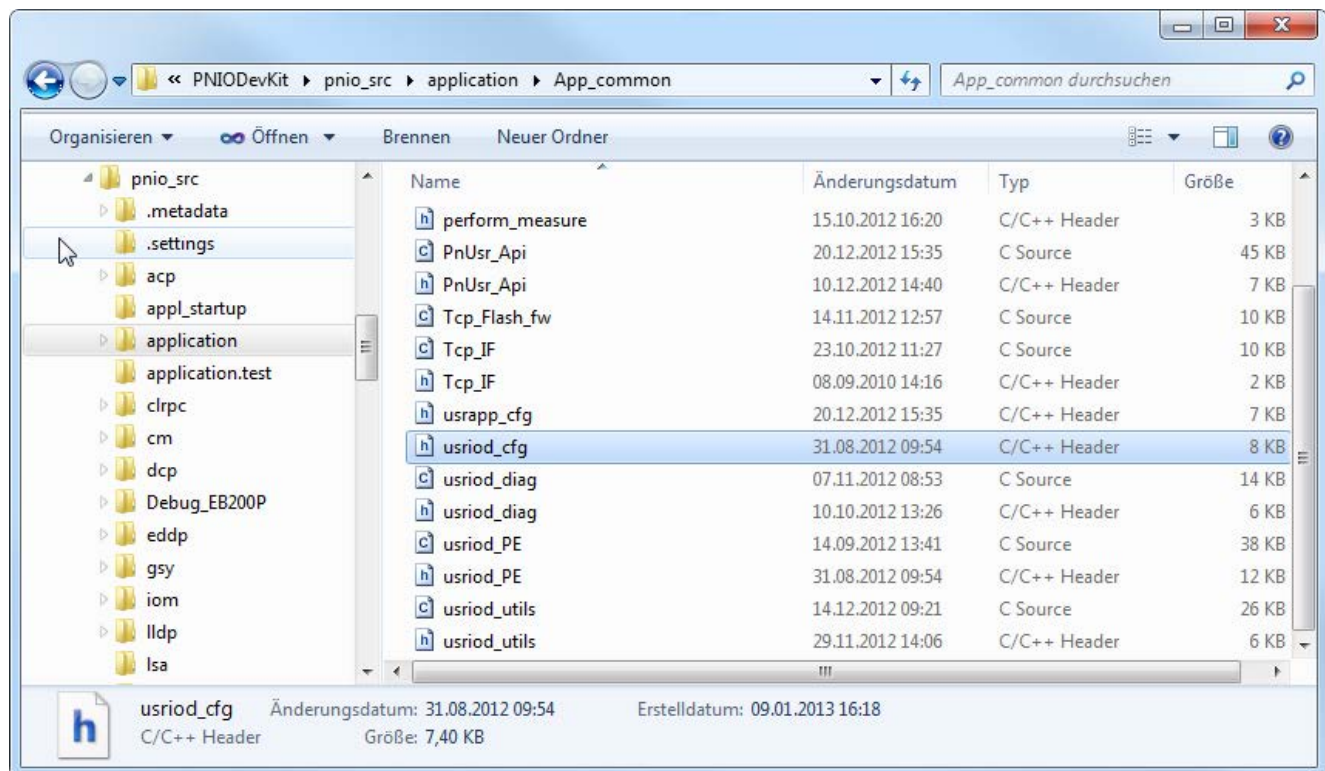


Bild 7-1 Konfigurationsdatei "usriod_app.h"

Weitere Beispiele "App3_xxx", "App4_yyy", können integriert werden. Die Auswahl des entsprechenden Anwenderbeispiels erfolgt beispielsweise über den Compilerschalter

```
#define EXAMPL_DEV_CONFIG_VERSION
```

in der Datei "\App_common\usrapp_cfg.h".

Die Bedienung der Anwenderbeispiele erfolgt mit einem Terminalprogramm, das über USB an die UART des ERTEC 200P angeschlossen wird (Anschluss X11). Die Schnittstellenparameter der UART sind:

"115 kBaud, 8bit data, no parity, 1 stopbit, no flow control"

Der grundsätzliche Aufbau der Beispiele sowie die Konsolenbedienung sind weitgehend ähnlich. Sie bestehen jeweils aus den beiden Tasks "mainAppl()" und "Task_CycleIO", die jeweils in der Datei "usriod_main.c" (SI) bzw. "usriod_main_dbai.c" (DBAI) enthalten sind. Die Aufgaben dieser Tasks sind:

- "mainAppl()":
 - Initialisierung und Start des PNIO-Stacks
 - Übergabe der Ist-Modulkonfiguration an den PNIO-Stack
 - Erzeugen und Start von "Task_CycleIO"
 - Bedienung der Konsolenschnittstelle in einer Endlosschleife
- "Task_CycleIO()":
 - Lesen der Output-Daten vom PROFINET IO-Controller und Ausgabe an physikalische Ausgänge, Bereitstellung des IOCS an den Stack (der IOCS sagt aus, ob die Ausgänge ordnungsgemäß ausgegeben werden konnten).
 - Lesen der physikalischen Eingänge und Übergabe der Daten, zusammen mit dem Provider State IOPS, an den PROFINET IO-Stack (der IOPS sagt aus, ob die eingelesenen Daten gültig sind).
 - Warten bis zum nächsten Zyklus

Hinweis

Wichtig ist, dass alle Tasks, die über die Anwenderschnittstelle mit dem PROFINET-Stack kommunizieren, mit der Funktion "OsCreateThread()" erzeugt wurden und, dass ihnen mit "OsCreateMsgQueue()" eine Message Queue zugeordnet wurde.

7.1.1 Device-Applikationsbeispiel für das Standard-Interface (SI)

Dieses Anwenderbeispiel ist bereits im Auslieferungszustand auf dem Flash des Evaluation Boards hinterlegt und startet automatisch nach dem Einschalten der Versorgungsspannung. Zusätzlich ist es als ablauffähiges Image im Unterverzeichnis "BootableBinary" abgelegt.

Das "SI" beruht auf einem Callback-Konzept, indem nach dem Anstoß der Datenübertragung (lesend oder schreibend) für den IO-Datenaustausch für jedes Submodul jeweils eine Callback-Funktion aufgerufen wird. Das Handling wird für die Applikation dadurch sehr einfach, da diese weder die Struktur der IOCR kennen muss, noch die einzelnen Application Relations (ARs) verwalten muss. Die Sichtweise der Applikation auf die Daten ist submodul-bezogen.

Das "DBAI" hingegen hat eine AR- und IOCR-bezogene Sichtweise der Applikation auf die Daten. Die Applikation muss die einzelnen ARs verwalten und für jede AR einen eigenen Datenaustausch durchführen. Dazu erhält die Applikation die Pointer auf die Input- und Output-IOCR und kann direkt darin die IO-Daten bzw. IOPS/IOCS lesen und schreiben. Voraussetzung dafür ist aber, dass die Applikation sowohl die einzelnen ARs als auch den Aufbau der darin enthaltenen Daten-IOCRs kennt. Sie erhält diese mit dem Connect-Telegramm.

Weitere Informationen zu SI und DBAI finden Sie in der Schnittstellenbeschreibung /3/.

Eine Bedienungsanleitung zu diesem Beispiel finden Sie außerdem im Kapitel Erste Schritte mit PROFINET IO-Device-Kommunikation (Seite 45).

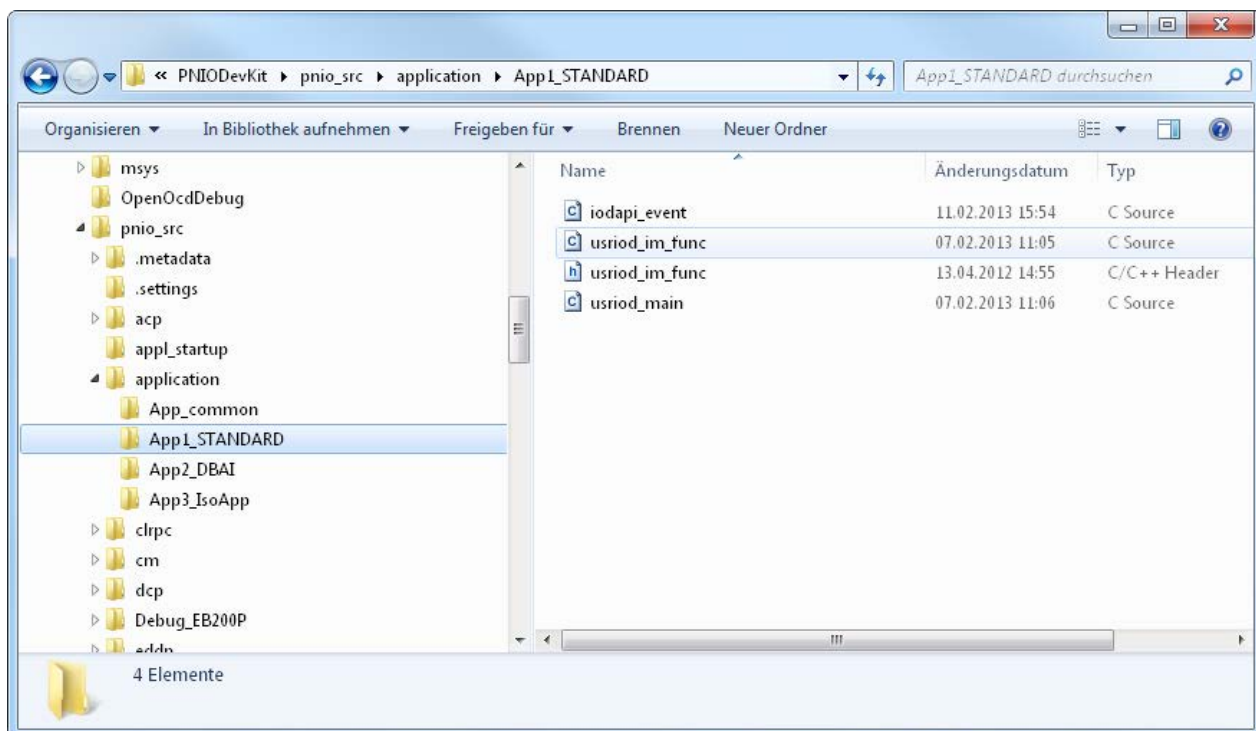


Bild 7-2 Dateien des Standard-Interface-Beispiels (SI)

Dieses Anwenderbeispiel wird eingebunden, wenn in der Header-Datei "Application\App_Common\usrapp_cfg.h"

```
# define EXAMPL_DEV_CONFIG_VERSION 1
```

gesetzt ist.

7.1.2 Device-Applikationsbeispiel für das Direct Buffer Access Interface (DBAI)

Den spezifischen Sourcecode für dieses Anwenderbeispiel finden Sie im Verzeichnis "(...)\Application\App2_DBAI".

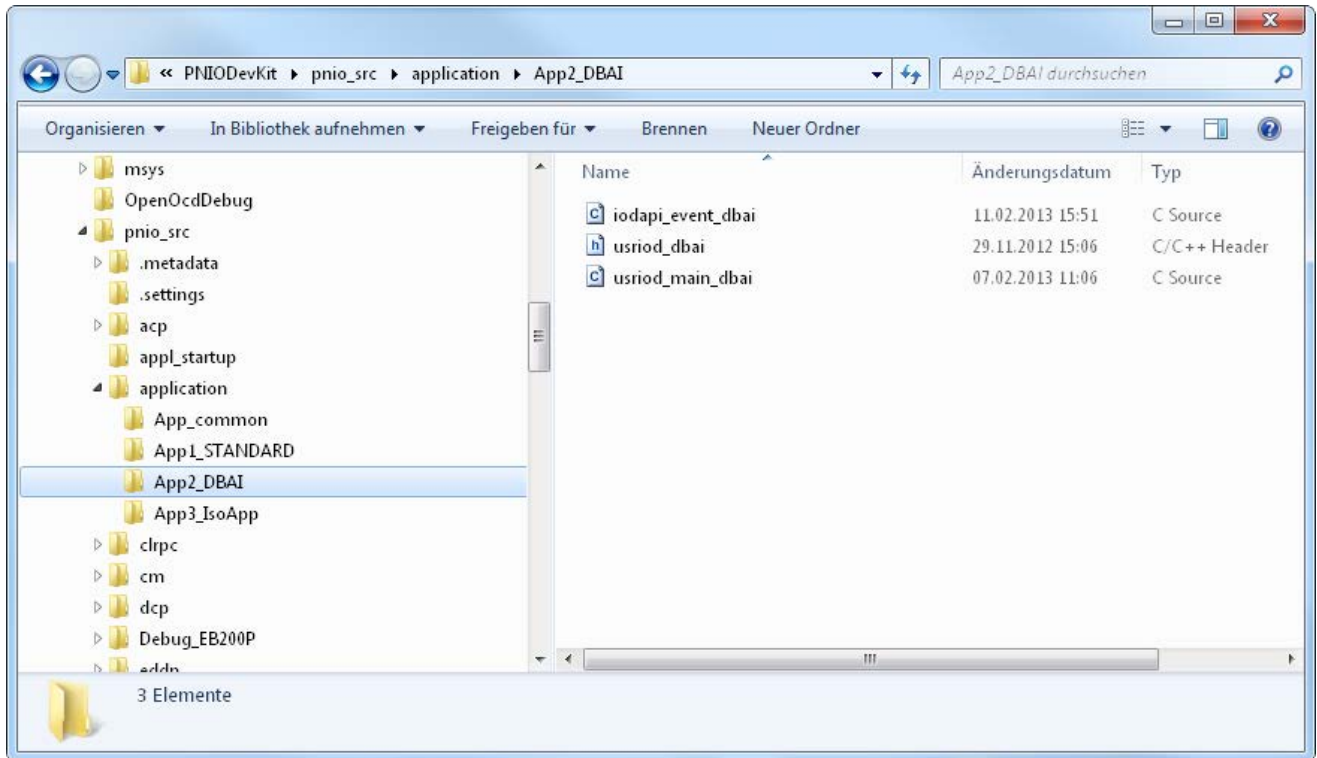


Bild 7-3 Dateien des DBAI-Beispiels

Dieses Anwenderbeispiel wird eingebunden, wenn in der Header-Datei "`\"Application\App_Common\usrapp_cfg.h`"

```
# define EXAMPL_DEV_CONFIG_VERSION 2
```

gesetzt ist.

7.1.3 Device-Applikationsbeispiel für eine taktsynchrone Applikation mit IRT

Den spezifischen Sourcecode für dieses Anwenderbeispiel finden Sie im Verzeichnis "(...)\Application\App3_IsoApp".

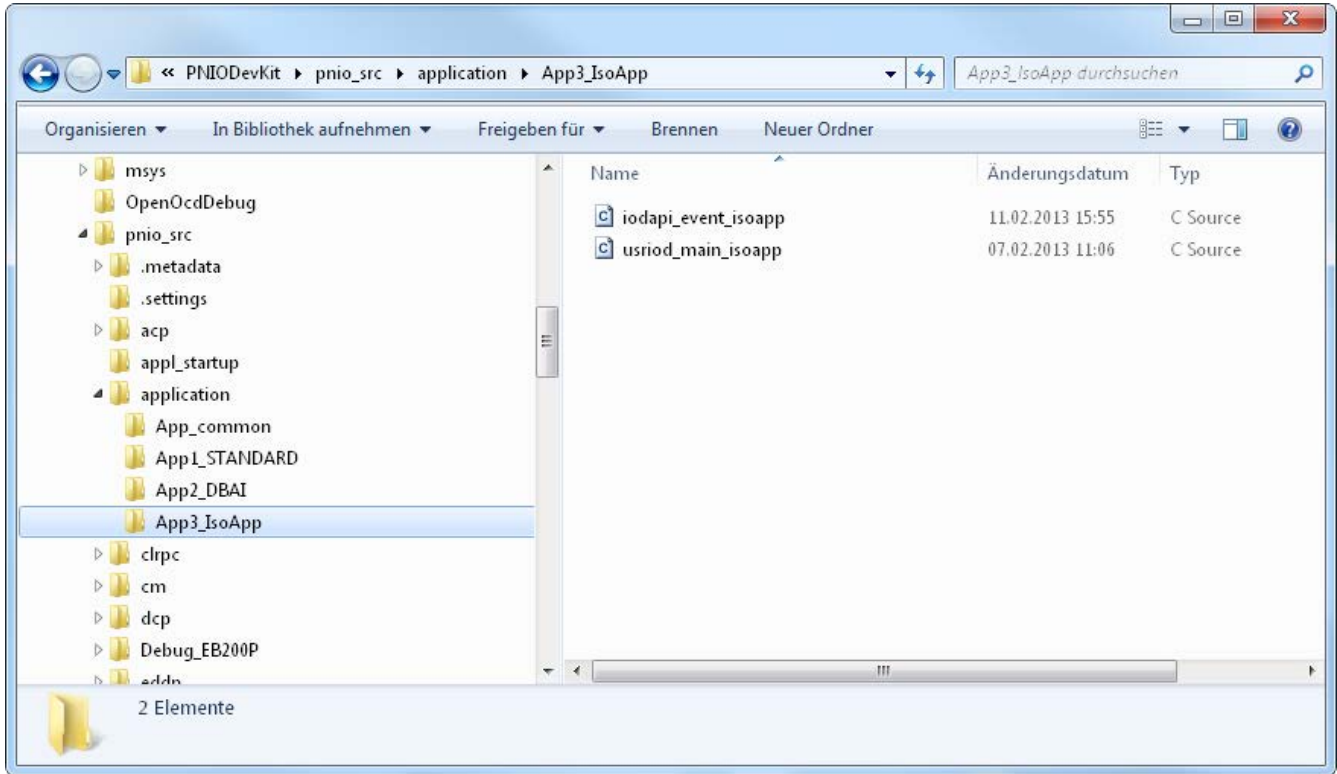


Bild 7-4 Dateien des IsoApp-Beispiels

Dieses Anwenderbeispiel wird eingebunden, wenn in der Header-Datei "`\Application\App_Common\usrapp_cfg.h`"

```
# define EXAMPL_DEV_CONFIG_VERSION 3  
gesetzt ist.
```

7.2 Funktionalitäten aus Anwendersicht (Use Cases)

7.2.1 Use Case RT

Für RT können Sie sowohl das Standard-Interfacebeispiel "App1_Standard" als auch das Beispiel für den direkten Pufferzugriff "App2_DBAI" unverändert verwenden.

Für die Projektierung verwenden Sie die zu Ihrem Controller passende "xxx_RT.zip"-Projektdatei, siehe Kapitel Controller-Applikationsbeispiele (Seite 63).

7.2.2 Use Case IRT (ohne takt synchrone Applikation)

Das Standard-Interfacebeispiel "App1_Standard" kann sowohl für RT als auch für IRT verwendet werden. Die konfigurierten Standard-IO-Module dieser Beispielapplikation können im Prinzip ebenfalls sowohl für RT als auch für IRT verwendet werden.

Für die Projektierung verwenden Sie die Projektdatei "CPU317-2_EB200P_IRT.zip" oder eine zu Ihrem Controller passende Projektdatei, siehe Kapitel Controller-Applikationsbeispiele (Seite 63).

Benötigen Sie für IRT auch eine takt synchrone Applikation, die die im Engineering konfigurierten Werte für T_IO_Input und T_IO_Output korrekt behandelt, so ist dafür ein eigenes Anwenderbeispiel im Evaluation Kit enthalten, siehe Kapitel Use Case IRT mit takt synchroner Applikation (Seite 56).

Hinweis

Die Ausgabe von Konsolenmeldungen, aber auch ein angeschlossener JTAG-Debugger können das Ablauf timing der Software beeinflussen. Dies kann eventuell bei IRT-Betrieb mit kurzen Zykluszeiten (z. B. 250 µs) zu Problemen führen. In diesem Fall sollten die Konsolenmeldungen abgeschaltet bzw. der JTAG-Debugger entfernt werden.

7.2.3 Use Case IRT mit taktsynchroner Applikation

Mit einer taktsynchronen PROFINET-Applikation ist es möglich, die Anwendung hochgenau mit dem Bustakt zu synchronisieren. Dabei werden von der Applikation sogenannte T_IO_Input- und T_IO_Output-Werte mittels eines speziellen Anlauf-Records (IsochronousModeData, Record-Index 0x8030) vorgegeben, die vom Engineering System konfiguriert wurden. Damit ist es möglich, geräteübergreifend Eingangssignale zu einem genau vorgegebenem Zeitpunkt (T_IO_Input) zu sampeln und Ausgangssignale zu einem genau vorgegebenem Zeitpunkt (T_IO_Output) zu aktivieren. Die beiden Parameter "T_IO_Input" und "T_IO_Output" sind dabei Delay-Werte, die die Verzögerung zum Zyklusbeginn (NewCycle) spezifizieren.

Für diesen Anwendungsfall ist die Device-Beispielapplikation "App3_IsoApp" vorgesehen. Diese verwendet spezielle Input-Output-Module aus der GSD-Datei, die nur im IRT-Betrieb zulässig sind und taktsynchron arbeiten.

Für die Realisierung der T_IO_Input- bzw. T_IO_Output-Zeitbedingungen ist es möglich, verschiedene GPIOs (1...7) zu verwenden, die mit einer parametrierbaren Verzögerung zum Zyklusbeginn (Newcycle) einen Triggerpuls ausgeben. Nach dem gleichen Konzept ist es auch möglich, Interrupt-Routinen zu installieren, die mit einer parametrierbaren Verzögerung zum Zyklusbeginn eine vom Anwender spezifizierte Callback-Routine aufrufen.

Mehr über die technologischen Zusammenhänge finden Sie in der PROFINET IO-Spezifikation /5/ sowie in der Schnittstellenbeschreibung /3/, Kapitel "Taktsynchrone Applikation mit IRT, T_Input- und T_Output-Auswertung".

Für die Projektierung verwenden Sie die Projektdatei "CPU317-2_EB200P_IRTIsoApp.zip" oder eine zu Ihrem Controller passende Projektdatei, siehe Kapitel Controller-Applikationsbeispiele (Seite 63).

Hinweis

Die Ausgabe von Konsolenmeldungen, aber auch ein angeschlossener JTAG-Debugger können das Ablauftiming der Software beeinflussen. Dies kann eventuell bei IRT-Betrieb mit kurzen Zykluszeiten (z. B. 250 µs) zu Problemen führen. In diesem Fall sollten die Konsolenmeldungen abgeschaltet bzw. der JTAG-Debugger entfernt werden.

7.2.4 Use Case Tool Calling Interface (TCI)

Controller-Applikation und CPD-Tool

Ein lauffähiges Anwenderbeispiel für TCI finden Sie auf der CD im Verzeichnis "(...)\tools\TCI". Dort ist ein CPD-Beispieltool enthalten, welches über den Communication Server von STEP 7 eine Verbindung zum PROFINET Device EB 200P aufnimmt und einen 4 byte langen Beispielrecord (Record-Index 50) auf das Device schreibt bzw. liest. Der Inhalt des 4 byte langen Record kann in einer Maske des CPD-Tools verändert werden.

Das CPD-Tool wurde mit Microsoft Visual Studio 2008 erstellt, ein entsprechendes Projekt "S7TCITest.dsp" sowie der Sourcecode liegen im Verzeichnis "\tools\TCI\CpdTool\Msvc".

Eine Anleitung für das CPD-Tool finden Sie auf der CD im Unterverzeichnis "tools\TCI\CpdTool\doc".

Auf dem PC, auf dem das CPD-Tool läuft, muss STEP 7 mit Communication Server-Funktionalität installiert und gestartet sein.

Device-Applikation

TCI ist im Standard-Applikationsbeispiel "App1_Standard" enthalten, d. h. dieses muss auf dem Device geladen und gestartet werden.

Für das TCI-Beispiel wird bei einem Write Record Request auf Record-Index 50 ein 4 byte langer Beispiel-Record gespeichert und bei einem entsprechenden Read Record Request auf diesen Index wieder zurückgelesen.

Beim Schreiben/Lesen des Record wird jeweils eine Meldung an der Konsolenschnittstelle angezeigt.

Hinweis

Um mit dem Device Tool über den Communication Server mit dem Device zu kommunizieren, ist STEP 7 V5.5 SP3 oder höher erforderlich.

Siehe auch

Controller-Applikationsbeispiele (Seite 63)

7.2.5 Use Case PROFlenergy

Device-Applikation

Die PROFlenergy-Funktionalität ist in den Anwenderbeispielen Standard-Interface ("App1_Standard") und DBA-Interface ("App2_DBAI") enthalten. Dort werden die PROFlenergy-Services gemäß Spezifikation auf Record-Schreib- und -Lesezugriffe abgebildet, die auf dem Record-Index 0x80a0 basieren. Der Ablauf ist dabei grundsätzlich folgender:

1. Es wird ein Write Record Request an das PROFINET Device mit Index 0x80a0 gesendet.
2. Die Response wird durch ein oder mehrere Read Record Requests mit Index 0x80a0 gepollt, solange, bis das Kommando ohne bzw. mit Fehlerrückmeldung beendet wurde.

Im Anwenderbeispiel (Standard-Interface und DBA-Interface) wird daher für den Write Record Request auf Index 0x80a0 die Funktion "PROFlenergy_RequestHandler()" aufgerufen. Die Daten des Requests werden in einer Datenstruktur zwischengespeichert, da diese beim folgenden Record Read auf den gleichen Index wieder benötigt werden.

Die Antwort auf den Request wird in der Funktion "PROFlenergy_ResponseHandler()" aufgebaut, welche bei einem Read Record Request mit Index 0x80a0 ausgeführt wird.

Für die einzelnen PROFlenergy-Services wird jeweils beispielhaft eine Meldung im Terminalprogramm ausgegeben. Da das Beispiel nur die Prinzipien von PROFlenergy verdeutlichen soll, wird bereits in der ersten Response "Auftrag beendet" zurückgemeldet. Eine Ablaufsimulation mittels Timer ist hier nicht enthalten, um das Beispiel einfach zu halten.

Für die Projektierung verwenden Sie die zu Ihrem Controller passende "xxx_RT.zip"-Projektdatei, siehe Kapitel Controller-Applikationsbeispiele (Seite 63).

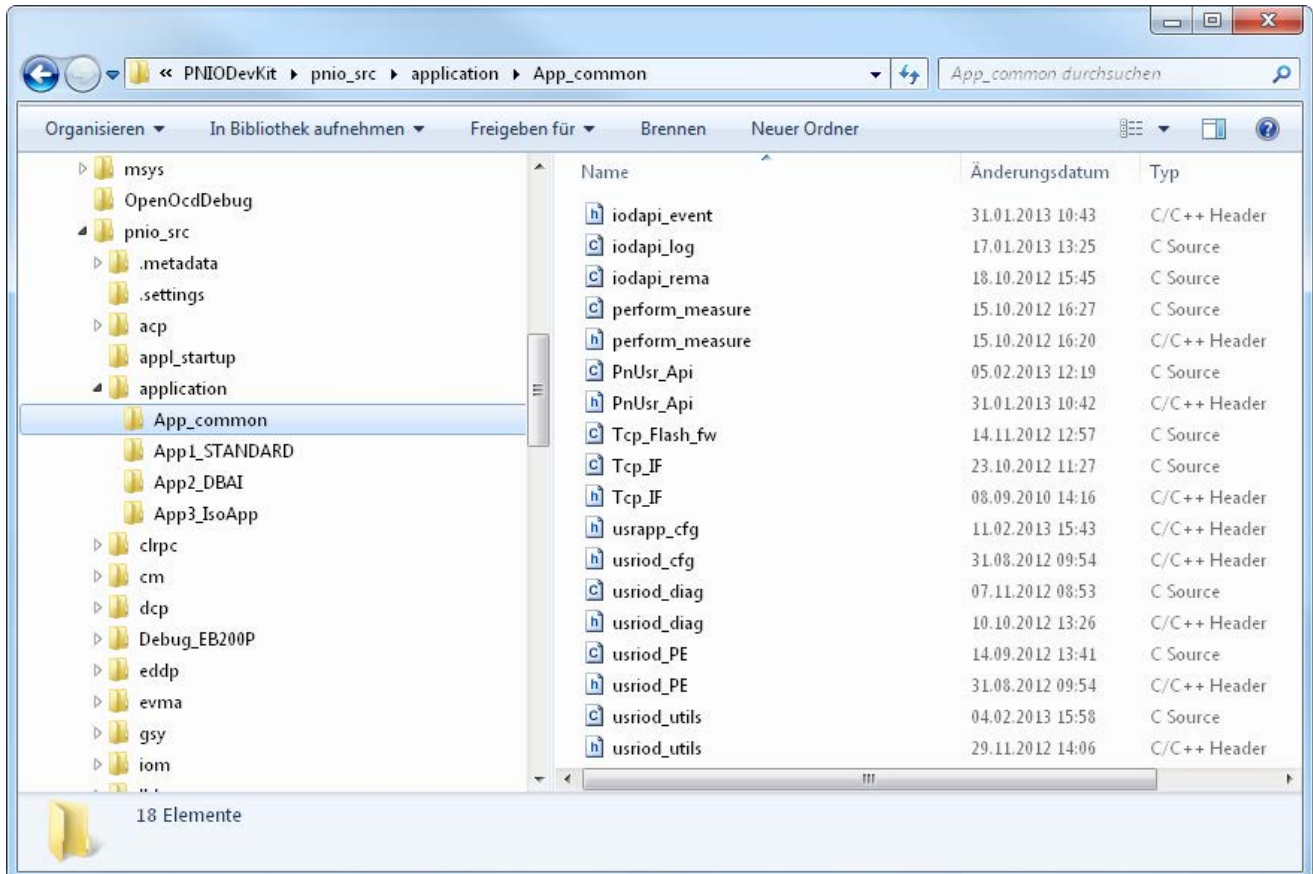


Bild 7-5 Applikationsdatei für PROFInergy-Beispiel

Die eigentliche PROFInergy-Funktionalität ist im Modul "(...)\application\usriod_pe.c" implementiert. Datenstrukturen der PROFInergy-Services sind in der Header-Datei "(...)\application\usriod_pe.h" implementiert.

Controller-Applikation

Für die Projektierung verwenden Sie die Projektdatei "CPU317-2_EB200P_RT.zip" oder eine zu Ihrem Controller passende Projektdatei, siehe Kapitel Controller-Applikationsbeispiele (Seite 63).

Für SIMATIC S7-300 sind im STEP 7-Anwenderprogramm die Funktionsbausteine FB815 (Start_Pause, End_Pause) und FB816 (offene PE-Schnittstelle für PE_QUERY_MODES, PEM_STATUS, PE_IDENTIFY, ...) enthalten. Diese werden im OB1 aufgerufen und können über ein Merkerbit gestartet werden:

FB815	START_PAUSE	Merker 210.0
FB815	END_PAUSE	Merker 210.1
FB816	OFFENE SCHNITTSTELLE	Merker 220.0

7.2.6 Use Case Medienredundanz (MRP)

Device-Applikation

Das Standard-Interfacebeispiel "App1_Standard" ist in diesem Fall zu verwenden. Die beiden Ports des EB 200P können in der Projektierung als Ringports konfiguriert werden. Die MRP-Fähigkeit wird durch einen Übergabeparameter in der Funktion der Anwenderschnittstelle "PNIO_device_open()" eingestellt. Die Aktivierung der MRP-Funktionalität erfolgt über das Engineering im Projektierungstool.

Ist MRP-Fähigkeit voreingestellt, so werden die Ports 1 und 2 als Ringports definiert, damit das Gerät als Teilnehmer des MRP-Rings im Engineering System konfiguriert werden kann. Außerdem muss dazu in der Applikation ein MRP-fähiger DAP (DAP3) in Slot 0 gesteckt sein. Dieser beinhaltet die MRP-Client-Funktionalität, aber nicht die MRP-Manager-Funktionalität.

Anhand des geladenen Beispiel-DAPs (2 oder 3) wird in der Device-Applikation automatisch die MRP-Fähigkeit eingestellt. Der DAP wird bei aktiviertem REMA im Flash (REMA-Daten) gespeichert, so dass die Software beim nächsten Anlauf (und allen weiteren) automatisch mit dem richtigen DAP hochläuft.

Für die Projektierung verwenden Sie die Projektdatei "CPU317-2_EB200P_MRP.zip" oder eine zu Ihrem Controller passende Projektdatei, siehe Kapitel Controller-Applikationsbeispiele (Seite 63).

Als Controller wird dabei eine SIMATIC CPU 317-2 PN/DP mit 2 Ethernetports verwendet, die als Ring mit einem EB 200P verbunden ist.

Hinweis

Für die MRP-Funktionalität ist eine CPU-Firmware Version 3.2 oder höher erforderlich.

Hinweis

Um einen evtl. Ringschluss beim Hochlauf eines MRP-Device sicher zu vermeiden, ist beim zweiten Ringport des EvalKit eine zusätzliche Wartezeit von 1,65 s im PROFINET-Stack implementiert, bevor an diesem Ringport der Link-Up erfolgt.

Hinweis

Medienredundanz und Fast Startup

Bei aktivierter Medienredundanz ist Fast Startup (FSU) nicht möglich. Falls außerdem der MRP-fähige DAP3 gesteckt ist und gleichzeitig FSU aktiviert werden soll, so muss der PROFINET IO-Controller beim Connect einen Anlauf-Record schicken, mit dem MRP abgeschaltet wird. Ansonsten funktioniert FSU nicht. Falls der PROFINET IO-Controller diesen "MRP off"-Record nicht schicken kann, so darf auf dem Device die MRP-Fähigkeit (siehe Funktion "PNIO_device_open()") nicht aktiviert sein, d. h. DAP2 muss in diesem Beispiel gesteckt werden.

Für diesen Anwendungsfall müssen DAP2 und 3 dieselben Eigenschaften haben, mit Ausnahme der MRP-Fähigkeit (eingeschaltet in DAP3 und ausgeschaltet in DAP2).

Falls die MRP-Fähigkeit konfiguriert wurde, aber noch keine REMA-Daten für das PDEV auf dem Device vorhanden sind, läuft dieses defaultmäßig immer mit aktiviertem MRP hoch.

Dies ist notwendig, um beim ersten Einschalten ohne vorherige Konfiguration der Teilnehmer einen logischen Ring zu vermeiden (kreisende Telegramme).

7.2.7 Use Case Shared Device

Ein Shared Device-Betrieb ist prinzipiell im RT- und IRT-Betrieb möglich. Es können aber nicht zwei IRT-Verbindungen gleichzeitig aktiviert sein, d. h. eine von beiden muss in diesem Fall als RT-Verbindung konfiguriert werden.

Für RT oder IRT mit einem Shared Device kann das Standard-Interfacebeispiel "App1_Standard" unverändert verwendet werden. Das DBAI-Beispiel hingegen kann nicht eingesetzt werden, da dort die Verwaltung mehrerer ARs nicht enthalten ist. Bei Bedarf kann der Anwender dies aber entsprechend erweitern.

Für die Projektierung verwenden Sie die Projektdatei "CPU317-2_EB200P_Shared.zip" oder eine zu Ihrer Konfiguration passende Projektdatei, siehe Kapitel Controller-Applikationsbeispiele (Seite 63).

Als Controller werden dabei zwei SIMATIC CPUs 317-2 PN/DP mit je zwei Ethernetports verwendet.

Hinweis

Für die Shared Device-Funktionalität ist eine CPU-Firmware Version 3.2 oder höher erforderlich.

Hinweis

Wenn der Sync-Master eines Shared Device (das mit IRT mit "hoher Performance" betrieben wird) ausfällt, dann kann es dazu kommen, dass das Shared Device für andere darauf zugreifende IO-Controller kurz ausfällt.

7.3 GSD-Dateien

Für das Evaluation Board EB 200P ist auf der CD im Verzeichnis "(...)\GSDML" eine GSD-Datei hinterlegt.

Die GSD-Datei enthält folgende DAPs (Device Access Points):

- DAP1 nur für alte PROFINET IO-Controller ohne PDEV
- DAP2 Standard-DAP ohne MRP, ohne Shared Device
- DAP3 Standard-DAP + MRP + Shared Device

Für hochdynamische Motion Control-Applikationen sind in der GSD-Datei außerdem spezielle IRT-Module konfiguriert, bei denen durch Vorgabe von T-Input- und T-Output-Zeiten alle Ein- und Ausgänge der Devices einer IRT-Domäne zeitgleich gelesen bzw. geschrieben werden können. Diese können ausschließlich für den IRT-Betrieb (RT Class3) konfiguriert werden. Die anderen Module können hingegen für RT und IRT (Class1, Class3) betrieben werden.

Hinweis

Es wird empfohlen, bei Erweiterungen der Funktionalität eines Gerätes die Modul-ID des betroffenen DAP gegenüber der Vorgängerversion zu verändern. Damit wird sichergestellt, dass eine Projektierung, die solche Funktionen beinhaltet, nicht unzulässigerweise auf ein Altgerät geladen werden kann.

7.4 Controller-Applikationsbeispiele

7.4.1 Controller-Applikationsbeispiele für SIMATIC S7-300 CPUs

Im Verzeichnis "SIMATIC_STEP7" auf der Evaluation Kit-CD sind für das EB 200P Beispiele für den RT- und IRT (RT Class3)-Betrieb hinterlegt.

Die folgende Tabelle zeigt, welche Controller-Beispiele mit welcher Device-Applikation betrieben werden können.

Tabelle 7- 1 Controller-Beispiele mit Device-Applikation

Dateiname	Inhalt	Device-Beispielapplikation
CPU317-2_EB200P_RT.zip	RT, CPU 317-2 PN/DP + EB 200P	App1_Standard, App2_DBAI
CPU317-2_EB200P_IRT.zip	IRT (high performance), CPU 317-2 PN/DP + EB 200P	App1_Standard, App2_DBAI
CPU317-2_EB200P_Shared	RT, 2 x CPU 317-2 PN/DP (Firmware ≥ V3.2) + EB 200P	App1_Standard
CPU317-2_EB200P_MRP	RT, CPU 317-2 PN/DP + EB 200P im redundanten Betrieb	App1_Standard
CPU317-2_EB200P_IsoApp	IRT, takt synchrone Device-Applikation	App3_IsoApp

Weitere Beispiele können in zukünftigen Versionen hinzugefügt werden.

7.5 Laden einer Applikationsfirmware in den Flash

Da die Firmware im Anlauf vom Flash ins RAM geladen und dort gestartet wird, kann ein Firmware-Update über die Applikation durchgeführt werden. Dazu ist auf dem Device eine entsprechende Funktionalität in allen Anwenderbeispielen enthalten. Dabei wird von einem PC-Programm ("TcfFwLoader.exe") zunächst über TCP eine Firmware auf das Device geladen und dort von der Applikation in den Flash programmiert.

Ein lauffähiges Firmware-Image, das für den Download notwendige PC-Programm sowie eine kurze Bedienungsanleitung ("Readme.txt") finden Sie auf der CD im Unterverzeichnis "(...)\BootableBinary".

Eigene Devices erstellen

Nachdem Sie nun einen Einblick in das Evaluation Kit ERTEC 200P bekommen haben, beschreibt dieses Kapitel in kurzer Form die notwendigen Schritte, um eine eigene PROFINET IO-Device-Applikation zu erstellen und zu testen.

1. Definieren der device-spezifischen Daten

Entsprechend der Funktion Ihres Device definieren Sie folgende Daten:

- benötigte DAPs (Direct Access Point)
- benötigte Module und Submodule (mit oder ohne Hochlaufparameter)
- user-spezifische Diagnosen/Alarmer
- user-spezifische azyklische Daten (Records lesen/schreiben)
- I&M Records

2. Anpassen der GSD-Datei (siehe dazu auch /4/)

- Allgemeine Daten anpassen:
 - Vendor ID
 - Vendor Name
 - Device ID
 - Main Family / Product Family, etc.
- Device-spezifische Daten anpassen (wie in Schritt 1 definiert):
 - DAPs
 - Module und Submodule
 - Diagnosen/Alarmer
 - Azyklische Records

3. Geänderte GSD-Datei überprüfen

- Überprüfen Sie die geänderte GSD-Datei mit dem PROFINET XML Viewer.

Hinweis

Die PROFIBUS-Nutzerorganisation stellt einen PROFINET XML Viewer mit integriertem GSD-Checker zum Download zur Verfügung. Dieser ist für Mitglieder kostenfrei.

4. Geänderte GSD-Datei im Projektierungstool installieren

- Installieren Sie Ihre GSD-Datei wie beschrieben in Kapitel Projektbeispiel für PROFINET IO-Device bearbeiten (Seite 37).

5. Erstellen und Laden der eigenen Projektierung

- Im Projektierungstool die Projektierung anpassen und übersetzen
- Geänderte Projektierung in den PROFINET IO-Controller laden

6. Device-Applikation anpassen

- Allgemeine Daten anpassen:
 - Vendor ID
 - Vendor Name
 - Device ID, etc.
- devicespezifische Daten anpassen:
 - definierte DAPs
 - definierte Module und Submodule
 - user-spezifische Diagnosen/Alarmer
 - user-spezifische Records
 - I&M Records
- Test-Kommandoliste anpassen
 - Eventuell Test-Kommandos an Ihre definierten Testfunktionen anpassen.

7. Device-Applikation übersetzen

- Eclipse (siehe Kapitel Eclipse: Build der PROFINET IO-Applikation durchführen (Seite 26))

8. Device-Applikation ins EB 200P laden und testen

- Für das Flashen der Firmware siehe Kapitel Laden einer Applikationsfirmware in den Flash (Seite 63).
- Für das Debuggen siehe Dokument "Howto_use_Amontec_JTAGkey-Tiny_on_EB200P.pdf" im installierten Unterverzeichnis "(...)\PNIODevkit\OpenOcdDebug".

Anhang

A.1 Abkürzungen/Begriffsverzeichnis

ACP	A cyctic C ommunication P rotocol, bezeichnet eines der Software-Basispakete des IO-Stacks
API	A pplication P rocess I dentifier
AR	A pplication R elationship
BSP	B oard S upport P ackage
CLRPC	C onnectionless R emote P rocedure C all, bezeichnet eines der Software-Basispakete des IO-Stacks
CM	C ontext M anagement, bezeichnet eines der Software-Basispakete des IO-Stacks
DAP	D evice A ccess P oint, spezifischer Eintrag in der GSD Datei
DBAI	D irect B uffer A ccess I nterface
DCP	D iscovery and basic C onfiguration P rotokoll, bezeichnet eines der Software-Basispakete des IO-Stacks
DK_SW	D evelopment K it S oftware (Entwicklungskit für Plattformen, basierend auf Standard-Ethernetcontrollern)
EB 200P	E valuation B oard für ERTEC 200P (Bestandteil des Evaluation Kit ERTEC 200P)
EDDI	E thernet D evice D river für IRTE switch im ERTEC 200/400 (frühere Versionen: EDD_ERTEC)
EDDP	E thernet D evice D river für PN switch im ERTEC 200P
EDDS	E thernet D evice D river für S tandard Ethernet Controller (frühere Versionen: EDD_soft)
EDD	E thernet D evice D river, Oberbegriff für EDDI, EDDP, EDSS
ELOG	E rror L ogging für Debug-Zwecke (Level 1 = nur Fehler ausgeben)
GSD	G eneric S tation D escription
GSDML	G SD M arkup L anguage
GSY	G eneric S ync Modul, bezeichnet eines der Software-Basispakete des IO-Stacks
IOCR	I nput/ O utput C ommunication R elationship
IOCS	I nput/ O utput Object C onsumer S tatus
IOPS	I nput/ O utputObject P rovider S tatus
IRT	I sochrone R eal T ime, Class 2 (IRT Class 2) oder Class 3 (IRT Class 3)
LLDP	L ink L ayer D iscovery P rotocol (IEEE 802.1AB, Allows stations to exchange chassis and port information)
LSA	L ayer S tructure A rchitecture
LW	Laufwerk
MIB	M anagement I nformation B ase. Datenbasis für SNMP Dienste

MRP	Media Redundancy Protocol
NARE	Name Address Resolution
NRT	Non Realtime ist ein Oberbegriff für alle Nicht – Realtime Telegramme (nicht Typ 0x8892)
NV	Non Volatile
OHA	Object Handler
OS	Operating System , bezeichnet hier die Abstraktionsschicht für ein beliebiges Betriebssystem, auf welches der IO-Stack portiert werden soll.
PDEV	Physical Device
PN-IO	PROFINET IO
PNO	PROFIBUS Nutzer Organisation
PCF	Polymeric Cladded Fiber (optisches Übertragungsmedium)
POF	Polymeric Optical Fiber (optisches Übertragungsmedium)
RT	Realtime ist ein Oberbegriff für azyklische und zyklische Realtime
RT Class	Realtime-Klasse gemäß PROFINET IO-Spezifikation
SI	Standard Interface
SNMP	Simple Network Management Protokoll
SOCK	UDP Socket Interface für PROFINET IO, bezeichnet eines der Software-Pakete des IO-Stacks
TAP	Test Access Port
UDP	User Datagram Protokoll
UUID	Universal Unique Identifier

A.2 Literaturverzeichnis

/1/

GSDML Specification for PROFINET IO Version 2.3 oder höher
PROFIBUS Nutzerorganisation e.V.

/2/

EB 200P Handbuch

(Manual_EB200P_V1.0.pdf)

/3/

Schnittstellenbeschreibung Evaluation Kit ERTEC 200P PN IO V4.0

(Schnittstellenbeschreibung_EvalKit_ERTEC200P_V4.0.pdf)

/4/

GSDML-Einstieg leicht gemacht V1.2

(GSDML-GettingStarted.pdf)

/5/

**PROFINET IO Application Layer Service Definition und
PROFINET IO Application Layer Protocol Specification**

(Downloadbar von der PNO Website (<http://www.profinet.com>))

