

SIEMENS

SIMATIC

Industrie-PC SIMATIC IPC Support Package for VxWorks

Bedienhandbuch

<u>Einleitung</u>	1
<u>Beschreibung</u>	2
<u>Installation auf dem Entwicklungsrechner</u>	3
<u>VxWorks-Image erzeugen</u>	4
<u>Bootloader für VxWorks</u>	5
<u>Nachladbares Kernel Modul erzeugen</u>	6
<u>Real Time Process Applikation (RTP) erstellen</u>	7
<u>Real Time Process Applikation (RTP) debuggen</u>	8
<u>Arbeiten mit der Shell</u>	9
<u>PROFINET-Treiber</u>	10
<u>PROFIBUS-Treiber</u>	11
<u>Hardwarenahe Funktionen</u>	12
<u>DiagMonitor-Agent für VxWorks</u>	13
<u>VxWorks V7</u>	14
<u>Service und Support</u>	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.

Inhaltsverzeichnis

1	Einleitung	7
2	Beschreibung	11
2.1	Überblick	11
2.2	Systemumgebung	12
2.3	Verwendung von VxWorks V7	12
3	Installation auf dem Entwicklungsrechner	13
4	VxWorks-Image erzeugen	15
4.1	BSPs für die SIMATIC IPCs auswählen	15
4.2	Komponenten für das VxWorks-Image einbinden	18
4.3	VxWorks-Image konfigurieren	27
4.3.1	Allgemeine Einstellungen für das VxWorks-Image festlegen	27
4.3.2	Bootparameter von der Festplatte lesen.....	28
4.3.3	Ethernet-Schnittstelle konfigurieren.....	28
4.3.4	COM-Schnittstelle oder VGA-Schnittstelle für die Ausgabe konfigurieren	29
4.3.5	ftp-Verbindung konfigurieren	29
4.3.6	Verbindung zum Debuggen aktivieren.....	29
4.4	Weitere nützliche Komponenten.....	30
4.5	Wichtige Hinweise zu VxWorks	30
4.6	VxWorks-Image-Erstellung starten	30
5	Bootloader für VxWorks	31
5.1	VxWorks BootApp.....	32
5.1.1	VxWorks BootApp erzeugen.....	32
5.1.2	VxWorks auf der Festplatte des SIMATIC IPC installieren.....	43
5.2	GRUB Bootloader	46
5.3	GRUB Legacy Bootloader	48
5.4	GRUB Bootloader und Windows	50
6	Nachladbares Kernel Modul erzeugen	55
7	Real Time Process Applikation (RTP) erstellen	59
8	Real Time Process Applikation (RTP) debuggen	61
9	Arbeiten mit der Shell	65
9.1	Target Shell Timeout verhindern	65
9.2	Kommandozeile editieren	65

10	PROFINET-Treiber	67
10.1	PROFINET-Funktionen	67
10.2	PROFINET-Treiber	68
10.2.1	CP 16xx-Treiber	68
10.2.2	PN-Treiber.....	69
10.3	IO-Base-Schnittstelle	69
10.4	PROFINET-Aufrufe im Downloadable Kernel Modul (DKM) nutzen.....	69
10.5	PROFINET-Aufrufe im Real Time Process (User Mode) nutzen.....	70
10.6	Meldungen des PROFINET-Treibers.....	71
10.7	PROFINET-Treiber konfigurieren	72
10.7.1	CP 16xx-Treiber konfigurieren	72
10.7.2	PN-Treiber konfigurieren.....	74
11	PROFIBUS-Treiber	75
11.1	PROFIBUS-Funktionen.....	75
11.2	DP-Base-Schnittstelle	75
11.2.1	Programmierhandbuch zur DP-Base-Schnittstelle	75
11.2.2	DP-Slave-Daten senden (Aktualisierung der Informationen im Programmierhandbuch)	75
11.3	PROFIBUS-Aufrufe im Downloadable Kernel Modul (DKM) nutzen	77
11.4	PROFIBUS-Aufrufe im Real Time Process (User Mode) nutzen	78
11.5	Meldungen des PROFIBUS-Treibers.....	78
11.6	PROFIBUS konfigurieren.....	79
11.7	Projektierung erstellen	79
11.7.1	Mit SIMATIC Manager STEP7 projektieren	80
11.7.2	Mit STEP 7 (TIA-Portal) projektieren	82
12	Hardwarenahe Funktionen	85
12.1	Hardwarenahe Funktionen im Downloadable Kernel Modul nutzen.....	85
12.2	Hardwarenahe Funktionen im Real Time Process (User Mode) nutzen	85
12.3	Meldungen des Treibers	86
12.4	Hardwarenahe Funktionen konfigurieren.....	87
12.5	Zugriffsfunktionen für hardwarenahe Funktionen	89
12.5.1	GetCPUTypeDMI	90
12.5.2	ReadTemperature	91
12.5.3	ReadFanStatus	93
12.5.4	ReadBatteryStatus.....	95
12.5.5	Betriebsstundenzähler	95
12.5.5.1	SetOpHoursCounter	96
12.5.5.2	GetOpHoursCounter	97
12.5.5.3	UpdateOpHoursCounter	98
12.5.6	Watchdog	99
12.5.6.1	StartWatchdog	99
12.5.6.2	TriggerWatchdog.....	100

12.5.7	LED	100
12.5.7.1	LED setzen	101
12.5.7.2	LED zurücksetzen	102
12.5.8	GetSMARTStatus	103
12.5.9	GetCMOSDateTime	104
12.5.10	SetCMOSDateTime	105
13	DiagMonitor-Agent für VxWorks	107
13.1	Übersicht	107
13.2	DMAPI-Schnittstelle	108
13.3	Betriebsstundenzähler des DiagMonitor-Agent	109
13.4	VxWorks-Image erstellen	109
13.5	Meldungen	115
13.6	Konfiguration	115
14	VxWorks V7	117
14.1	Installation auf dem Entwicklungsrechner	117
14.2	Image erzeugen	118
14.3	Bootloader für VxWorks	125
14.4	Nachladbares Kernel Modul erzeugen	125
14.5	Real Time Process Applikation (RTP) erzeugen	126
14.6	PROFINET-Aufrufe im Real Time Process (User Mode) nutzen	127
A	Service und Support	129
	Index	131

Einleitung

SIMATIC IPC

Das Echtzeitbetriebssystem VxWorks Runtime System V6.9 (32 Bit) und VxWorks V7 (im Kompatibilitätsmodus V6.9, 32 Bit) können Sie auf folgenden SIMATIC IPCs installieren:

- SIMATIC IPC227E
- SIMATIC IPC277E
- SIMATIC IPC427D
- SIMATIC IPC627D
- SIMATIC IPC647D
- SIMATIC IPC827D
- SIMATIC IPC847D

Themenbereiche in diesem Handbuch

- Konfiguration der hardwarenahen Funktionen:
 - CPU-Grundeinstellung
 - Temperaturüberwachung
 - Lüfterüberwachung
 - Batterieüberwachung
 - Watchdog
 - User-LEDs
 - Remanenzspeicher
 - Betriebsstundenzähler
 - SMART-Statusüberwachung
- Konfiguration der folgenden im SIMATIC IPC Support Package for VxWorks enthaltenen Komponenten:
 - PROFIBUS-Treiber für Onboard-Schnittstelle "PROFIBUS" (kompatibel mit CP 5622)
 - PROFINET-Treiber für Onboard-Schnittstelle "PROFINET" (kompatibel mit CP 1616)
 - PROFINET-Treiber für Onboard-Schnittstellen "Ethernet"
 - SIMATIC IPC DiagMonitor Agent
- Handhabung der API-Funktionen von PROFIBUS DP (DP-Base-Schnittstelle)
- Handhabung der API-Funktionen von PROFINET (IO-Base-Schnittstelle)
- Handhabung der DMAPi-Funktionen

Zielgruppe

Dieses Handbuch wendet sich an Inbetriebsetzer und Programmierer, die mit dem Echtzeitbetriebssystem VxWorks vertraut sind.

Weiterführende Informationen

Informationen, die sich mit dem generellen Aufbau und der Funktionsweise von VxWorks beschäftigen, finden Sie im VxWorks Application Programmer's Guide, VxWorks Kernel Programmer's Guide und den VxWorks API References.

Informationen über die SIMATIC IPC227E, SIMATIC IPC277E, SIMATIC IPC427D, SIMATIC IPC627D, SIMATIC IPC647D, SIMATIC IPC827D bzw. SIMATIC IPC847D entnehmen Sie der Betriebsanleitung des jeweiligen SIMATIC IPC (im Lieferumfang des SIMATIC IPC enthalten).

Generelle Informationen zu PROFINET sowie die Beschreibung der Zugriffsfunktionen finden Sie im Programmierhandbuch SIMATIC NET PROFINET "IO-Base-Anwenderprogrammierschnittstelle" (Datei "PGH_IO-Base_0.pdf"), das auf der mitgelieferten CD enthalten ist.

Generelle Informationen zu PROFIBUS sowie die Beschreibung der Zugriffsfunktionen finden Sie im Programmierhandbuch SIMATIC NET "Programmierschnittstelle DP-Base für CP 5613/CP 5614" (Datei "mn_dp-base-api_0.pdf"), das auf der mitgelieferten CD enthalten ist.

Konventionen

Bezeichnungen

- Für die Produktbezeichnung SIMATIC IPC227E wird auch die Abkürzung IPC227E verwendet.
- Für die Produktbezeichnung SIMATIC IPC277E wird auch die Abkürzung IPC277E verwendet.
- Für die Produktbezeichnung SIMATIC IPC427D wird auch die Abkürzung IPC427D verwendet.
- Für die Produktbezeichnung SIMATIC IPC627D wird auch die Abkürzung IPC627D verwendet.
- Für die Produktbezeichnung SIMATIC IPC647D wird auch die Abkürzung IPC647D verwendet.
- Für die Produktbezeichnung SIMATIC IPC827D wird auch die Abkürzung IPC827D verwendet.
- Für die Produktbezeichnung SIMATIC IPC847D wird auch die Abkürzung IPC847D verwendet.
- Für alle Geräte gemeinsam wird auch die Bezeichnung SIMATIC IPC verwendet.
- Für die PROFINET Onboard-Schnittstelle der SIMATIC IPC werden auch die Abkürzungen CP 1616, CP 1616 onboard oder CP 16xx verwendet.

- Für die Produktbezeichnung SIMATIC NET PROFIBUS CP 5622 PCI EXPRESS X1-CARD wird auch die Abkürzung CP 5622 verwendet.
- Ein PROFINET IO-System besteht aus einem PROFINET IO-Controller und seinen zugeordneten PROFINET IO-Devices.
 - Für PROFINET IO-Controller wird die Bezeichnung IO-Controller verwendet.
 - Für PROFINET IO-Device wird die Bezeichnung IO-Device verwendet.
- Für die Pfade "<installDir>\WindRiver\vxworks-6.9" oder "<installDir>\WindRiver\vxworks-7" wird auch die Abkürzung "<WIND_BASE>" verwendet.

Abbildungen

Die in diesem Handbuch gezeigten Screenshots wurden mit Wind River Workbench V3.3.6 (VxWorks V6.9) bzw. Wind River Workbench V4.0.5 (VxWorks V7) erstellt. Bei Verwendung anderer Versionen kann es zu Abweichungen von den Abbildungen kommen.

Industrial Security

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Lösungen, Maschinen, Geräten und/oder Netzwerken unterstützen. Sie sind wichtige Komponenten in einem ganzheitlichen Industrial Security-Konzept. Die Produkte und Lösungen von Siemens werden unter diesem Gesichtspunkt ständig weiterentwickelt. Siemens empfiehlt, sich unbedingt regelmäßig über Produkt-Updates zu informieren.

Für den sicheren Betrieb von Produkten und Lösungen von Siemens ist es erforderlich, geeignete Schutzmaßnahmen (z. B. Zellschutzkonzept) zu ergreifen und jede Komponente in ein ganzheitliches Industrial Security-Konzept zu integrieren, das dem aktuellen Stand der Technik entspricht. Dabei sind auch eingesetzte Produkte von anderen Herstellern zu berücksichtigen. Weitergehende Informationen über Industrial Security finden Sie unter (<http://www.siemens.de/industrialsecurity>).

Um stets über Produkt-Updates informiert zu sein, melden Sie sich für unseren produktspezifischen Newsletter an. Weitere Informationen hierzu finden Sie unter (http://www.siemens.de/automation/csi_de_WW).

Disclaimer für Fremdsoftware-Updates

Dieses Produkt beinhaltet Fremdsoftware. Für Updates/Patches an der Fremdsoftware übernimmt die Siemens AG die Gewährleistung nur, soweit diese im Rahmen eines Siemens Software Update Servicevertrags verteilt oder von der Siemens AG offiziell freigegeben wurden. Andernfalls erfolgen Updates/Patches auf eigene Verantwortung. Mehr Informationen rund um unser Software Update Service Angebot erhalten Sie im Internet unter Software Update Service (<http://www.automation.siemens.com/mcms/automation-software/de/software-update-service/>).

Hinweise zur Absicherung von Administrator-Accounts

Einem Benutzer mit Administratorrechten stehen an dem System weitreichende Zugriffs- und Manipulationsmöglichkeiten zur Verfügung.

Achten Sie daher auf eine angemessene Absicherung der Administrator-Accounts, um unberechtigte Veränderungen zu verhindern. Verwenden Sie dazu sichere Passwörter und nutzen einen Standard-Benutzer-Account für den regulären Betrieb. Weitere Maßnahmen wie beispielsweise der Einsatz von Security-Richtlinien sind nach Bedarf anzuwenden.

Beschreibung

2.1 Überblick

Das SIMATIC IPC Support Package for VxWorks unterstützt die zusätzlichen Hardwareschnittstellen der SIMATIC IPCs, welche von VxWorks nicht unterstützt werden wie z. B. PROFIBUS, PROFINET und hardwarenahe Funktionen (z. B. Temperatur auslesen). Das Support Package enthält die notwendigen Treiber und Anwendungsbeispiele, um eigene Kundenanwendungen sowohl im Kernel- wie auch im RTP-Mode zu erstellen.

Aus den verschiedenen Wind River Produktvarianten des Echtzeitbetriebssystems VxWorks wird die "Platform for Industrial Devices" eingesetzt. Daraus wird das VxWorks Runtime System V6.9 (32 Bit) als Laufzeitumgebung und die Workbench V3.3 als Entwicklungsumgebung oder das VxWorks Runtime System V7 (32 Bit) als Laufzeitumgebung und die Workbench V4 als Entwicklungsumgebung verwendet.

Um die Funktionalitäten nutzen zu können, wird das SIMATIC IPC Support Package for VxWorks auf dem Entwicklungsrechner installiert. Anschließend kann das Zielsystem in Betrieb genommen werden. Wichtige Grundeinstellungen müssen auf dem Entwicklungsrechner vorbereitet und auf das Zielsystem übertragen werden.

Für die PROFIBUS-, PROFINET- und hardwarenahen Funktionen sind im SIMATIC IPC Support Package for VxWorks Online-Hilfe und Code Completion enthalten.

2.2 Systemumgebung

Entwicklungssystem

Komponente	Voraussetzungen
Hardware	PC
Betriebssystem	Windows 7 (32 Bit oder 64 Bit)
Projektierungssoftware für PROFIBUS DP	STEP 7 ab Version 5.5 SP3 Hotfix 3 (optional) STEP 7 (TIA-Portal) ab V13 (optional) NCM PC ab Version 5.5 SP3 aus Developer Kit DK-16xx PN IO (optional)
Projektierungssoftware für PROFINET	STEP 7 ab Version 5.5 SP3 mit Hardware Support Package HSP 1084 und 1085 (CP 1616 onboard V2.6) (optional) STEP 7 (TIA-Portal) ab V13 (optional) NCM PC ab Version 5.5 SP3 aus Developer Kit DK-16xx PN IO (optional)
Entwicklungswerkzeuge	Workbench V3.3 für VxWorks V6.9 Workbench V4 für VxWorks V7

Die Projektierungssoftware zum Herunterladen finden Sie unter: Service und Support (Seite 129).

Zielsystem

Komponente	Voraussetzungen
Hardware	SIMATIC IPC227E bzw. SIMATIC IPC277E bzw. SIMATIC IPC427D bzw. SIMATIC IPC627D bzw. SIMATIC IPC647D bzw. SIMATIC IPC827D bzw. SIMATIC IPC847D
Betriebssystem	ab VxWorks V6.9 (32 Bit) ab VxWorks V7 im Kompatibilitätsmodus V6.9 (32 Bit)

2.3 Verwendung von VxWorks V7

Die nachfolgenden Kapitel gelten für VxWorks V6.9. Die Besonderheiten von VxWorks V7 sind in folgendem Kapitel beschrieben: VxWorks V7 (Seite 117).

Installation auf dem Entwicklungsrechner

Vorgehensweise

Um das SIMATIC IPC Support Package für VxWorks V6.9 zu installieren, gehen Sie wie folgt vor:

1. Legen Sie die Produkt-CD in das Laufwerk Ihres Entwicklungsrechners ein.
2. Starten Sie den Dateimanager (z. B. Explorer).
3. Rufen Sie auf der Produkt-CD im Ordner "IPC_Support_Package_for_VxWorks_6.9" das Programm "setup.exe" auf und folgen Sie den Anweisungen.

Ergebnis

Die Installation ist abgeschlossen.

Ablage der Dateien im Dateisystem

Nach der Installation sind die Dateien in dem folgenden Verzeichnisbaum abgelegt:

```
(<WIND_BASE>\target\3rdparty\siemens)
|
+---docs
|      *.pdf           Documentation for Siemens components
|
+---examples
|      *.*            Examples for Siemens components
|
+---h
|      *.h           Header files for Siemens components
|
+---lib
|      lib*.a        Library for Siemens components
|
\--- <additional directories and files>
```

Beachten Sie die Readme-Datei mit wichtigen Informationen zum SIMATIC IPC Support Package for VxWorks.

Die Readme-Datei befindet sich im Hauptverzeichnis der CD und im Verzeichnis "<WIND_BASE>\target\3rdparty\siemens".

VxWorks-Image erzeugen

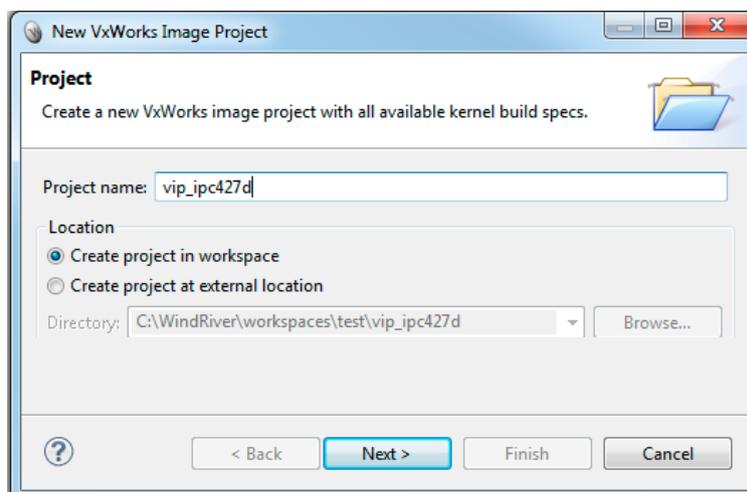
4.1 BSPs für die SIMATIC IPCs auswählen

Folgende SIMATIC IPCs sind für folgende BSPs von VxWorks geeignet:

SIMATIC IPC	BSP von VxWorks	Board bundle
IPC427D	itl_sandybridge	Intel Emerald Lake II
IPC227E, IPC277E, IPC627D, IPC827D, IPC647D, IPC847D	itl_haswell	Intel Flathead Creek
Alle SIMATIC IPCs mit Intel-Baugruppen	itl_x86	-

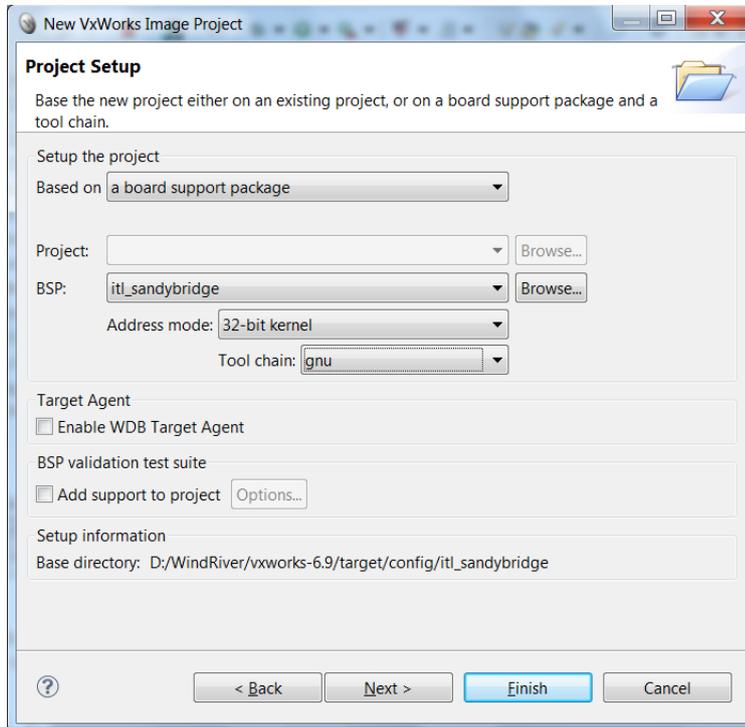
Neues VxWorks-Image Project anlegen

- Öffnen Sie die Wind River Workbench.
Sie finden diese im Windows Startmenü unter "Start > All programs > Wind River > Workbench".
- Wählen Sie einen Workspace, in dem Sie Ihr VxWorks-Image Project erstellen möchten.
- Wählen Sie "File > New > Project".
- Öffnen Sie im folgenden Dialogfenster im Listenfeld den Ordner "VxWorks 6.x" und wählen Sie dann die Option "VxWorks Image Project".
- Klicken Sie "Next >".
- Geben Sie im folgenden Dialogfenster einen "Project name" (z. B. "vip_ipc427d") ein und wählen Sie die Option "Create project in workspace".

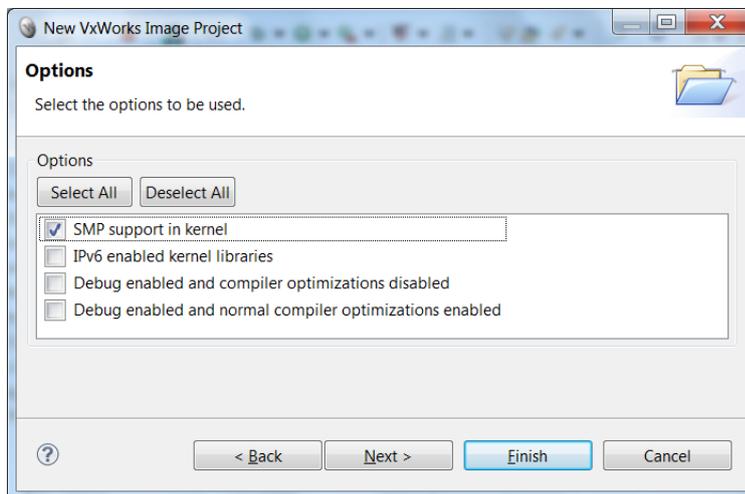


- Klicken Sie "Next >".

8. Wählen Sie das BSP und die Tool-Kette aus.
 - Wählen Sie unter BSP den passenden Eintrag zu Ihrem SIMATIC IPC aus (z. B. "itl_sandybridge").
 - Wählen Sie unter "Tool chain" den Eintrag "gnu" aus.
 - Wenn das Zielsystem mit der Workbench kommunizieren soll (für Debug-Zwecke), aktivieren Sie die Option "Enable WDB Target Agent". Aus sicherheitstechnischen Gründen (unbefugter Zugriff von extern) sollte der WDB Target Agent nur während der Entwicklung aktiviert werden.

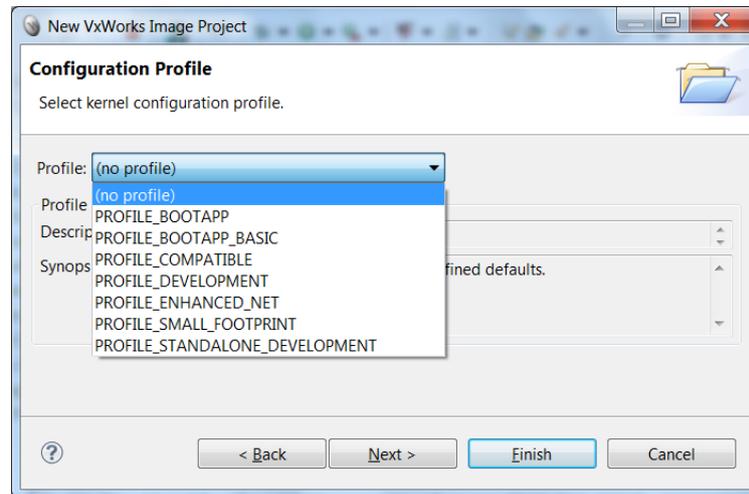


9. Aktivieren Sie die Multiprozessor-Unterstützung (SMP).



10. Wählen Sie das Profil aus.

Die Workbench bietet beim Erzeugen eines VxWorks-Image verschiedene Profile an. Abhängig vom Profil werden bestimmte Komponenten automatisch eingebunden. Wird z. B. das Profil "PROFILE_STANDALONE_DEVELOPMENT" verwendet, so liegt die Symbolverwaltung auf dem Target. Im folgenden Beispiel wird "no profile" ausgewählt.



11. Überspringen Sie den nachfolgenden Dialog "Indexer".

12. Klicken Sie "Finish".

4.2 Komponenten für das VxWorks-Image einbinden

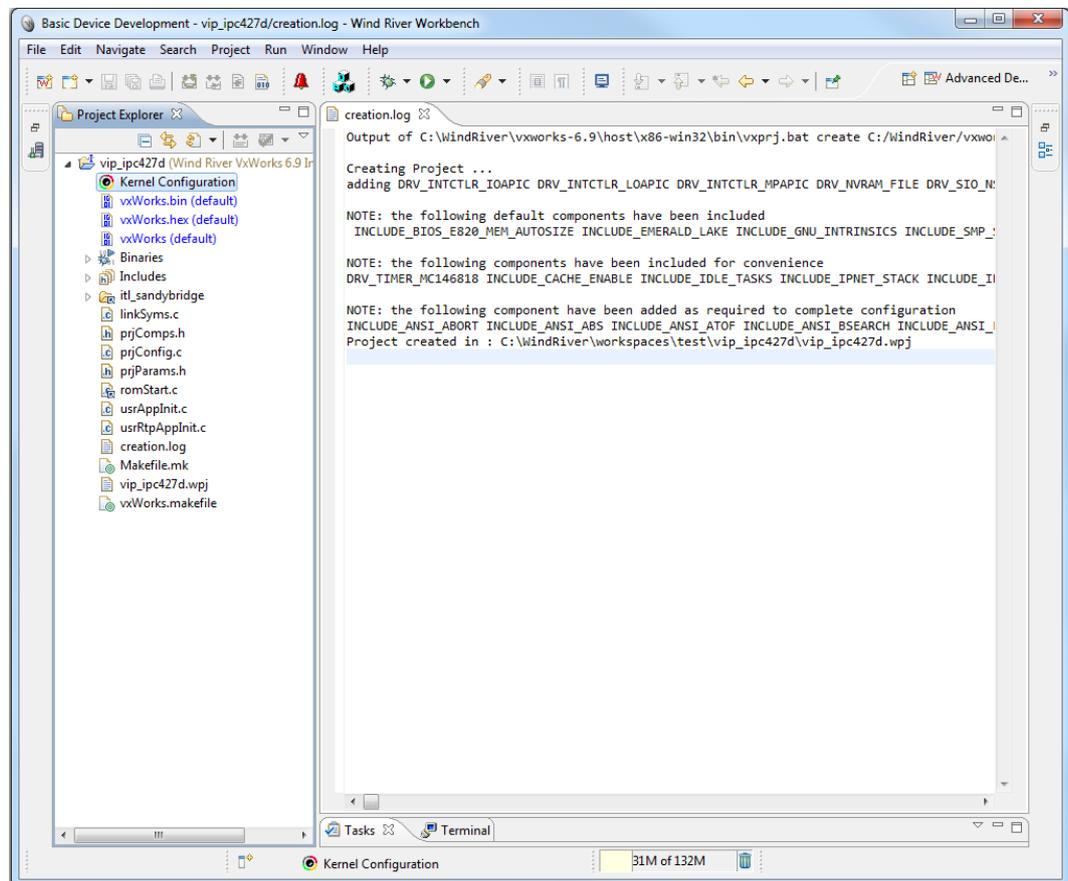
In dem ausgewählten BSP werden bestimmte Module hinzugefügt, konfiguriert oder entfernt. Dies hängt auch von dem verwendeten Bootloader ab.

Ausgehend von "no profile" und kein "WDB Target Agent" sind folgende Einstellungen notwendig.

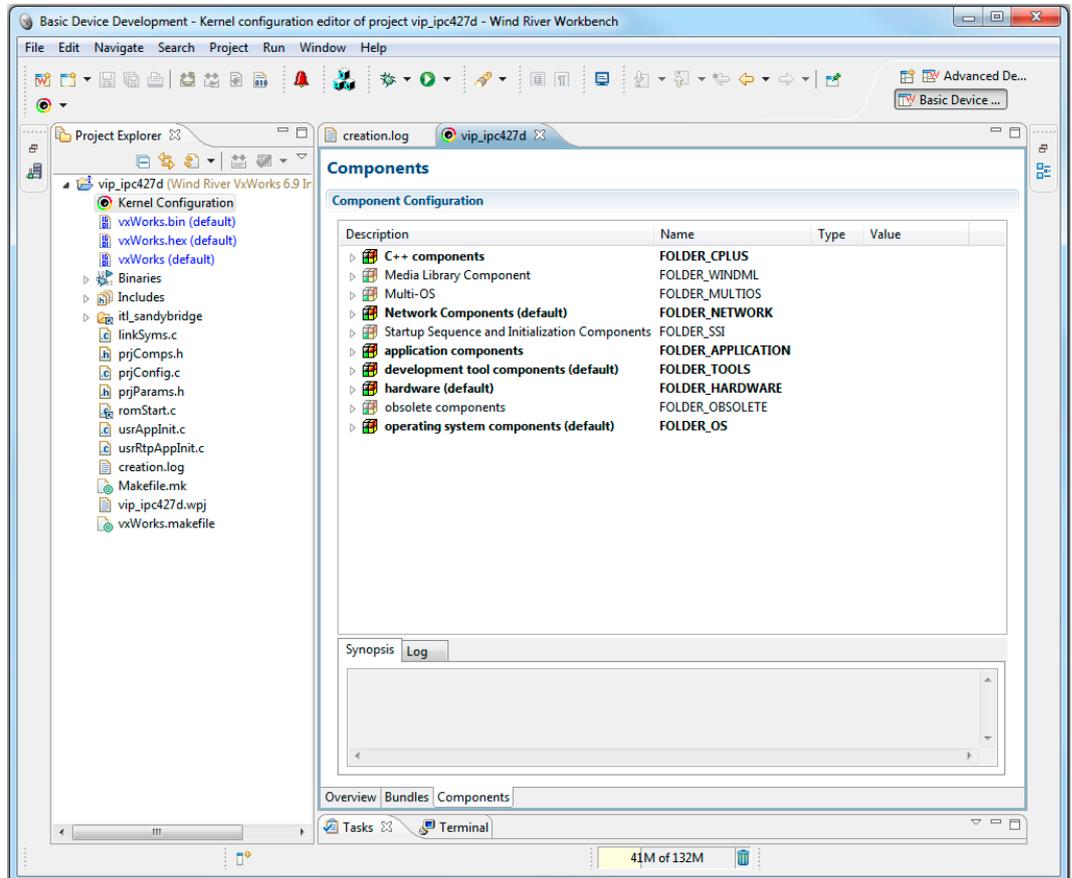
Die Schritte in dem folgenden Beispiel gelten für den IPC427D mit dem BSP "itl_sandybridge". Andere IPCs können abweichende Schritte erfordern.

Board Bundle einbinden

1. Wählen Sie die Projektansicht.
2. Doppelklicken Sie in der Baumstruktur auf "Kernel Configuration".

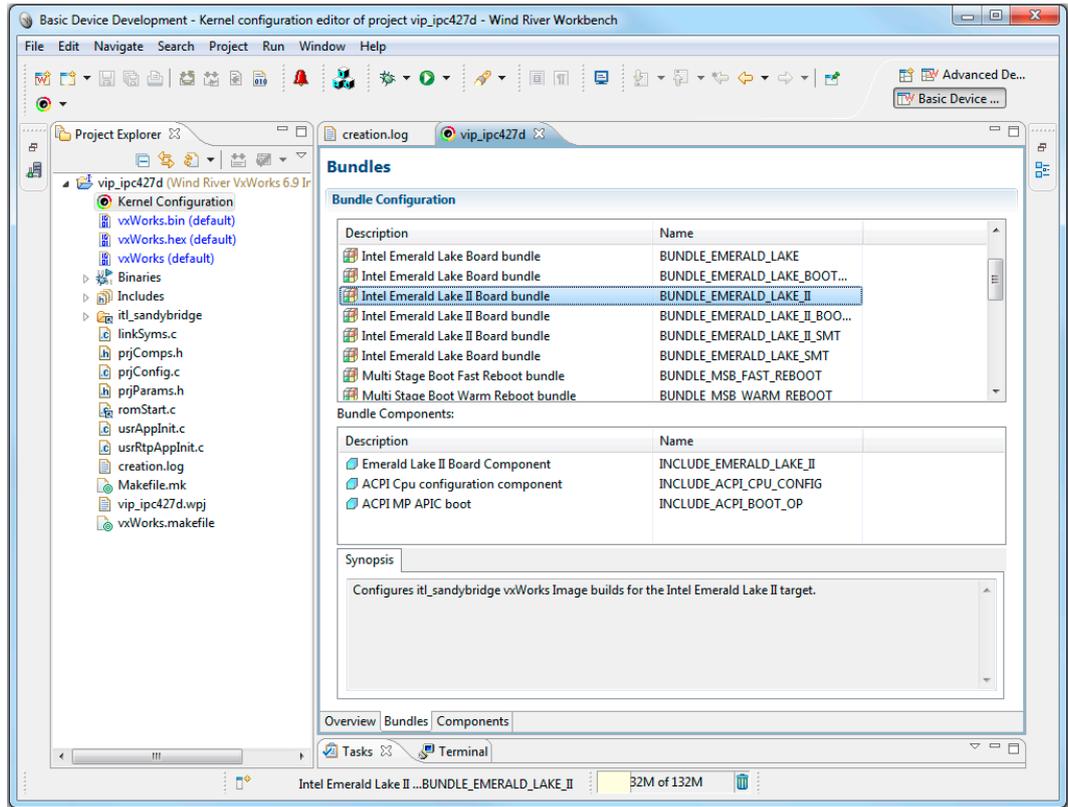


Der Bereich zur "Component Configuration" wird geöffnet. In diesem Bereich sind die verschiedenen Komponenten in Gruppen (Folder) angeordnet. Mit Doppelklick auf die Gruppe werden die Unterkomponenten der Gruppe angezeigt.



3. Wählen Sie das Register "Bundles".
4. Klicken Sie mit der rechten Maustaste auf das für Ihren IPC passende Bundle, in diesem Beispiel auf "Intel Emerald Lake II Board bundle". Klicken Sie im Kontextmenü auf "Add".

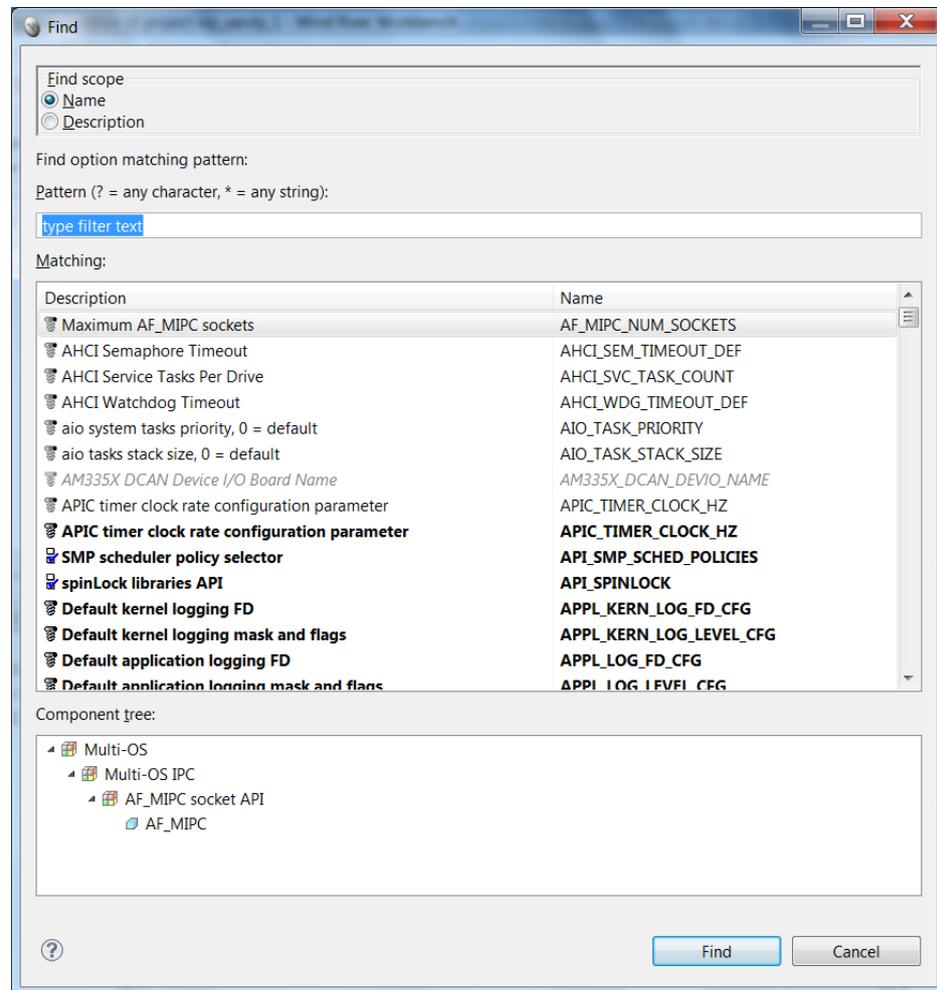
Durch Hinzufügen dieses Bundles werden automatisch die Komponenten "Emerald Lake II Board Component", "ACPI Cpu configuration component" und "ACPI MP APIC boot" eingebunden.



CP 16xx-Treiber einbinden

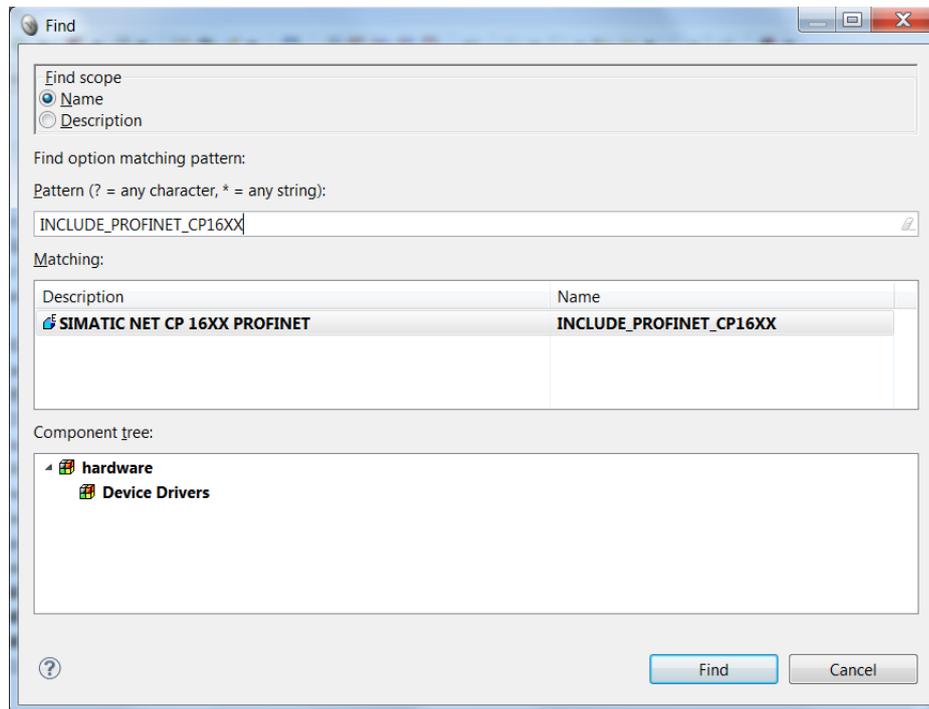
1. Wählen Sie im Register "Components" weitere Komponenten aus.

Mit "Strg + F" öffnen Sie einen Dialog zur Suche nach bestimmten Komponenten. Unter "Find scope" wählen Sie die Filter "Name" oder "Description" aus.



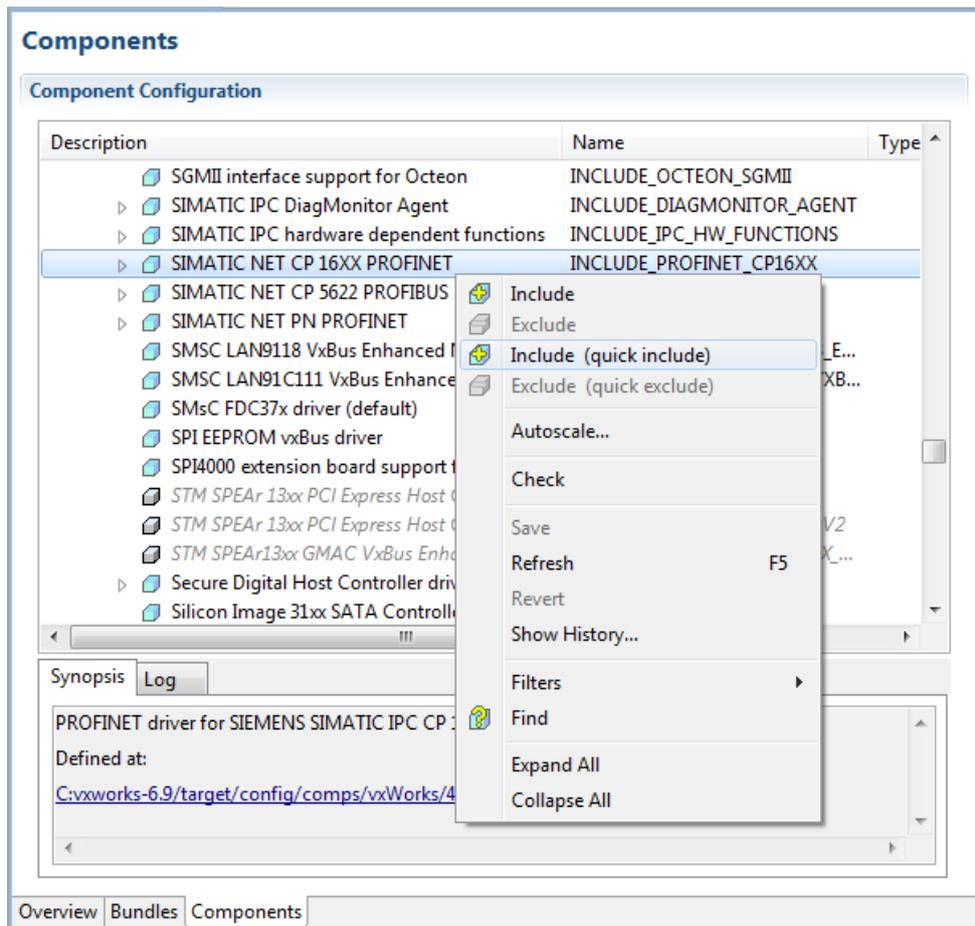
2. Wählen Sie unter "Find scope" den Filter "Name" aus.

- Um nach dem CP 16xx-Treiber für PROFINET zu suchen, geben Sie im Eingabefeld "Pattern" den Suchtext "INCLUDE_PROFINET_CP16XX" ein.

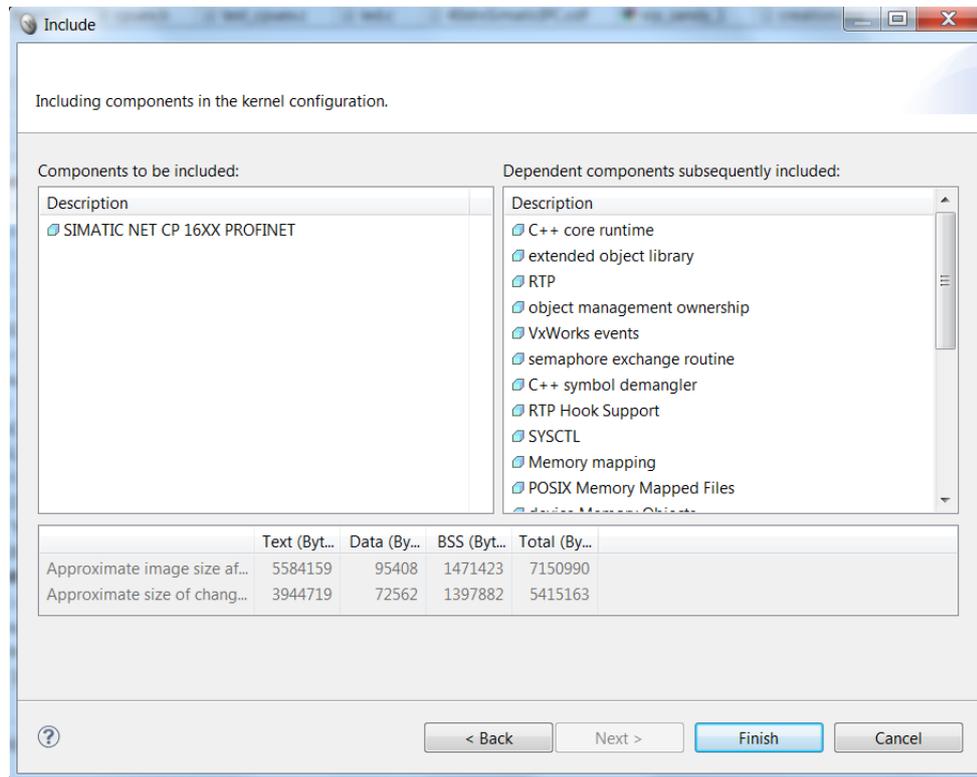


- Klicken Sie auf "Find".

5. Klicken Sie mit der rechten Maustaste auf den ausgewählten Treiber und wählen Sie "Include".



6. Der Treiber wird eingebunden. Dabei werden alle vom Treiber benötigten Komponenten automatisch mit ausgewählt.



7. Binden Sie die anderen Treiber des SIMATIC IPC Support Package for VxWorks auf die gleiche Weise ein.

Beachten Sie, dass Sie nur einen der beiden PROFINET-Treiber einbinden können. Weitere Informationen zu den PROFINET-Treibern finden Sie im Kapitel: PROFINET-Funktionen (Seite 67).

- SIMATIC IPC DiagMonitor Agent
- SIMATIC IPC hardware dependent functions
- SIMATIC NET CP 5622 PROFIBUS
- SIMATIC NET CP 16XX PROFINET
- SIMATIC NET PN PROFINET

Weitere Komponeten einbinden

Binden Sie, wie oben beschrieben, weitere gewünschte Komponenten ein.

Für den Zugriff auf das Filesystem (AHCI und USB), TCP/IP und eine Shell binden Sie folgende Komponenten aus dem VxWorks-Standardbaum ein.

Komponenten

In der nachfolgenden Liste sind Komponenten, die automatisch eingebunden werden, und Komponenten, die ausgewählt werden müssen, mit folgenden Symbolen gekennzeichnet:

- Komponenten, die automatisch eingebunden werden
- Komponenten, die manuell ausgewählt werden müssen
- Multi-stage warm reboot type (**INCLUDE_MULTI_STAGE_WARM_REBOOT**)
 - Multi-stage boot support (**INCLUDE_MULTI_STAGE_BOOT**)
- AHCI SATA Controller (**INCLUDE_DRV_STORAGE_AHCI**)
 - File System Event Utilities (**INCLUDE_FS_EVENT_UTIL**)
 - Raw filesystem (**INCLUDE_RAWFS**)
 - Event Reporting Framework (**INCLUDE_ERF**)
 - File System Monitor (**INCLUDE_FS_MONITOR**)
 - Extended Block Device (**INCLUDE_XBD**)
 - Device Manager (**INCLUDE_DEVICE_MANAGER**)
- PC Console (**INCLUDE_PC_CONSOLE**)
 - Motorola 6845 VGA driver (**DRV_VGA_M6845**)
 - Intel 8042 keyboard driver (**DRV_KBD_I8042**)
- EHCI Init (**INCLUDE_EHCI_INIT**)
 - USB Common Stack (**INCLUDE_USB**)
 - USB Common Stack Init (**INCLUDE_USB_INIT**)
 - EHCI (**INCLUDE_EHCI**)
 - USB Host Controller Start (**INCLUDE_HCD_BUS**)
- USB GEN2 Mass Storage Init (**INCLUDE_USB_GEN2_STORAGE_INIT**)
 - XBD Block Device (**INCLUDE_XBD_BLK_DEV**)
 - USB Host Class Driver Init (**INCLUDE_USB_HOST_CLASS_INIT**)
 - USB GEN2 Helper Init (**INCLUDE_USB_GEN2_HELPER**)
 - USB GEN2 Mass Storage (**INCLUDE_USB_GEN2_STORAGE**)
- dosFs File System Components (dosFs2) (**FOLDER_DOSFS2**)
 - DOS File System VFAT Directory Handler (**INCLUDE_DOSFS_DIR_VFAT**)
 - DOS File System FAT12/16/32 Handler (default) (**INCLUDE_DOSFS_FAT**)
 - Dos FS BIO buffer size (default) (**INCLUDE_DOSFS_VOL_BIO_BUFFER_SIZE**)
 - Dos FS Show Routines (default) (**INCLUDE_DOSFS_SHOW**)
 - DosFs File System Main Module (dosFs2) (default) (**INCLUDE_DOSFS_MAIN**)
 - Print message level (default) (**INCLUDE_DOSFS_PRTMSG_LEVEL**)
- Kernel shell components (**FOLDER_SHELL**)
 - Extended object library(**INCLUDE_OBJ_OPEN**)
 - RTP (**INCLUDE_RTP**)
 - Memory show routine (**INCLUDE_MEM_SHOW**)
 - Vm library show routine(**INCLUDE_VM_SHOW**)
 - Task hook show routine (**INCLUDE_TASK_HOOKS_SHOW**)

- Task show routine (**INCLUDE_TASK_SHOW**)
- Tip serial line connection utility (**INCLUDE_TIP**)
- Object management ownership (**INCLUDE_OBJ_OWNERSHIP**)
- VxWorks events (**INCLUDE_VXEVENTS**)
- Semaphore exchange routine (**INCLUDE_SEM_EXCHANGE**)
- C++ symbol demangler (**INCLUDE_CPLUS_DEMANGLER**)
- File System and Disk Utilities (**INCLUDE_DISK_UTIL**)
- Pseudo terminal driver (**INCLUDE_PTYDRV**)
- Target loader (**INCLUDE_LOADER**)
- RTP Hook support (**INCLUDE_RTP_HOOKS**)
- SYSCTL (**INCLUDE_SYSCTL**)
- Memory mapping (**INCLUDE_MMAP**)
- Host/target breakpoint synchronization (**INCLUDE_WDB_BP_SYNC**)
- WDB RTP support (**INCLUDE_WDB_RTP**)
- WDB RTP breakpoints (**INCLUDE_WDB_RTP_BP**)
- WDB RTP control support (**INCLUDE_WDB_RTP_CONTROL**)
- Debugging facilities (**INCLUDE_DEBUG**)
- Host/target modules and symbols synchronization (**INCLUDE_WDB_MDL_SYM_SYNC**)
- RTP Show (**INCLUDE_RTP_SHOW**)
- Built-in symbol table (**INCLUDE_STANDALONE_SYM_TBL**)
- Target unloader (**INCLUDE_UNLOADER**)
- Symbol table show routines (**INCLUDE_SYM_TBL_SHOW**)
- Socket API System Call support (**INCLUDE_SC_SOCKETLIB**)
- Target symbol table (**INCLUDE_SYM_TBL**)
- System symbol table initialization (**INCLUDE_SYM_TBL_INIT**)
- NETWORK SYSCTL (**INCLUDE_NET_SYSCTL**)
- System Call Hook Support (**INCLUDE_SYSCALL_HOOKS**)
- System Address Space Allocator (**INCLUDE_ADR_SPACE_LIB**)
- Address Space Allocator Show Routines (**INCLUDE_ADR_SPACE_SHOW**)
- Debug shell commands (**INCLUDE_DEBUG_SHELL_CMD**)
- Virtual memory show shell commands (**INCLUDE_VM_SHOW_SHELL_CMD**)
- Address space shell commands (**INCLUDE_ADR_SPACE_SHELL_CMD**)
- Target loader shell command (**INCLUDE_MODULE_SHELL_CMD**)
- Unloader shell command (**INCLUDE_UNLOADER_SHELL_CMD**)
- Coprocessor show routine (**INCLUDE_COPROCESSOR_SHOW**)
- IPNet sysctl integration (**INCLUDE_IPNET_SYSCTL**)
- Host table sysctl support (**INCLUDE_HOST_TBL_SYSCTL**)

- Remote Command sysctl support (**INCLUDE_REMLIB_SYSCTL**)
 - C line interpreter (default) (**INCLUDE_SHELL_INTERP_C**)
 - Command line interpreter (**INCLUDE_SHELL_INTERP_CMD**)
 - File system shell commands (**INCLUDE_DISK_UTIL_SHELL_CMD**)
 - Process shell commands (**INCLUDE_RTP_SHELL_CMD**)
 - Process show shell commands (**INCLUDE_RTP_SHOW_SHELL_CMD**)
 - Serial line connection commands (**INCLUDE_TIP_CMD**)
 - Shell banner (default) (**INCLUDE_SHELL_BANNER**)
 - Symbol shell commands (**INCLUDE_SYM_SHELL_CMD**)
 - Target-resident kernel shell (default) (**INCLUDE_SHELL**)
 - Task shell commands (**INCLUDE_TASK_SHELL_CMD**)
- USB GEN2 keyboard attaching to vxWorks Shell (**INCLUDE_USB_GEN2_KEYBOARD_SHELL_ATTACH**)
 - USB GEN2 Keyboard (**INCLUDE_USB_GEN2_KEYBOARD**)
 - USB GEN2 Keyboard Init (**INCLUDE_USB_GEN2_KEYBOARD_INIT**)
- PING Components (**FOLDER_PING**)
 - Getopt function (**INCLUDE_GETOPT**)
 - Ping wrapper (**INCLUDE_IPWRAP_PING**)
 - IPCOM shell command interface (**INCLUDE_IPCOM_SHELL_CMD**)
 - IPCOM ping commands (**INCLUDE_IPPING_CMD**)
 - PING client (**INCLUDE_PING**)

4.3 VxWorks-Image konfigurieren

Zur Konfiguration des VxWorks-Image öffnen Sie mit "Strg + F" einen Dialog zur Suche und nehmen Sie dann die folgenden Einstellungen vor.

4.3.1 Allgemeine Einstellungen für das VxWorks-Image festlegen

Timertick

Stellen Sie den Timertick auf 1000 ein, d. h. Tickintervall ist dann 1 Millisekunde.
Der Timertick wird bei "SYS_CLK_RATE" eingestellt.

SignOn Meldungen (Banner) der Shell

Sie können die SignOn Meldungen (Banner) der Shell durch Ausschließen dieser Komponente entfernen.

Wählen Sie bei "INCLUDE_SHELL_BANNER" den Wert "exclude".

Anzahl der Cores

Stellen Sie die Anzahl der Cores mit "VX_SMP_NUM_CPUS" ein. Die Anzahl der Cores hängt von der installierten CPU und den BIOS Einstellungen Ihres IPC ab.

4.3.2 Bootparameter von der Festplatte lesen

Wenn Sie VxWorks von einer Festplatte starten (bootline wird von /ataX:X/nvram.txt gelesen), benötigen Sie folgende zusätzliche Komponenten:

- AHCI warm start device component (**INCLUDE_SYS_WARM_AHCI**)
 - DOS filesystem backward-compatibility (**INCLUDE_DOSFS**)
 - DOS File System Volume Formatter Module (**INCLUDE_DOSFS_FMT**)
 - DOS File System Consistency Checker (**INCLUDE_DOSFS_CHKDSK**)
 - DOS File System Old Directory Format Handler (**INCLUDE_DOSFS_DIR_FIXED**)
- Driver for file-based non-volatile RAM support (**DRV_NVRAM_FILE**)

Für die Komponente "INCLUDE_SYS_WARM_AHCI" sind folgende Einstellungen notwendig:

Parameter	Wert
BOOTROM_DIR ¹	"/ata0:1"
NV_RAM_SIZE	(0x1000)
SYS_WARM_TYPE	SYS_WARM_AHCI

¹ Der Parameter BOOTROM_DIR gibt die Festplatte an, von der die Datei "nvram.txt" mit dem Bootparametern gelesen wird. Überprüfen Sie, ob Sie diesen Wert anpassen müssen.

4.3.3 Ethernet-Schnittstelle konfigurieren

Stellen Sie die IP-Adresse mit "INCLUDE_BSP_MACROS" > "DEFAULT_BOOT_LINE" ein (die IP-Adresse des Target ist "192.168.1.62"):

DEFAULT_BOOT_LINE: "fs(0,0)host:/ata0:1/vxWorks e=192.168.1.62 o=gei0 tn=IPC427D"

Hinweis: Für den Parameter "tn" (target name) wurde beispielhaft "IPC427D" angegeben.

Zweite Ethernet-Schnittstelle

Bei einer zweiten Ethernet-Schnittstelle benötigen Sie zusätzlich folgende Komponente:

- Interface #2 configuration (**INCLUDE_IPNET_IFCONFIG_2**)

IFCONFIG_2	"ifname gei1","devname gei1","inet 192.168.2.62", "gateway driver"
------------	--

4.3.4 COM-Schnittstelle oder VGA-Schnittstelle für die Ausgabe konfigurieren

Konfigurieren Sie die VGA-Schnittstelle wie folgt:

- Fügen Sie "INCLUDE_PC_CONSOLE" hinzu.

Konfigurieren Sie die COM-Schnittstelle wie folgt:

- Entfernen Sie "INCLUDE_PC_CONSOLE".
- Entfernen Sie "INCLUDE_USB_GEN2_KEYBOARD_SHELL_ATTACH".
- Fügen Sie "INCLUDE_SIO" hinzu.
 - Stellen Sie die Baudrate der COM-Schnittstelle mit "CONSOLE_BAUD_RATE" ein (z. B. 115200).
 - Die zu verwendende COM-Schnittstelle wählen Sie aus mit "CONSOLE_TTY" (COM1 = 0, COM2 = 1).

4.3.5 ftp-Verbindung konfigurieren

Binden Sie für die FTP-Verbindung folgendes Include ein: FTP Server (**INCLUDE_IPFTPS**)

Stellen Sie folgende 2 Parameter ein (z. B. für die Festplatte):

FTPS_INITIAL_DIR	"/ata0:1"
FTPS_ROOT_DIR	"/ata0:1"

Ergebnis

Sie können sich mit dem Server im VxWorks verbinden. Ein Benutzername wird dabei nicht ausgewertet. Sie können sich mit beliebigem Benutzernamen und ohne Passwort anmelden (z. B. ftp://192.168.1.62).

Hinweis

Im Produktivbetrieb sollte ein Benutzername und ein Passwort vergeben werden.

4.3.6 Verbindung zum Debuggen aktivieren

Sie können die TCP/IP Debug Verbindung zu Workbench aktivieren. Nehmen Sie folgende Einstellung vor:

- WDB is always enabled (**INCLUDE_WDB_ALWAYS_ENABLED**)

Zum Debuggen von Anwendungen, die als "Real Time Process Project (RTP)" erzeugt wurden, wählen Sie "INCLUDE_WDB_TSFS".

4.4 Weitere nützliche Komponenten

- Spy CPU activity commands (**INCLUDE_SPY_SHELL_CMD**)
 - Auxiliary clock (**INCLUDE_AUX_CLK**)
 - Spy (**INCLUDE_SPY**)
 - vxBus Aux Clk Support (**INCLUDE_VXB_AUX_CLK**)
- IPCOM ifconfig commands (**INCLUDE_IPIFCONFIG_CMD**)
 - IPCOM shell command interface (**INCLUDE_IPCOM_SHELL_CMD**)

4.5 Wichtige Hinweise zu VxWorks

Hinweis

Wichtige Hinweise und Informationen zu Einschränkungen von VxWorks finden Sie in der Readme-Datei.

Die Readme-Datei finden Sie im Hauptverzeichnis der CD oder im Installationsverzeichnis des SIMATIC IPC Support Package for VxWorks.

4.6 VxWorks-Image-Erstellung starten

Sie starten die Erstellung des VxWorks-Image über "Project > Build Project".

Bootloader für VxWorks

Es können verschiedene Bootloader verwendet werden. Nachfolgend werden Bootloader beschrieben:

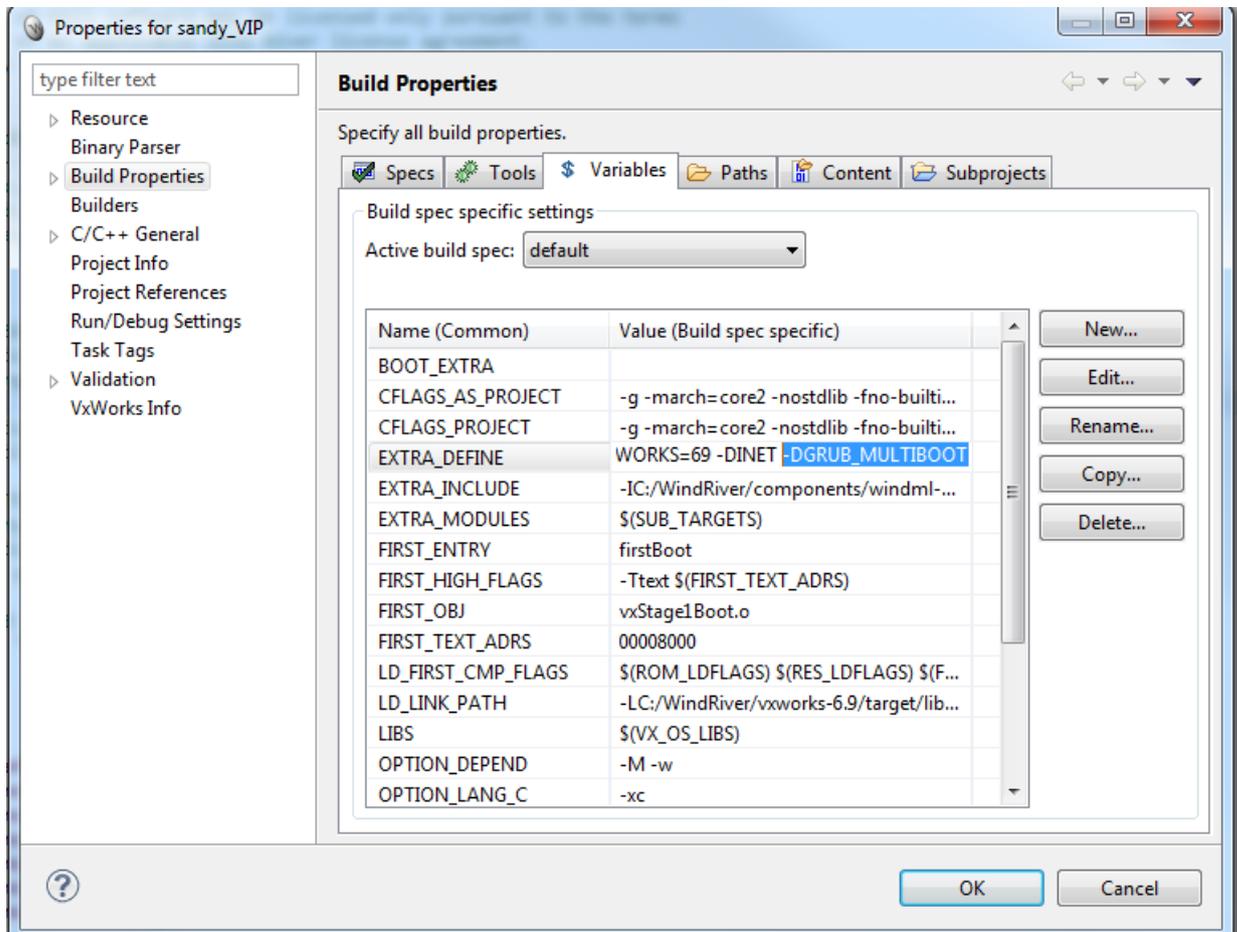
- VxWorks BootApp
- GRUB Bootloader
- GRUB Legacy BootLoader

Verwenden Sie GRUB, wenn bereits ein Betriebssystem installiert ist und Sie zusätzlich auch VxWorks benutzen möchten.

Bei Problemen mit dem VxWorks-Bootloader wenden Sie sich bitte an Wind River.

Allgemeine Voraussetzung für GRUB

1. Wählen Sie beim Erzeugen des VxWorks-Image "Properties/Build Properties/Variables/EXTRA_DEFINE".
2. Setzen Sie das Define "-DGRUB_MULTIBOOT".

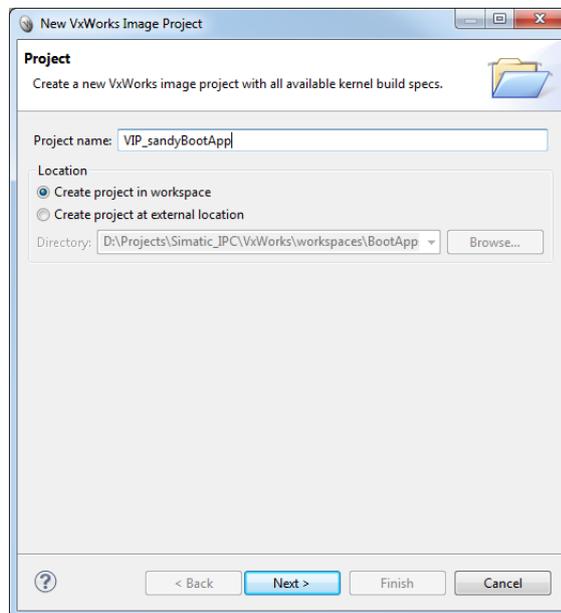


5.1 VxWorks BootApp

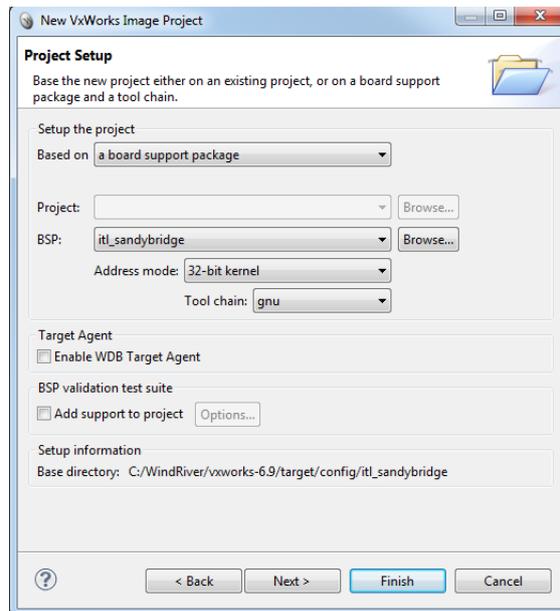
5.1.1 VxWorks BootApp erzeugen

Die nachfolgenden Schritte gelten für das BSP "itl_sandybridge" und können bei anderen BSPs unterschiedlich sein.

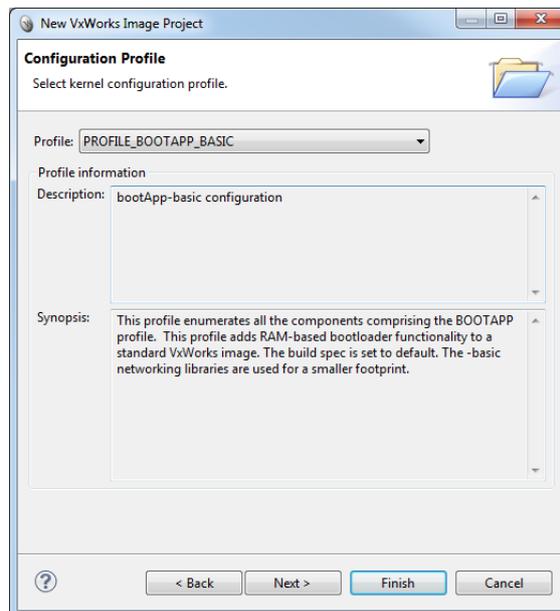
1. Öffnen Sie die Wind River Workbench.
Sie finden diese im Windows Startmenü unter "Start > All programs > Wind River > Workbench".
2. Wählen Sie einen Workspace, in dem Sie Ihr VxWorks Image Project erstellen möchten.
3. Wählen Sie "File > New > Project".
4. Öffnen Sie im folgenden Dialogfenster im Listenfeld den Ordner "VxWorks 6.x" und wählen Sie dann die Option "VxWorks Image Project".
5. Klicken Sie auf "Next".
6. Tragen Sie bei "Project name" den Projektnamen ein, z. B. "VIP_sandyBootApp".



- Wählen Sie unter "BSP" den Wert "itl_sandybridge". Der WDB Target Agent ist nicht aktiviert.

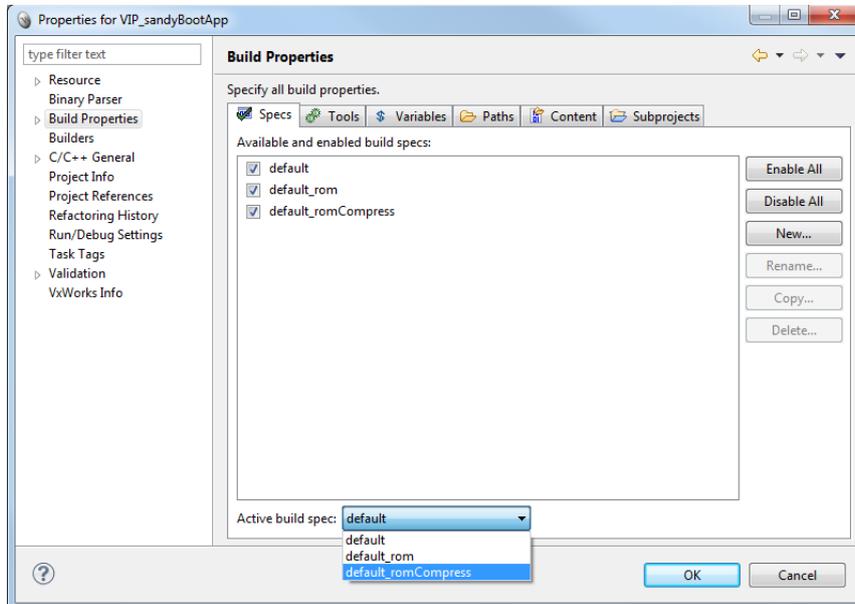


- SMP Unterstützung wird im Bootloader nicht benötigt. Überspringen Sie daher den nachfolgenden Dialog "Options". Wählen Sie unter "Profile" den Wert "PROFILE_BOOTAPP_BASIC" und klicken Sie auf "Finish".



- Klicken Sie mit der rechten Maustaste in der Baumstruktur auf das erstellte Projekt und öffnen Sie die Projekteinstellungen "Properties".

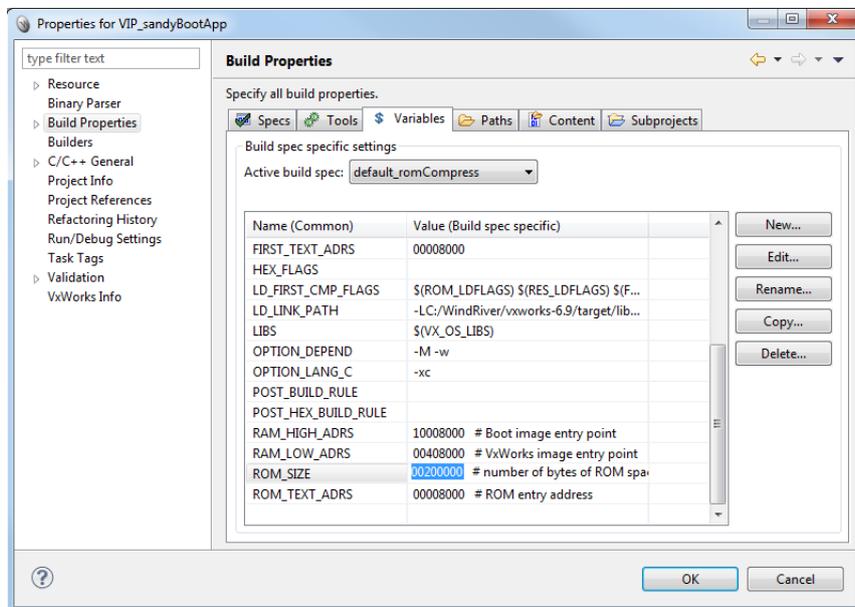
- 10. Wählen Sie im Register "Spec" im Listenfeld "Active build spec" die Option "default_romCompress". Diese Option ist nötig, da ein komprimiertes Image erzeugt werden muss.



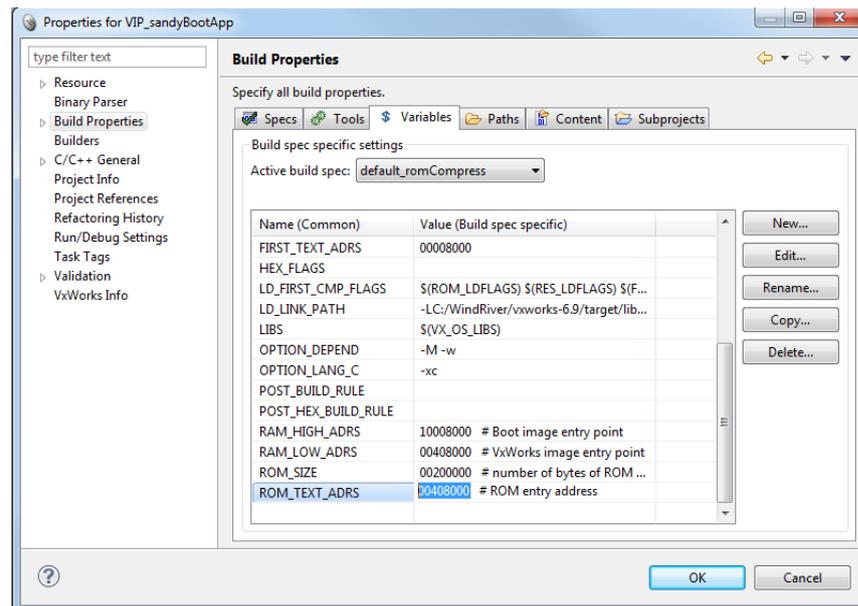
- 11. Die verwendeten Speicheradressen sind der Beschreibung "target.ref" des BSP "itl_sandyridge" entnommen.

Nehmen Sie im Register "Variables" folgende Einstellungen vor:

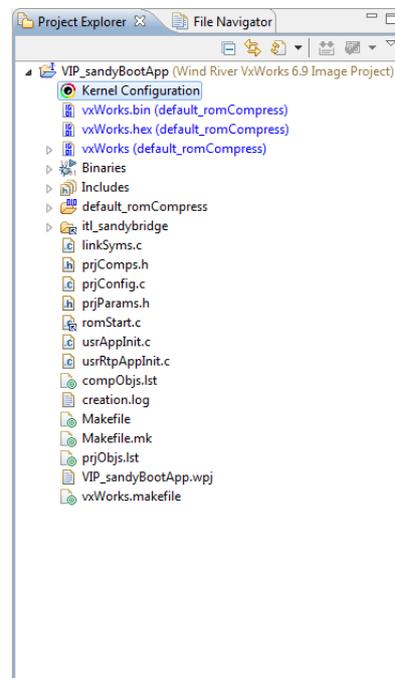
Stellen Sie den Wert von "ROM_SIZE" auf "00200000".



Stellen Sie den Wert von "ROM_TEXT_ADRS" auf "00408000".

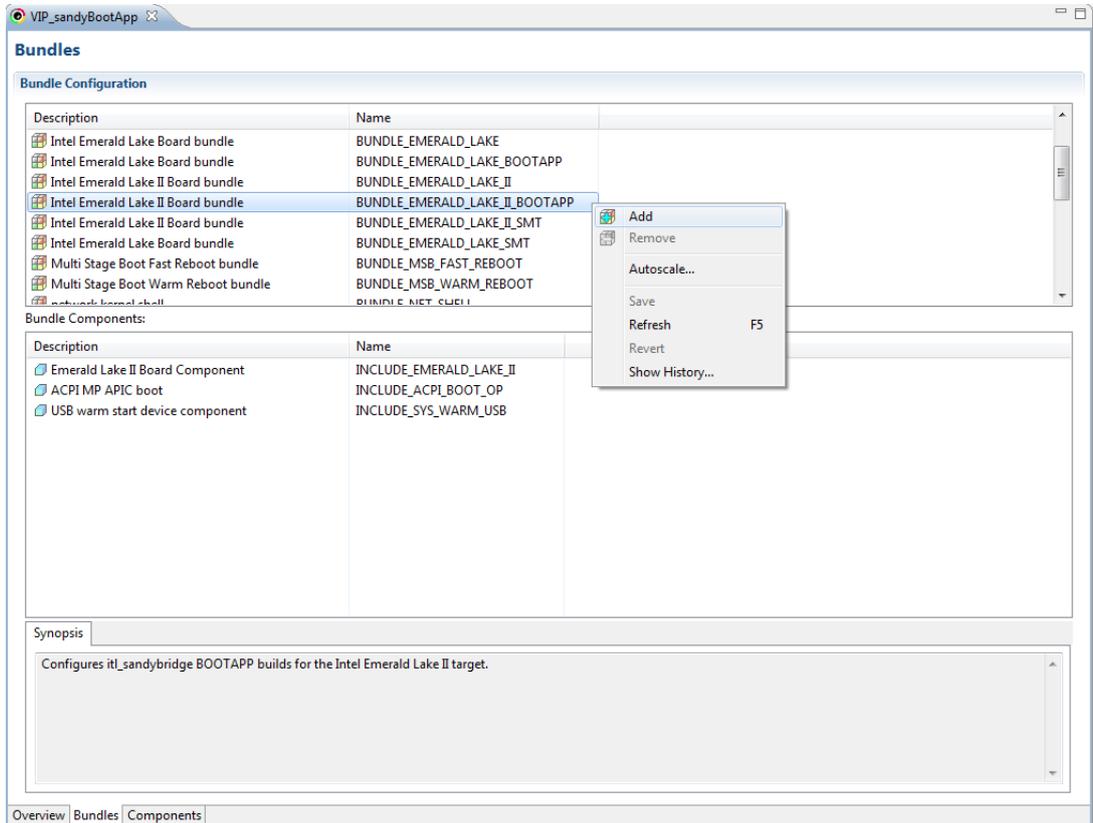


12. Doppelklicken Sie im "Project Explorer" in der Baumstruktur in Ihrem Projekt auf "Kernel Configuration".



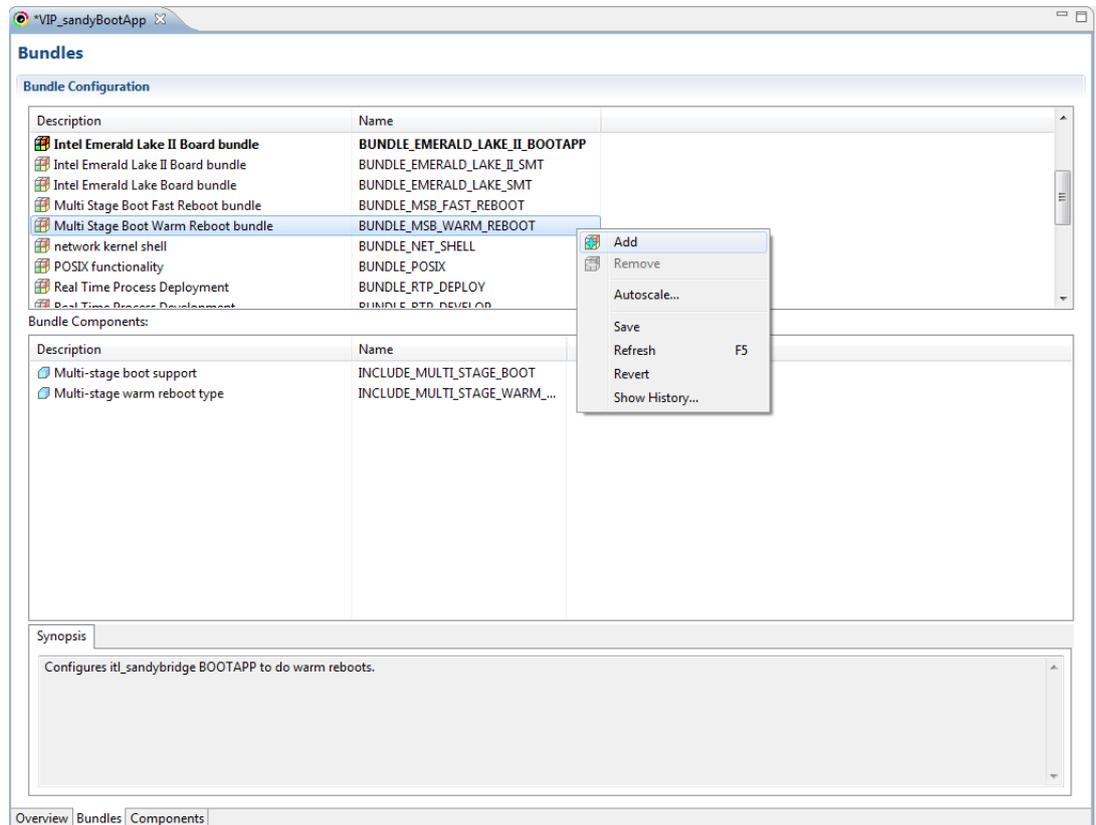
13. Wählen Sie das Register "Bundles" und klicken Sie mit der rechten Maustaste auf das Bundle "Intel Emerald Lake II Board bundle" mit dem Namen "BUNDLE_EMERALD_LAKE_II_BOOTAPP".

Wählen Sie aus dem Kontextmenü "Add".



14. Wählen Sie das Register "Bundles" und klicken Sie mit der rechten Maustaste auf das Bundle "Multi Stage Boot Warm Reboot bundle" mit dem Namen "BUNDLE_MSB_WARM_REBOOT".

Wählen Sie aus dem Kontextmenü "Add".



15. Wählen Sie das Register "Components".

Als Standardeinstellung werden die VxWorks-Bootparameter auf dem USB Device /bd0 erwartet.

Die zugehörige Komponente ist "USB warm start device component" mit dem Namen "INCLUDE_SYS_WARM_USB".

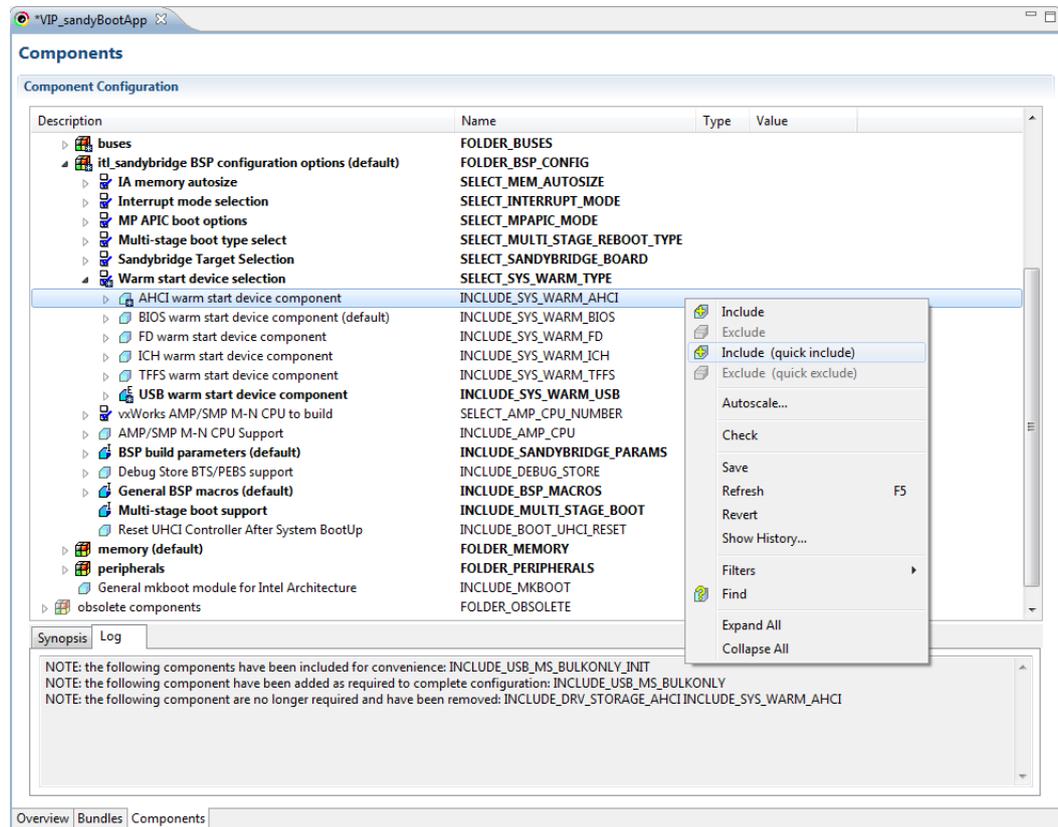
Wenn Sie von der Festplatte booten, klicken Sie mit der rechten Maustaste auf die Komponente "AHCI warm start device component" mit dem Namen "INCLUDE_SYS_WARM_AHCI".

Wählen Sie aus dem Kontextmenü "Include (quick include)".

Der benötigte Treiber für die Festplatte, "INCLUDE_DRV_STORAGE_AHCI", wird automatisch mit eingebunden.

Stellen Sie sicher, dass der "Boot Application FILESYSTEM Support" mit dem Namen "INCLUDE_BOOT_FILESYSTEMS" eingebunden ist.

Damit wird die Bootline vom Gerät /ata0:1 gelesen.

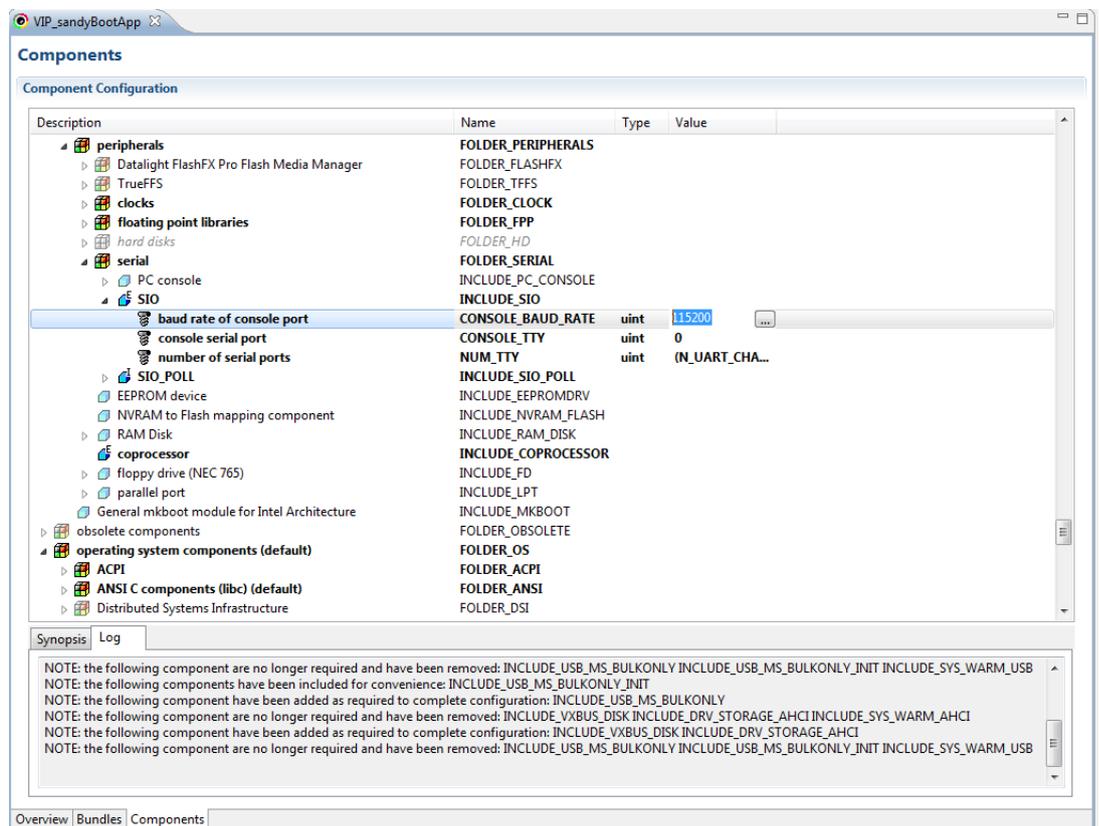


16. Navigieren Sie in der Baumstruktur zu "peripherals" und dann zur seriellen Schnittstelle "SIO".

Passen Sie die Baudrate der Bootkonsole "baud rate of console port" an die Einstellung des zu ladenden VxWorks-Image an. In diesem Beispiel ist die Baudrate 115200.

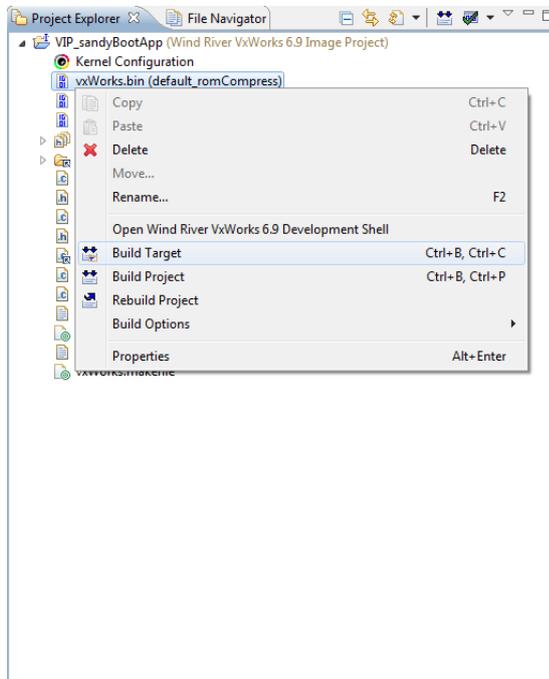
Hinweis

Sollte die Komponente INCLUDE_PC_CONSOLE eingebunden sein, entfernen Sie diese mit einem Rechtsklick auf die Komponente und anschließendem Klick im Kontextmenü auf "Exclude (quick exclude)".



17. Klicken Sie im "Project Explorer" in der Baumstruktur in Ihrem Projekt mit der rechten Maustaste auf das Target "vxWorks.bin".

Wählen Sie aus dem Kontextmenü "Build Target".

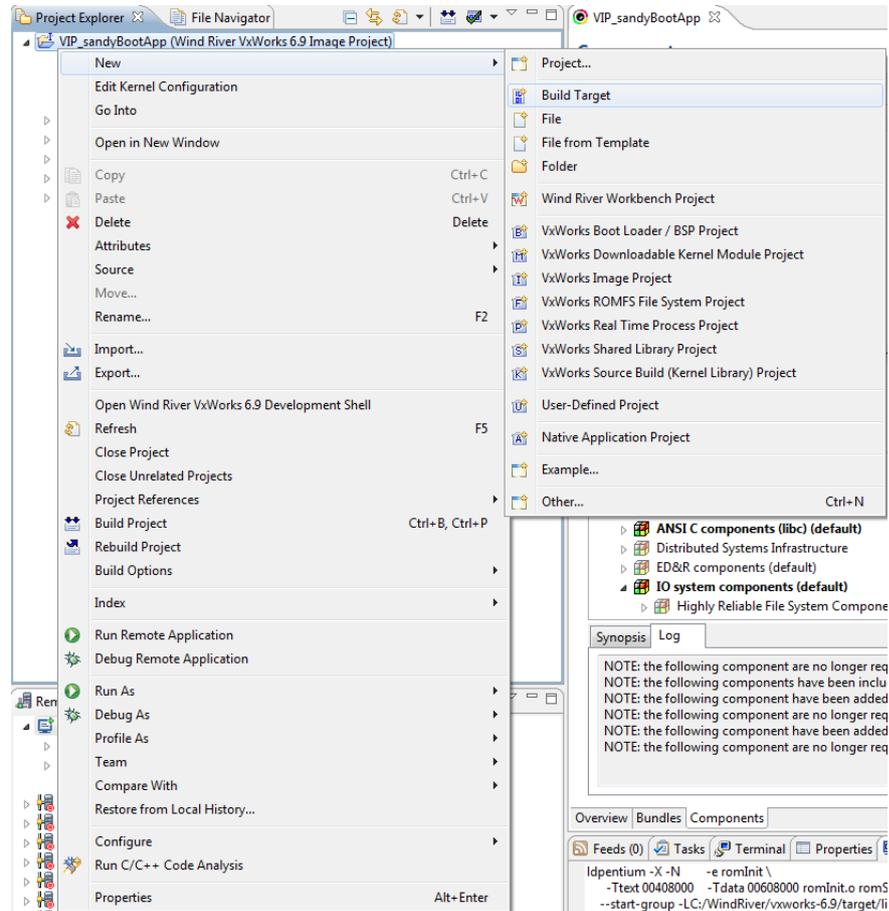


Der Second Stage Loader "vxWorks_romCompress.bin" wird erzeugt.

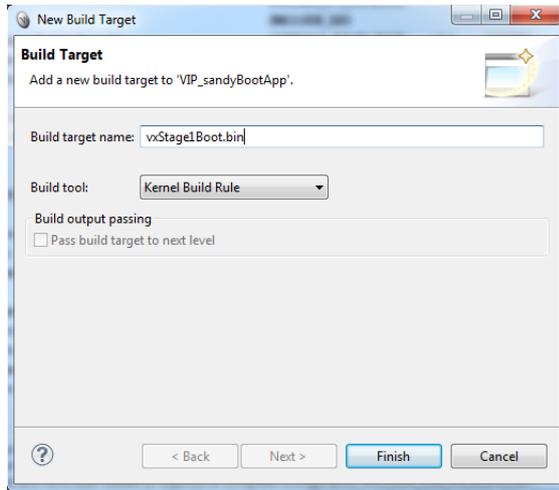
18. Es wird der First Stage Loader "vxStage1Boot.bin" benötigt.

Klicken Sie mit der rechten Maustaste im "Project Explorer" in der Baumstruktur auf ihr Projekt.

Wählen Sie "New" und dann "Build Target".

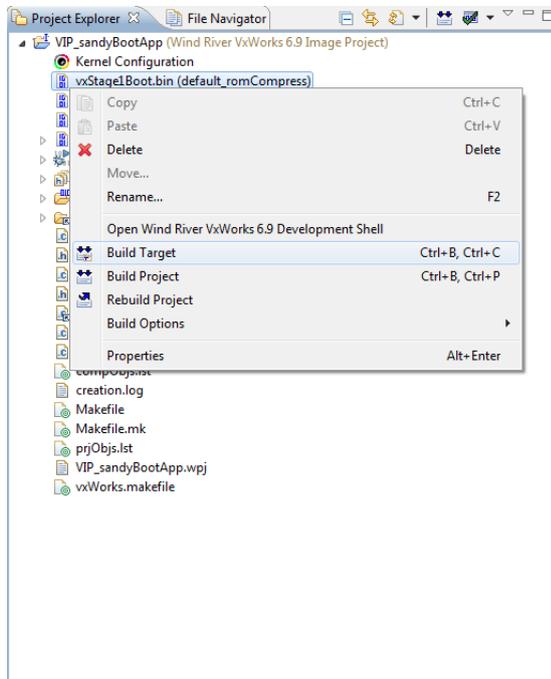


19. Tragen Sie im Feld "Build target name" "vxStage1Boot.bin" ein und klicken Sie auf "Finish".



Das neue Target wird erstellt.

20. Klicken Sie mit der rechten Maustaste auf "vxStage1Boot.bin" und wählen Sie "Build Target".



Der First Stage Loader wird gebaut.

21. Installieren Sie nun VxWorks auf der Festplatte Ihres SIMATIC IPC.

Problemlösung

Wenden Sie sich bei Problemen mit dem VxWorks-Bootloader an den Hersteller.

5.1.2 VxWorks auf der Festplatte des SIMATIC IPC installieren

Voraussetzung

Ein bootfähiger Windows-PE-USB-Speicherstick mit folgenden zusätzlichen Dateien, die in den Ordner \BSP zu kopieren sind:

mkbk.exe	Tool zur Erstellung des Bootsektors Sie können mkbk im Internet kostenlos herunterladen.
Bootsect.bin	Image des Bootsektors aus dem BSP "itl_sandybridge"
vxStage1Boot.bin	First Stage Bootloader Image
vxWorks_romCompress.bin	Second Stage Bootapp Image
vxWorks	VxWorks-Image zum Laden
nvrnm.txt	Bootparameter für die Second Stage Bootapp

Bootpartition mit einem Bootloader erstellen - Festplatte für "mkbk" konfigurieren

1. Erstellen Sie auf dem Windows-PE-USB-Speicherstick einen Ordner "BSP" und kopieren Sie alle benötigten Dateien in diesen Ordner (siehe Tabelle oben).
2. Stecken Sie den USB-Speicherstick in Ihren SIMATIC IPC und booten Sie Windows PE von diesem USB-Speicherstick.
3. Öffnen Sie die DOS-Eingabeaufforderung.
4. Geben Sie folgenden Befehl ein:

```
X:\Tools> diskpart
```

Das Kommandozeilen-Tool "DiskPart" wird gestartet.

5. Geben Sie folgenden Befehl ein:

```
DISKPART> list disk
```

Eine Liste der Datenträger wird angezeigt:

```
Disk ###  Status          Size      Free      Dyn  Gpt
-----  -
Disk 0    Online          465 GB   1024 KB
Disk 1    Online          1907 MB      0 B
```

6. Wählen Sie aus der Liste "Disk 0" aus. Die Festplatte ist Disk 0.

```
DISKPART> select disk 0
```

7. Prüfen Sie in der Liste der Datenträger, ob die richtige Festplatte ausgewählt ist.

Sie erkennen den ausgewählten Datenträger an dem Zeichen "*".

```
DISKPART> list disk
```

Disk ###	Status	Size	Free	Dyn	Gpt
* Disk 0	Online	465 GB	1024 KB		
Disk 1	Online	1907 MB	0 B		

8. Führen Sie den Befehl zum Löschen der Festplatte aus.

```
DISKPART> clean
```

9. Erstellen Sie eine aktive primäre Partition:

```
DISKPART> create partition primary size=2000
```

10. Wählen Sie die erste Partition:

```
DISKPART> select partition 1
```

11. Markieren Sie die ausgewählte Partition als aktiv:

```
DISKPART> active
```

12. Formatieren Sie die Partition als "FAT". Dies kann, abhängig von der Größe, bis zu 2 Minuten dauern.

```
DISKPART> format fs=fat
```

```
100 percent completed
```

13. Weisen Sie der Partition einen Laufwerksbuchstaben zu:

```
DISKPART> assign letter=C
```

Notieren Sie sich den zugewiesenen Laufwerksbuchstaben. Sie benötigen diese Information im Programm "mkb".

14. Beenden Sie DiskPart:

```
DISKPART> exit
```

VxWorks-Image auf die Festplatte kopieren

1. Wechseln Sie in den Ordner "BSP" und führen Sie den folgenden Befehl aus. Im hier beschriebenen Beispiel ist der Laufwerksbuchstabe der Startpartition "c":

```
mkbt -x bootsect.bin c:
```

Folgende Informationen werden angezeigt:

```
* Expert mode (-x)
```

```
* Warning different filesystem ID
```

```
Size=0bytes OEM="MSDOS5.0" VolLabel="NO NAME" FileSys="FAT16"
```

2. Kopieren Sie die Datei "vxStage1Boot.bin" als "bootrom.sys" auf die Festplatte.

Das Abbild auf der Festplatte muss zusammenhängend sein, sonst schlägt die erste Stufe des Bootvorgangs fehl.

Ein zusammenhängendes Abbild kann erzeugt werden, indem alle Abbilder von der Startpartition an einen temporären Ort kopiert, alle Abbilder von der Startpartition gelöscht und anschließend die einzelnen Abbilder zurück auf die Startpartition kopiert werden.

```
copy vxStage1Boot.bin c:bootrom.sys
```

3. Kopieren Sie "vxWorks_romCompress.bin" als "bootapp.sys" auf die Festplatte:

```
copy vxWorks_romCompress.bin c:bootapp.sys
```

4. Kopieren Sie das VxWorks-Image und die Datei "nvram.txt" auf die Festplatte:

```
copy vxWorks c:
```

```
copy nvram.txt c:
```

5. Entfernen Sie den USB-Speicherstick von Ihrem Zielsystem.

Beispiele für den Inhalt der Datei "nvram.txt"

Der VxWorks-Loader liest die Bootparameter aus der Datei "nvram.txt".

- Booten von USB:

```
fs(0,0)host:/bd0/vxWorks e=192.168.1.62 o=gei0 tn=IPC427D
```

Für den Parameter "tn" (target name) wurde beispielhaft "IPC427D" angegeben.

- Booten von HDD:

```
fs(0,0)host:/ata0:1/vxWorks e=192.168.1.62 o=gei0 tn=IPC427D
```

Für den Parameter "tn" (target name) wurde beispielhaft "IPC427D" angegeben.

Weiterführende Informationen

Eine genaue Beschreibung der Bootparameter finden Sie im Handbuch "VxWorks Kernel Programmer's Guide", das im Lieferumfang von VxWorks enthalten ist.

5.2 GRUB Bootloader

Voraussetzung

Ein Ubuntu-Live-USB-Speicherstick mit folgenden zusätzlichen Dateien:

- grub.cfg
Konfigurationsdatei von GRUB
- vxWorks
VxWorks-Image zum Laden

Beispiele für den Inhalt der Datei "grub.cfg"

GRUB liest die Einstellungen und Boot-Menüeinträge aus der Datei „grub.cfg“.

Es soll im Beispiel vxWorks mit target name "ipc277e" und IP-Adresse 192.168.1.66 von der ersten Partition der ersten Festplatte geladen werden:

```
# grub.cfg configuration file for Grand Unified Boot Loader (GRUB)

set timeout=10 # time before default configuration is started
set default=0 # number of default configuration, starts with 0

menuentry "VxWorks 6.9"{
    multiboot (hd0,msdos1)/vxWorks sysbootline:fs(0,0)host:/ata0:1/vxWorks
e=192.168.1.66 o=gei0 tn=ipc277
}
```

Ubuntu-Live-USB-Stick erstellen

Verwenden Sie zum Erstellen eines Ubuntu-Live-USB-Sticks das Tool "UNetbootin".

Sie können "UNetbootin" unter <http://unetbootin.sourceforge.net/> herunterladen.

1. Stecken Sie einen leeren FAT32-formatierten USB-Stick, welcher eine Mindestgröße von 2 GByte hat, an ihren Windows Rechner und starten Sie Unetbootin.exe.
2. Wählen Sie als Distribution den Listeneintrag "Ubuntu" und als Version eine "Live" Version z.B. "14.04_Live".
3. Alternativ können Sie ein Live Image (.iso) von <http://www.ubuntu.com> herunterladen und unter dem Punkt "Abbild" auswählen.
4. Wählen Sie als Typ "USB-Laufwerk" und bei Laufwerk den Buchstaben des USB-Sticks.
5. Erstellen Sie den Ubuntu-Live-USB-Stick mit Klick auf "OK".
6. Kopieren Sie die beiden Dateien "vxWorks" und "grub.cfg" in das Root-Verzeichnis des Ubuntu-Live-USB-Stick.

Bootpartition erstellen

1. Stecken Sie den Ubuntu-Live-USB-Stick in Ihren SIMATIC IPC und booten Sie Ubuntu im Live-Modus von diesem Ubuntu-Live-USB-Stick ("Try Ubuntu without installing").

Hinweis

Bei Installation auf einem Panel PC muss evtl. ein externer Monitor angeschlossen werden, damit der Ubuntu Desktop angezeigt wird.

1. Öffnen Sie das Tool "Gparted".
2. Wählen Sie die Festplatte aus, auf der Sie GRUB installieren möchten. Die erste Festplatte ist gewöhnlich "/dev/sda".
3. Erstellen Sie über "Device > Create Partition Table" eine "msdos" Partitionstabelle.
4. Erstellen Sie über "Partition > New" eine neue primäre Partition mit maximaler Größe 2 GB und Dateisystem FAT16.
5. Bestätigen Sie die Erstellung der Partition mit "Edit > Apply All Operations".
6. Setzen Sie für die erstellte Partition das "boot"-Flag über "Partition > Manage Flags".

GRUB installieren

1. Öffnen Sie ein Terminal-Fenster.
2. Wechseln Sie zum Superuser.
3. Binden Sie die erste Partition der Festplatte (hier /dev/sda) ein, auf der Sie GRUB installieren möchten.

```
~$ sudo su
```

```
# mkdir /mnt/HDD && mount /dev/sda1 /mnt/HDD
```

4. Installieren Sie GRUB auf der Festplatte (hier /dev/sda).

```
# grub-install --force --no-floppy --boot-directory=/mnt/HDD/boot /dev/sda
```

5. Wechseln Sie auf den Ubuntu-Live-USB-Stick (hier als /cdrom eingebunden).

```
# cd /cdrom
```

6. Kopieren Sie die Datei "grub.cfg" auf die Festplatte.

```
# cp grub.cfg /mnt/HDD/boot/grub
```

7. Kopieren Sie die Datei "vxWorks" auf die Festplatte.

```
# cp vxWorks /mnt/HDD
```

8. Entfernen Sie den Ubuntu-Live-USB-Stick und starten Sie den PC neu.

Nach dem Neustart erscheint ein GRUB2-Bootmenü.

5.3 GRUB Legacy Bootloader

Voraussetzung

Ein bootfähiger Windows-PE-USB-Speicherstick mit folgenden zusätzlichen Dateien:

- grubinst
Tool zur Installation von GRUB.

Sie können das Paket grubinst unter folgendem Link kostenlos herunterladen:

<http://sourceforge.net/projects/grub4dos/files/grubinst/>

Hinweis

Klicken Sie auf der Internet-Seite auf den Ordner "grubinst 1.0.1" und laden Sie anschließend die Datei "grubinst_1.0.1_bin_win.zip" herunter.

- menu.lst
Konfigurationsdatei von GRUB
- vxWorks
VxWorks-Image zum Laden

Beispiele für den Inhalt der Datei "menu.lst"

GRUB Legacy liest die Einstellungen und Boot-Menüeinträge aus der Datei "menu.lst".

Es soll im Beispiel vxWorks mit target name „ipc277e“ und IP-Adresse 192.168.1.66 von der ersten Partition der ersten Festplatte geladen werden:

```
# menu.lst configuration file for Grand Unified Boot Loader (GRUB)

timeout 10 # time before default configuration is started
default 0 # number of default configuration, starts with 0

title VxWorks 6.9
    root (hd0,0)
    kernel /vxWorks sysbootline:fs(0,0)host:/ata0:1/vxWorks e=192.168.1.66
    o=gei0 tn=ipc277e
```

Bootpartition erstellen

1. Entpacken Sie das Verzeichnis "grubinst" aus dem heruntergeladenen grubinst-Archiv.
2. Kopieren Sie das entpackte Verzeichnis "grubinst" in das Root-Verzeichnis des Windows-PE-USB-Speicherstick.
3. Kopieren Sie die Dateien "menu.lst" und "vxWorks" in das Root-Verzeichnis des Windows-PE-USB-Speicherstick.

4. Stecken Sie den USB-Speicherstick in Ihren SIMATIC IPC und booten Sie Windows PE von diesem USB-Speicherstick.

5. Öffnen Sie die DOS-Eingabeaufforderung und geben Sie folgenden Befehl ein:

```
X:\Tools> diskpart
```

6. Geben Sie im Kommandozeilen-Tool "DiskPart" folgenden Befehl ein:

```
DISKPART> list disk
```

Eine Liste der Datenträger wird angezeigt:

```
Disk ### Status Size Free Dyn Gpt
```

```
Disk 0 Online 465 GB 1024 KB
```

```
Disk 1 Online 1907 MB 0 B
```

7. Wählen Sie aus der Liste "Disk 0" aus. Die Festplatte ist Disk 0.

```
DISKPART> select disk 0
```

8. Prüfen Sie in der Liste der Datenträger, ob die richtige Festplatte ausgewählt ist.

Sie erkennen den ausgewählten Datenträger an dem Zeichen "***".

```
DISKPART> list disk
```

```
Disk ### Status Size Free Dyn Gpt
```

```
Disk 0 Online 465 GB 1024 KB
```

```
Disk 1 Online 1907 MB 0 B
```

9. Führen Sie den Befehl zum Löschen der Festplatte aus.

```
DISKPART> clean
```

10. Erstellen Sie eine aktive primäre Partition:

```
DISKPART> create partition primary size=2000
```

11. Wählen Sie die erste Partition:

```
DISKPART> select partition 1
```

12. Markieren Sie die ausgewählte Partition als aktiv:

```
DISKPART> active
```

13. Formatieren Sie die Partition als "FAT". Dies kann, abhängig von der Größe, bis zu 2 Minuten dauern.

```
DISKPART> format fs=fat
```

```
100 percent completed
```

14. Weisen Sie der Partition einen Laufwerksbuchstaben zu:

```
DISKPART> assign letter=C
```

15. Beenden Sie DiskPart:

```
DISKPART> exit
```

GRUB installieren

1. Wechseln Sie zum USB-Speicherstick und dort in den Ordner „grubinst“.
2. Führen Sie folgenden Befehl aus. Im hier beschriebenen Beispiel befindet sich die Startpartition auf der Festplatte (hd0):

```
grubinst (hd0)
```

3. Kopieren Sie die Datei "grldr" in die Wurzel der eben erstellten Bootpartition ("c:").

```
copy grldr c:
```

4. Wechseln Sie zurück in die Wurzel des USB-Speichersticks und kopieren Sie die GRUB-Konfigurationsdatei auf die Festplatte.

```
copy menu.lst c:
```

5. Kopieren Sie das VxWorks-Image auf die Festplatte.

```
copy vxWorks c:
```

6. Entfernen Sie den USB-Speicherstick von Ihrem Zielsystem.

5.4 GRUB Bootloader und Windows

Dieses Kapitel beschreibt, wie Sie einer bestehenden Windows 7-Installation mit Hilfe von GRUB ein bootfähiges VxWorks hinzufügen. Zur Installation von GRUB 2 wird im Folgenden Grub2Win verwendet. Dieses Tool verändert nicht den MBR (Master Boot Record), sondern fügt GRUB als Option in den Windows 7-Bootloader ein.

Der GRUB-Bootloader lädt das VxWorks-Image, das auf dem Massenspeicher (AHCI, USB) abgelegt ist.

Voraussetzungen

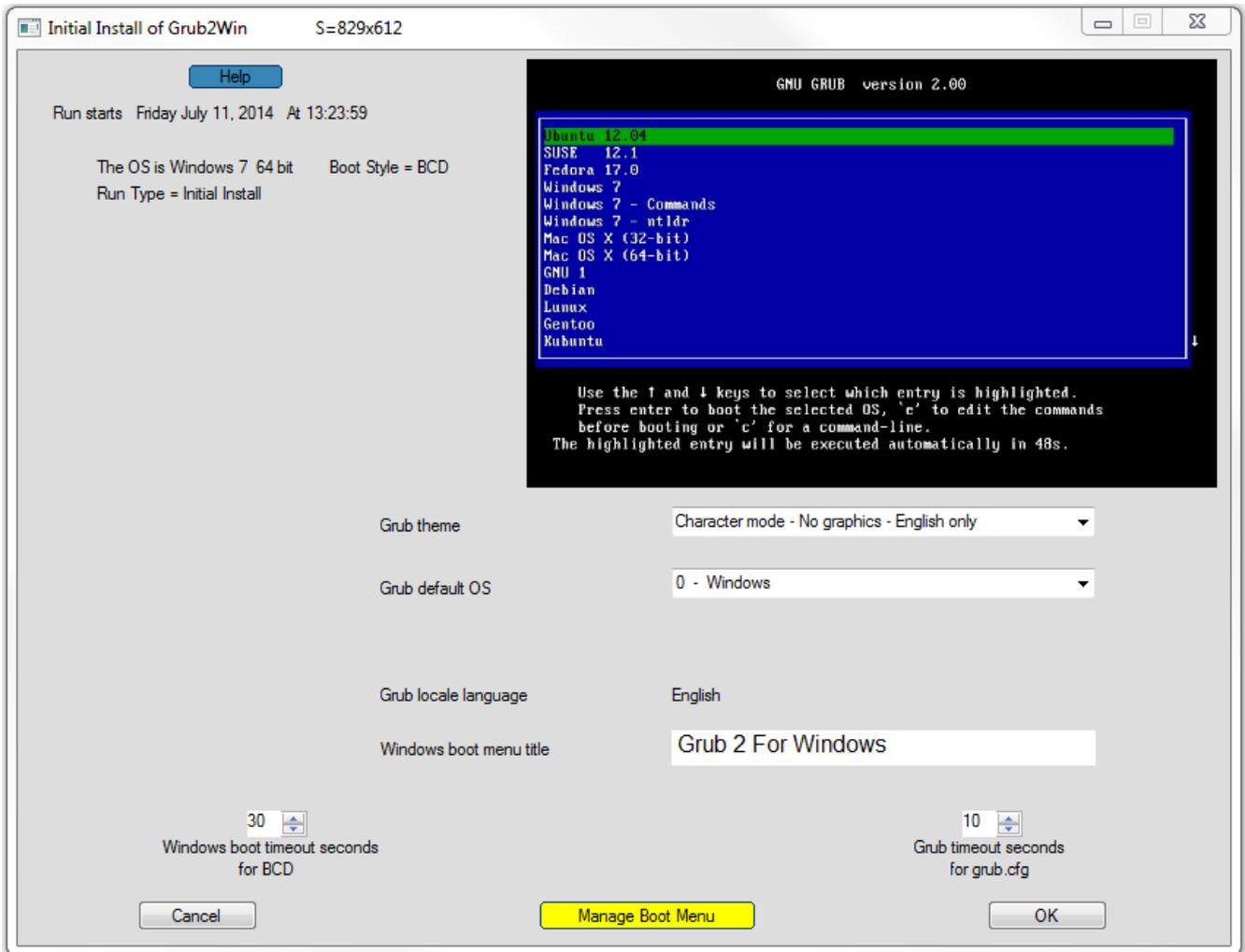
- Sie haben das Tool Grub2Win (<http://sourceforge.net/projects/grub2win/>) aus dem Internet heruntergeladen und in einem temporären Verzeichnis gespeichert.
- Das VxWorks-Image ist auf einer Partition der Festplatte abgelegt.

GRUB installieren

Führen Sie, falls gefordert, die Installation als Administrator aus.

1. Starten Sie den Windows Explorer.
2. Entpacken Sie Grub2Win und kopieren Sie das Verzeichnis "grub2" in das Windows-Systemlaufwerk (normalerweise C:).
3. Wechseln Sie in das gerade kopierte Verzeichnis "grub2".

4. Starten Sie die Datei "grub2win.exe".
Es öffnet sich folgende Oberfläche:



Wird rechts oben anstelle des Bootmenü-Screenshots angezeigt, dass kein Screenshot verfügbar ist, so haben Sie das Verzeichnis "grub2" nicht in das Windows-Systemlaufwerk kopiert.

5. Wählen Sie aus der Liste "Grub theme" ein Bootmenü-Design aus.
6. Passen Sie ggf. die Boot-Timeout-Zeiten des BCD- und Grub-Bootloaders an.
7. Klicken Sie auf OK.

Der GRUB-Bootloader wird installiert und die Konfigurationsdatei "grub2\grub.cfg" wird erstellt.

VxWorks zu GRUB hinzufügen

1. Öffnen Sie die Grub-Konfigurationsdatei "grub.cfg" mit einem Texteditor (z. B. WordPad).
2. Fügen Sie in die "grub2win-user-section" den folgenden Eintrag ein:

```
# start-grub2win-user-section
*****

#

#

menuentry "VxWorks" {

multiboot (hd0,msdos5)/vxWorks sysbootline:fs(0,0)host:/ata0:1/vxWorks
e=192.168.1.62 o=gei0 tn=ipc427d

}
```

Der Eintrag "(hd0,msdos5)/vxWorks" gibt den Ablageort des VxWorks-Image an. Dieser muss an Ihr System angepasst werden.

In diesem Beispiel liegt das VxWorks-Image in der Partition "msdos5" auf der Festplatte "hd0". Die Festplattennummer und Anordnung der Partitionen können Sie z. B. unter Computerverwaltung > Datenträgerverwaltung nachsehen.

Die VxWorks-sysbootline müssen Sie ebenfalls an Ihre Bedürfnisse anpassen.

3. Im Grub-Bootloader wird standardmäßig Windows vorausgewählt und nach der angegebenen Timeout-Zeit gestartet. Möchten Sie stattdessen VxWorks vorauswählen, starten sie das Programm "grub2win.exe" erneut und wählen Sie unter "Grub default OS" den Eintrag "VxWorks" aus.

Grub ist nun mit einem Eintrag für VxWorks installiert.

BCD Default Eintrag ändern (optional)

Beim Systemstart wird zuerst der Windows BCD-Bootloader geladen. Dieser zeigt nun die folgenden Einträge an:

- Windows 7:
Mit dem vorausgewählten Eintrag "Windows 7" wird direkt Windows gestartet.
- GRUB For Windows:
Mit diesem Eintrag wird der in den vorherigen Schritten installierte Grub-Bootloader gestartet.

Möchten Sie, dass Grub im BCD vorausgewählt ist und automatisch gestartet wird, führen Sie die folgenden Schritte aus:

1. Öffnen Sie unter Windows eine Eingabeaufforderung als Administrator.
2. Geben Sie folgenden Befehl ein:

```
C:\windows\system32>bcdedit /enum
```

Die Einträge des BCD werden angezeigt:

Windows-Start-Manager

```
Bezeichner {bootmgr}
device partition=X:
description Windows Boot Manager
locale de-DE
inherit {globalsettings}
default {current}
resumeobject {52024e9a-2855-11e2-b655-f29c655de332}
displayorder {current}
    {b9b88c23-066a-11e4-9cf0-001b1b414eae}
toolsdisplayorder {memdiag}
timeout 5
```

Windows-Startladeprogramm

```
Bezeichner {current}
device partition=C:
path \windows\system32\winload.exe
description Windows7
locale de-DE
inherit {bootloadersettings}
recoverysequence {52024e9c-2855-11e2-b655-f29c655de332}
recoveryenabled Yes
osdevice partition=C:
systemroot \windows
resumeobject {52024e9a-2855-11e2-b655-f29c655de332}
nx OptIn
```

Echtmodus-Startabschnitt

```
Bezeichner {b9b88c23-066a-11e4-9cf0-001b1b414eae}
device partition=C:
path \grub2\winloader\grub2.boot
description Grub 2 For Windows
```

3. Kopieren Sie den Bezeichner des Grub-Eintrags.
4. Setzen Sie "Grub 2" als Default.

```
C:\windows\system32>bcdedit /default {b9b88c23-066a-11e4-9cf0-001b1b414eae}
```

Der zweite Parameter des Befehls ist der im vorherigen Schritt kopierte Bezeichner des Grub-Eintrags.

Beim Systemstart wird nun Grub vorausgewählt.

Grub kann das VxWorks-Image nicht finden

Sollte Grub das VxWorks-Image nicht auf der Festplatte finden, kann es sein, dass die in der Datei "grub.cfg" angegebene Festplatten- oder Partitionsnummer nicht stimmt.

Überprüfen Sie den Pfad Ihres VxWorks-Image wie folgt:

1. Wechseln Sie in Grub mit ,c' zum Befehlszeilen-Modus.
2. Mit dem Befehl "ls" zeigen Sie alle verfügbaren Partitionen an.

```
grub> ls  
  
(hd0) (hd0,msdos5) (hd0,msdos2) (hd0,msdos1)
```

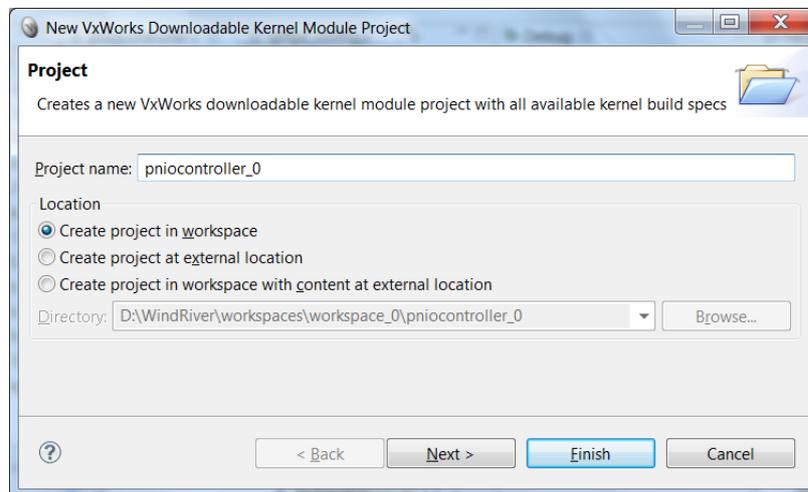
3. Prüfen Sie, auf welcher Partition das VxWorks-Image liegt. Sie erkennen die richtige Partition am Dateisystemtyp und am Namen.

```
grub> ls (hd0,msdos5)  
  
Partition hd0,msdos5: Filesystem type fat - Label 'VXWORKS' ...
```

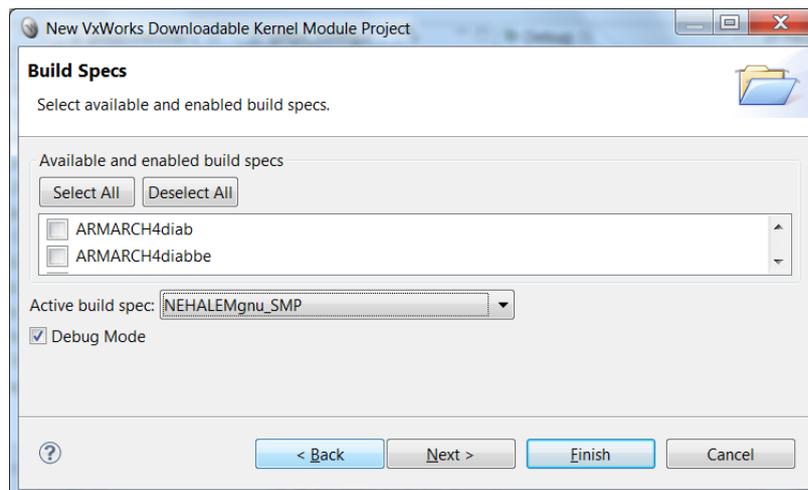
4. Wenn Sie die richtige Partitionsbezeichnung gefunden haben, ändern Sie den Eintrag in der Datei "grub.cfg" unter Windows entsprechend ab.

Nachladbares Kernel Modul erzeugen

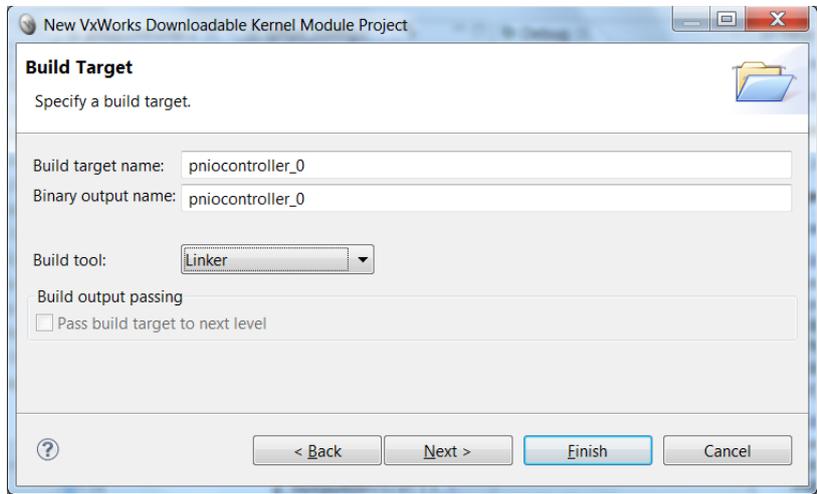
1. Öffnen Sie die Wind River Workbench.
2. Wählen Sie "File > New > Project".
3. Öffnen Sie im folgenden Dialogfenster im Listenfeld den Order "VxWorks 6.x" und wählen Sie dann die Option "VxWorks Downloadable Kernel Module Project".
4. Klicken Sie auf "Next".
5. Tragen Sie bei "Project name" den Namen für das Kernel Module Projekt ein und klicken Sie auf "Next".



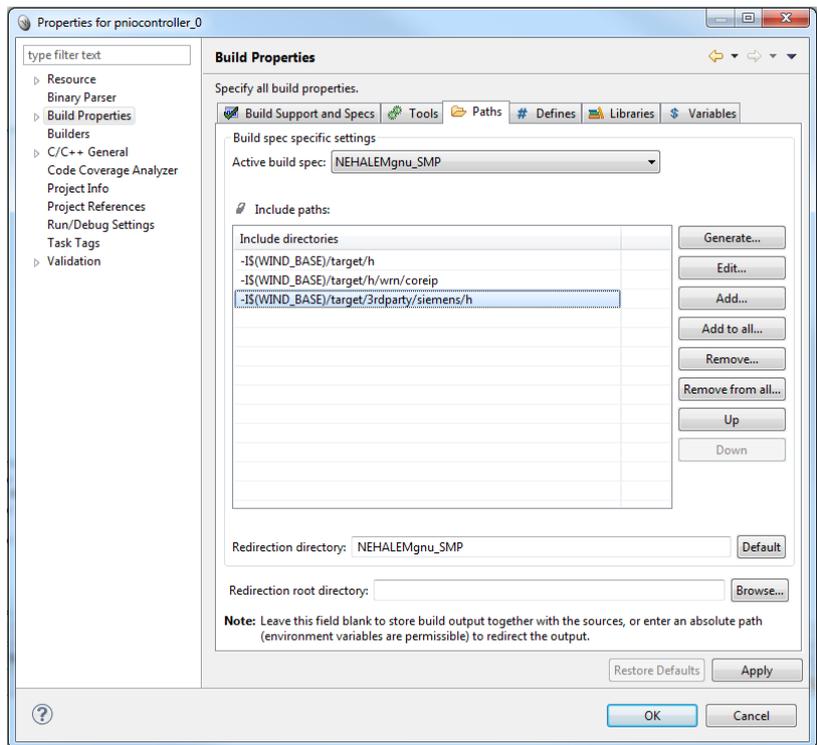
6. Überspringen Sie die nachfolgenden Dialoge "Project", "Built Defaults" und "Build Support".
7. Deaktivieren Sie im Dialog "Build Spec" alle Einträge außer "NEHALEMgnu_SMP".
8. Wählen Sie bei "Active build spec" die Option "NEHALEMgnu_SMP".



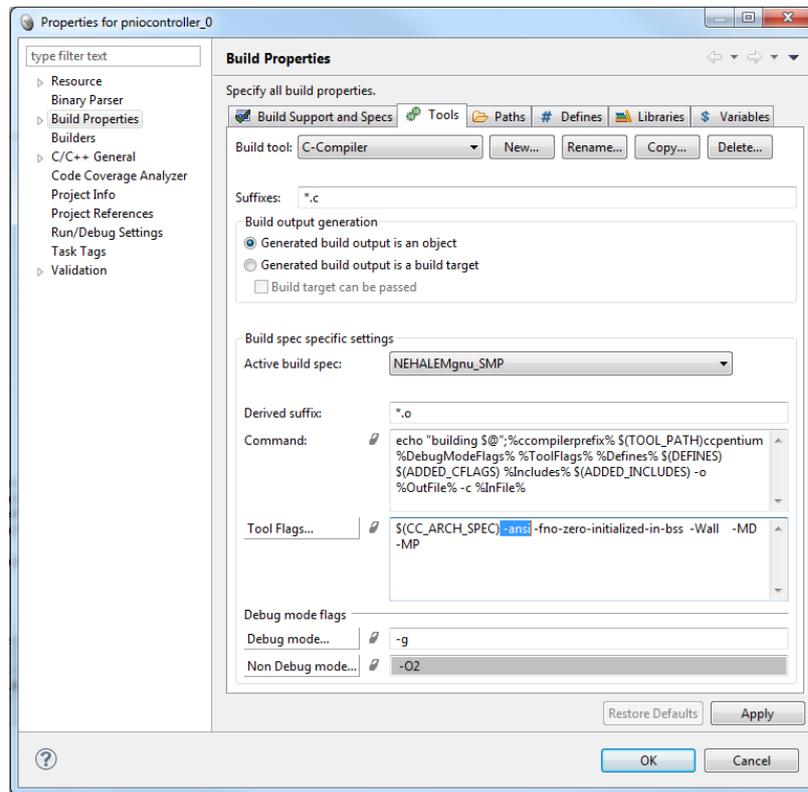
9. Klicken Sie auf "Next".
10. Übernehmen Sie im Dialog "Build Target" die Default-Einstellung. Achten Sie darauf, dass bei "Build tool" die Option "Linker" eingestellt ist.



11. Klicken Sie auf "Finish".
12. Öffnen Sie die Projekteinstellungen "Properties" und wechseln Sie zu den "Build Properties".
13. Fügen Sie im Register "Paths" den Suchpfad "-I\$(WIND_BASE)/target/3rdparty/siemens/h" mit "Add" hinzu.

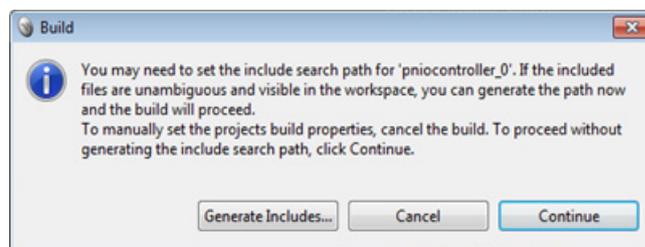


14. Wenn Sie in Ihrem C Source Code die Kommentarzeichen "//" verwenden, löschen Sie den Schalter "-ansi" aus dem Feld "Tool Flags" in dem Register "Tools".



15. Fügen Sie Ihren Source Code hinzu und kompilieren Sie das Kernel Modul mit Klick auf "Project > Build Project".

Wenn beim Kompilieren der Anwendung der untenstehende Dialog "Build" angezeigt wird und Sie alle Includepfade zu den Projekteinstellungen hinzugefügt haben, klicken Sie "Continue".



Ergebnis

Das nachladbare Kernel Modul ist erstellt.

Der Name der erzeugten nachladbaren Datei lautet in diesem Beispiel "pnicontroller_0.out". Diese Datei kann über die Workbench oder über das Filesystem auf dem Zielsystem geladen werden.

Real Time Process Applikation (RTP) erstellen

Eine Real Time Process Applikation erstellen Sie auf die gleiche Weise wie ein nachladbares Kernel Modul.

- Wählen Sie als Projekttyp die Option "VxWorks Real Time Process Project".
- Wählen Sie im Dialog "Build Specs" die Option "NEHALEMgnu_RTP".

Hinweis

Der Name der erzeugten Datei erhält die Erweiterung ".vxe".

Siehe auch

Nachladbares Kernel Modul erzeugen (Seite 55)

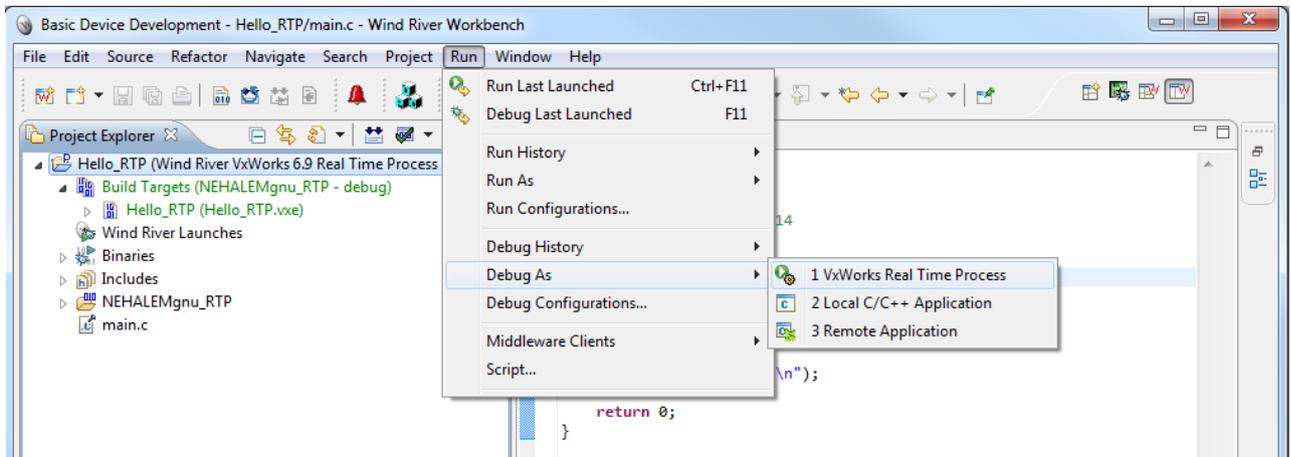
PROFINET-Aufrufe im Real Time Process (User Mode) nutzen (Seite 70)

PROFIBUS-Aufrufe im Real Time Process (User Mode) nutzen (Seite 78)

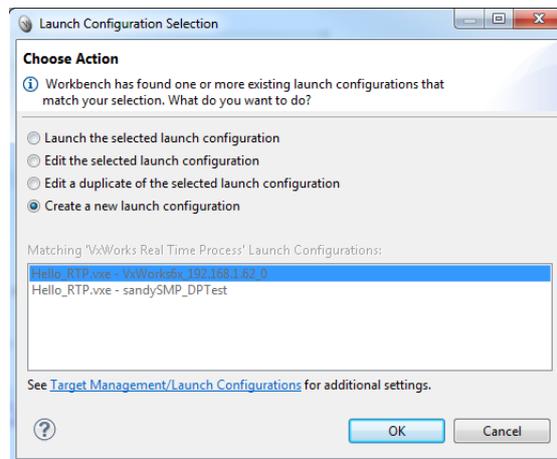
Hardwarenahe Funktionen im Real Time Process (User Mode) nutzen (Seite 85)

Real Time Process Applikation (RTP) debuggen

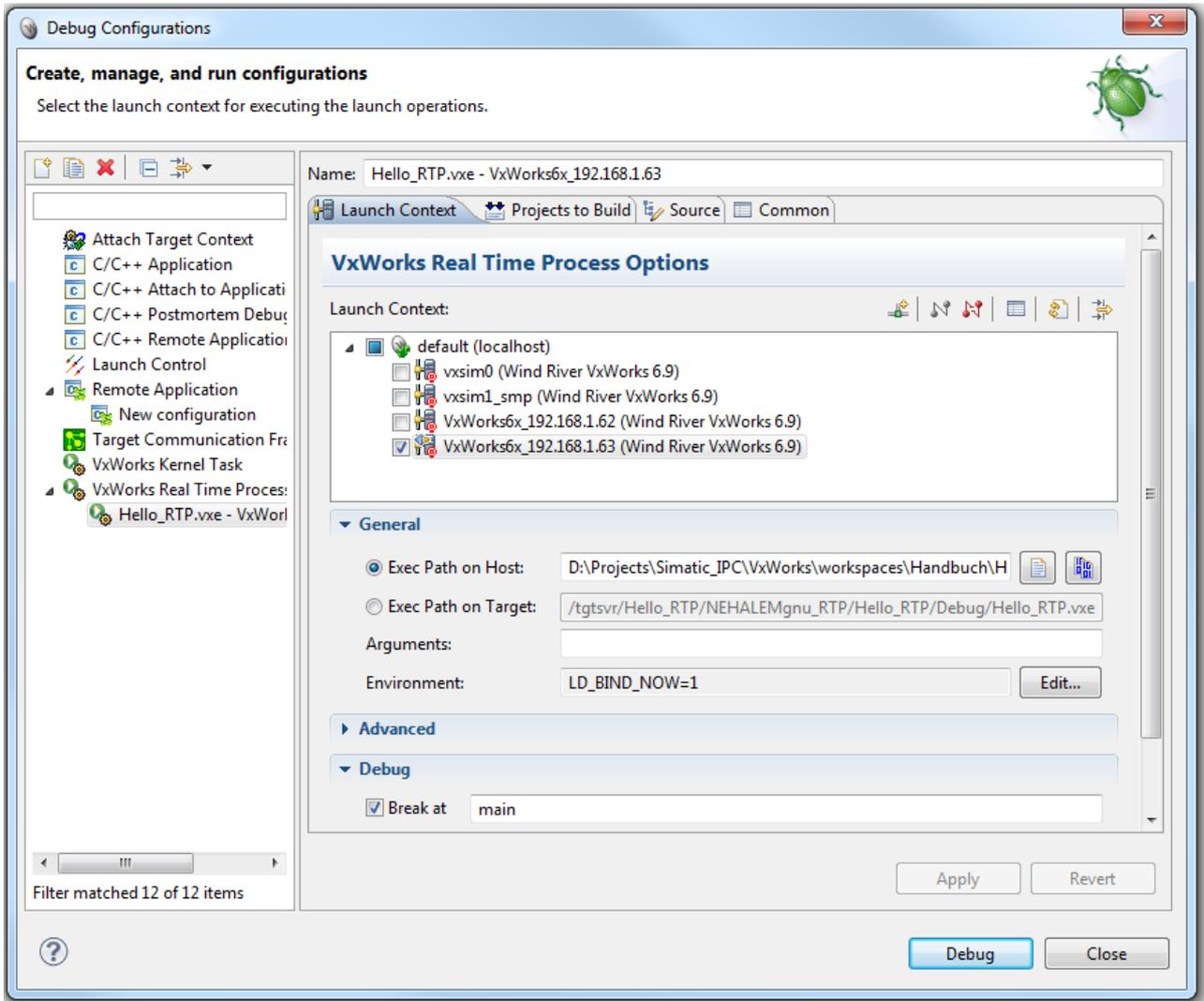
1. Öffnen Sie den "Project Explorer" und markieren Sie das RTP-Projekt.
2. Wählen Sie die Option "Run > Debug As > VxWorks Real Time Process".



3. Wählen Sie im Dialog "Launch Configuration Selection" die Option "Create a new launch configuration".



4. Wählen Sie im Register "Launch Context" die richtige Target-Server-Verbindung zu Ihrem Target.
Achten Sie darauf, dass im Bereich "General" unter der Option "Exec Path on Target" der Pfad mit "/tgtsvr/" beginnt.
5. Klicken Sie auf "Debug".



Ergebnis

Das RTP-Programm wird übertragen und das Debuggen wird gestartet.

Wenn die Debug-Verbindung fehl schlägt

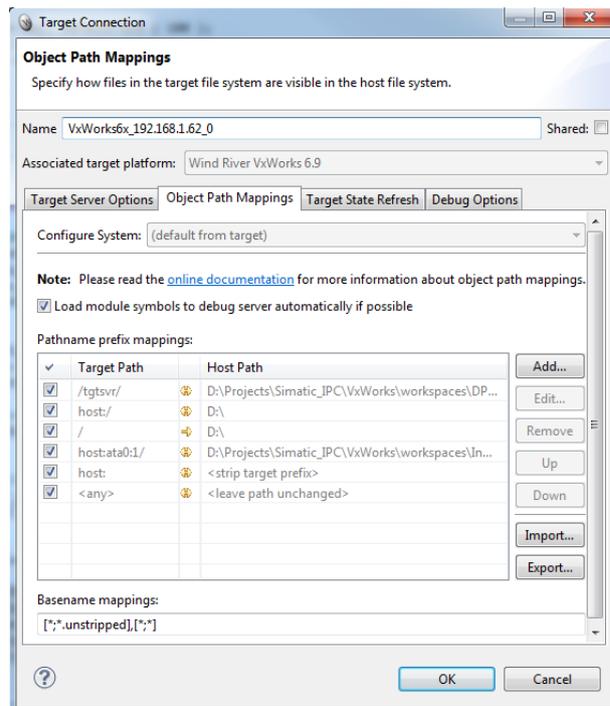
Wenn der Pfad bei "Exec Path on Target" mit "host:" beginnt (siehe oben Schritt 4) und das Aufbauen einer Debug-Verbindung fehl schlägt, gehen Sie wie folgt vor:

1. Öffnen Sie die Ansicht "Remote Systems" und klicken Sie mit der rechten Maustaste auf Ihre Targetverbindung und öffnen Sie den Dialog "Properties".

Die Target Server Einstellungen werden geöffnet.

2. Wählen Sie das Register "Object Path Mappings".

Achten Sie darauf, dass im Bereich "Pathname prefix mappings" der Eintrag "/tgtsvr/" an erster Stelle steht. Benutzen Sie zum Sortieren der Listeneinträge die Schaltflächen rechts neben der Liste.



3. Überprüfen Sie im Dialog "Debug Configuration", ob der Pfad der Option "Exec Path on Target" mit "/tgtsvr/" beginnt.

Arbeiten mit der Shell

9.1 Target Shell Timeout verhindern

Wenn die Target Shell bei einer ungültigen Eingabe für mehrere Minuten blockiert, überprüfen Sie, ob in der boot line der Parameter "host inet" (h=xxx.xxx.xxx.xxx) eingestellt ist. Wenn ja, dann entfernen Sie den Parameter.

boot line Beispiel ohne "host inet":

```
fs(0,0)host:/ata0:1/vxWorks e=192.168.1.62 f=0x8 o=gei0
```

boot line Beispiel mit "host inet":

```
fs(0,0)host:/ata0:1/vxWorks e=192.168.1.62 h=192.168.1.100 f=0x8 o=gei0
```

9.2 Kommandozeile editieren

[ESC] wechselt vom Eingabe- in den Bearbeitungsmodus.

[STRG + u] löscht die Kommandozeile und wechselt zum normalen Eingabemodus.

Key	Description
k	Backwards through history Buffer
j	Forwards through history buffer
h	Move cursor left
l	Move cursor right
a	Append after cursor
A	Append at end of line
i	Insert before cursor
I	Insert at beginning of line
rc	Replace current character with character c
x	Delete current character
dd	Delete entire line
nG	Go to history line n
/string	Search for string

PROFINET-Treiber

10.1 PROFINET-Funktionen

PROFINET-Funktionen

Folgende PROFINET-Funktionen stehen zur Verfügung:

- Controller-Funktionen
- Device-Funktionen
- Produktspezifische Funktionen

Es wird nur eine PROFINET-Schnittstelle unterstützt. Die API-Schnittstelle ist nicht reentrant fähig, d. h. es darf zu einem Zeitpunkt nur eine PROFINET-Funktion aufgerufen werden.

Betriebsarten

Die IO-Base-Anwenderprogrammierschnittstelle unterstützt folgende Betriebsarten:

- RT (Real Time)
- IRT (Isochronous Real Time)

Eine genaue Beschreibung der einzelnen Funktionen finden Sie im Programmierhandbuch "SIMATIC NET PROFINET IO-Base-Anwenderprogrammierschnittstelle" (Datei PGH_IO-Base_0.pdf), das im Verzeichnis "<WIND_BASE>\target\3rdparty\siemens\docs" zu finden ist.

Applikationen können im Kernel Mode (Downloadable Kernel Module Project, DKM) oder im User Mode (Real Time Process Project, RTP) laufen.

PROFINET-Treiber

Es gibt folgende zwei PROFINET-Treiber:

- CP 16xx-Treiber für PROFINET
Der CP 16xx-Treiber für PROFINET unterstützt den CP 1616 onboard und wird nachfolgend als CP 16xx-Treiber bezeichnet. Dabei stehen sowohl Controller- als auch Device-Funktionalität in den Betriebsarten RT und IRT zur Verfügung. Die notwendige PROFINET-Firmware ist im Lieferumfang enthalten.
- PN-Treiber für PROFINET
Der PN-Treiber verwendet für das PROFINET-Protokoll eine Ethernet-Schnittstelle des IPC. Dabei steht nur die PROFINET-Controller-Funktionalität in der Betriebsart RT zur Verfügung.

Beachten Sie, dass Sie nur einen der beiden PROFINET-Treiber einbinden können.

10.2 PROFINET-Treiber

10.2.1 CP 16xx-Treiber

Projektierung

Die Projektierung erfolgt mit NCM PC, SIMATIC STEP 7 bzw. STEP 7 (TIA-Portal). Bei der Projektierung entsteht die Projektierungsdatei "*.xdb". Diese kann mit dem mitgelieferten Programm "pnioload" oder online mit den entsprechenden SIMATIC Tools geladen werden.

Verwenden Sie zur Projektierung als Controller eine PC-Station mit CP 1616-Onboard-Kommunikationsmodul und zur Projektierung als Device "Weitere Feldgeräte > PROFINET IO > I/O > Siemens AG > SIMATIC PC-CP > CP 1616 onboard".

Die Prototypen der PROFINET-Funktionsaufrufe sind in der Datei "vxwpmio.h" im Verzeichnis "<WIND_BASE>\target\3rdparty\siemens\h" enthalten.

Prüfen Sie bei der ersten Inbetriebnahme der CP 1616-Onboard-Schnittstelle für PROFINET, ob die Firmware aktualisiert werden muss. Verwenden Sie die Firmware-Datei "fw16xx-*.fwl" aus folgendem Verzeichnis:

- <WIND_BASE>\target\3rdparty\siemens\profinet\cp16xx\firmware\

Die Firmware der CP 1616-Onboard-Schnittstelle für PROFINET kann mit folgenden Optionen aktualisiert werden:

- pnioload (wird mitgeliefert)
- SIMATIC NET Firmware Lader von STEP 7 (TIA-Portal)

Das erforderliche HSP ist über den Technical Support verfügbar, siehe Kapitel "Service und Support (Seite 129)".

Hinweis

Neustart des Systems nach Aktualisierung der Firmware

Wenn Sie zur Aktualisierung der Firmware das Programm "SIMATIC NET Firmware Lader" verwenden, starten Sie das System nach erfolgreicher Aktualisierung neu.

Hinweis

Betriebsanzeige L2 bei SIMATIC IPC427D rot

Die Betriebsanzeige LED L2 des SIMATIC IPC427D hat die Funktion der Busfault-LED, d. h. die LED ist rot, so lange kein Partner angeschlossen oder die Konfiguration falsch ist.

PROFenergy-Funktionen

PROFenergy-Funktionen werden nicht unterstützt.

10.2.2 PN-Treiber

Projektierung

Der PN-Treiber unterstützt nur den zweiten Netzwerkanschluss "X2P1" des SIMATIC IPC.

Für den vom PN-Treiber verwendeten Netzwerkanschluss darf kein Netzwerk-Treiber von VxWorks konfiguriert werden.

Die Projektierung erfolgt über STEP 7 (TIA-Portal) ab V13 mit HSP0074. Verwenden Sie zur Projektierung ein PC-System mit PN-Driver Kommunikationsmodul. Als PROFINET Interface der PN-Treiber Station muss das Interface-Submodul "Real-time" projektiert werden. Bei der Projektierung entsteht die Projektierungsdatei "*.xml". Diese wird nur beim Start des Treibers ausgewertet. Während des Betriebs ist keine Änderung der Projektierung möglich.

Das erforderliche HSP ist über den Support verfügbar, siehe Kapitel "Service und Support (Seite 129)".

PROFenergy-Funktionen

PROFenergy-Funktionen werden nicht unterstützt.

10.3 IO-Base-Schnittstelle

Eine genaue Beschreibung der IO-Base-Schnittstelle finden Sie im Programmierhandbuch "SIMATIC NET ROFINET IO-Base-Anwenderprogrammierschnittstelle" (Datei "PGH_IO-Base_0.pdf"), das im Verzeichnis "<WIND_BASE>\target\3rdparty\siemens\docs" zu finden ist.

Hinweis

Der Aufruf "SERV_CP_set_type_of_station" wird nicht unterstützt.

10.4 PROFINET-Aufrufe im Downloadable Kernel Modul (DKM) nutzen

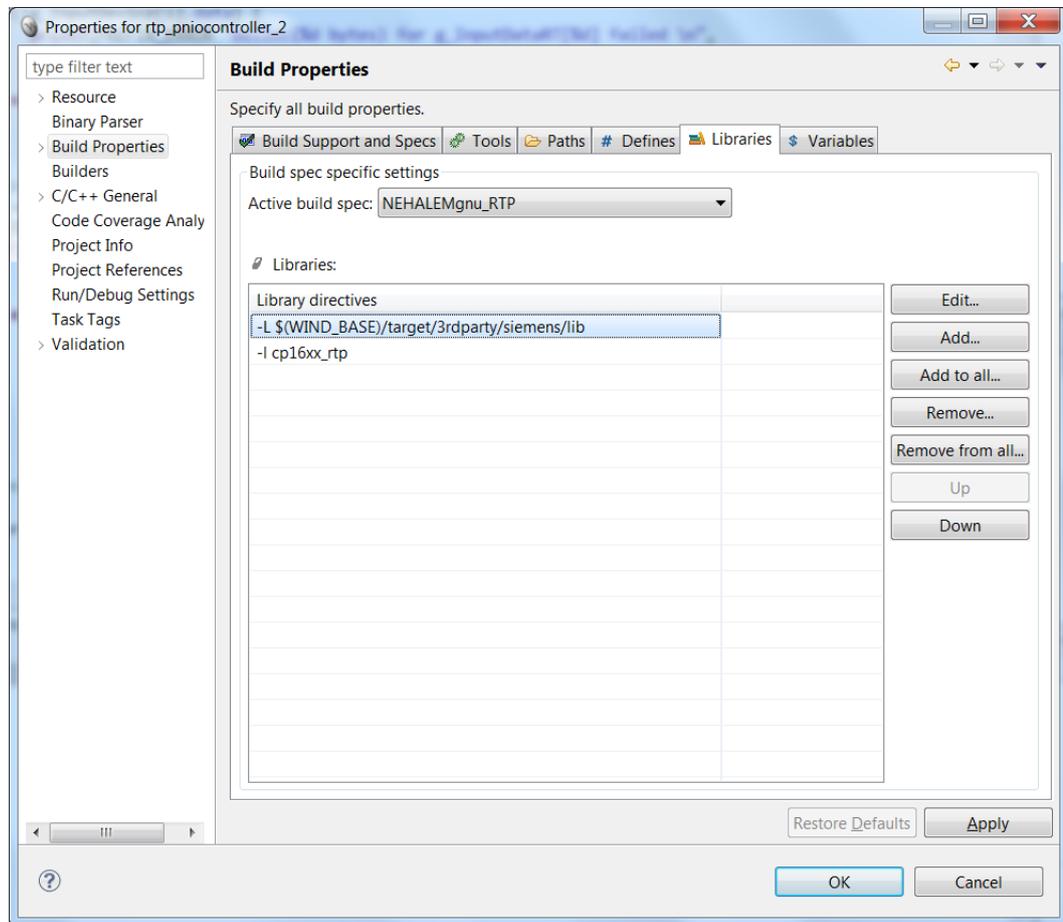
Alle PROFINET-Aufrufe können auch in einem Downloadable Kernel Modul (DKM) genutzt werden. Verwenden Sie dazu die mitgelieferte Headerdatei "vxwvpio.h". Es muss keine Bibliothek eingebunden werden.

10.5 PROFINET-Aufrufe im Real Time Process (User Mode) nutzen

Sie können alle PROFINET-Aufrufe in einem Real Time Process (RTP) nutzen. Verwenden Sie dazu die mitgelieferte Headerdatei "vxwpnio.h".

Bei Verwendung des CP 16xx-Treibers muss die mitgelieferte Bibliothek "libcp16xx_rtp.a", bei Verwendung des PN-Treibers die mitgelieferte Bibliothek "libpn_rtp.a" aus dem Verzeichnis <WIND_BASE>\target\3rdparty\siemens\lib eingebunden werden.

Im RTP-Mode wird fast der gesamte Treibercode auf User-Ebene (Privilege Level 3) ausgeführt.



Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 51 verwendet. Informationen dazu, wie dieser Wert konfiguriert wird, finden Sie im Kapitel "PROFINET-Treiber konfigurieren (Seite 72)".

10.6 Meldungen des PROFINET-Treibers

Meldungen auf der Systemkonsole

```
VxWorks PROFINET Driver for CP 16xx V1.0.x  
(C) Copyright 2014, Siemens AG. All rights reserved.  
* Number of boards: 1  
* CP1616 IRQ:19, MAC: 00.01.02.03.04.05
```

oder:

```
VxWorks PROFINET PN (RT) Driver for standard LAN controller V1.0.x  
(C) Copyright 2014, Siemens AG. All rights reserved.
```

Format der Fehlermeldungen

Der PROFINET-Treiber gibt Warnungen und Fehlermeldungen an der Systemkonsole aus. Diese Warnungen und Fehlermeldungen haben folgendes Format:

```
*** cp16xx: <text>
```

oder:

```
*** pn: <text>
```

Hinweis

Wenden Sie sich im Fehlerfall an den Technical Support, siehe Kapitel "Service und Support (Seite 129)".

Vorgehensweise

Wird der PROFINET-Treiber aufgrund eines Fehlers beendet, starten Sie das System neu.

10.7 PROFINET-Treiber konfigurieren

10.7.1 CP 16xx-Treiber konfigurieren

Die Basis-Priorität (Default: 0) der Tasks und die Syscall-Gruppe (Default: 51) werden in der "Kernel Configuration" unter der Komponente "INCLUDE_PROFINET_CP16XX" angepasst.

Component Configuration			
Description	Name	Type	Value
SGMII interface support for Octeon	INCLUDE_OCTEON_SGMII		
SIMATIC IPC hardware dependent functions	INCLUDE_IPC_HW_FUNCTIONS		
SIMATIC NET CP 16XX PROFINET	INCLUDE_PROFINET_CP16XX		
CP 16XX driver priority	PNIO_CP16XX_PRIORITY	uint	0
CP 16XX syscall group number	PNIO_CP16XX_SYSCALLGROUP	uint	51
SIMATIC NET CP 5622 PROFIBUS	INCLUDE_PROFIBUS_CP5622		
SIMATIC NET PN PROFINET	INCLUDE_PROFINET_LAN		
SMSC LAN9118 VxBus Enhanced Network Driver	INCLUDE_SMSCLAN9118_VXB_END		
SMSC LAN91C111 VxBus Enhanced Network Driver	INCLUDE_SMSCLAN91C111_VXB_END		
SMsC FDC37x driver (default)	DRV_SUPERIO_SMC37X		
SDI EEPROM VxBus driver	DRV_SDI EEPROM		

Die Basis-Priorität wird mit dem Parameter "PNIO_CP16XX_PRIORITY" konfiguriert.

Die Default Basis-Priorität ist "0", also die höchste Priorität. Die Priorität darf einen Wert zwischen 0 und 150 annehmen.

Von der Basis-Priorität sind die Prioritäten der Tasks abhängig, die der PROFINET-Treiber anlegt.

In der folgenden Tabelle ist angegeben, welche Tasks der PROFINET-Treiber anlegt und welche Prioritäten diesen Tasks zugeordnet sind.

Task-Name	Beschreibung	Priorität
PN_DPR0	Task zum Auslesen der Daten aus dem DPR der Baugruppe	Basis-Priorität + 99
PN_CMD	Task zur Verarbeitung asynchroner Kommandos, wie z. B. Reset von Baugruppen	Basis-Priorität + 99
PN_TIMER_TASK	Hilfs-Task für den Firmware Watchdog	Basis-Priorität
PN_SYNCH0_<x>	Task zur Verarbeitung des SYNCH-Events der Baugruppe	Basis-Priorität + 99
PN_ALARM0_<x>	Task zur Verarbeitung des ALARM-Events der Baugruppe	Basis-Priorität + 99
PN_MODIND0_<x>	Task zur Verarbeitung des MODIND-Events der Baugruppe	Basis-Priorität + 99
PN_DATAREC0_<x>	Task zur Verarbeitung des DATAREC-Events der Baugruppe	Basis-Priorität + 99
PN_WD0_<x>	Task zur Verarbeitung des WATCHDOG-Events der Baugruppe	Basis-Priorität
PN_STARTOP0_<x>	Task zur Verarbeitung des STARTOP-Events der Baugruppe	Basis-Priorität + 1
PN_OPFAULT0_<x>	Task zur Verarbeitung des OPFAULT-Events der Baugruppe	Basis-Priorität + 1

Task-Name	Beschreibung	Priorität
PN_NCYCLE0_<x>	Task zur Verarbeitung des NEWCYCLE-Events der Baugruppe	Basis-Priorität + 1
PN_CPINFO	Task zur Verarbeitung der SERV_CP_info-Aufrufe	Basis-Priorität + 99
PN_READDOWN	Task zum FW-Laden und Reset der Baugruppe	Basis-Priorität + 99

Hinweis

Folgende Tasks werden unmittelbar nach dem Start des PROFINET-Treibers angelegt:

- PN_DPR0
- PN_CMD
- PN_TIMER_TASK

Alle anderen Tasks werden erst bei Verwendung der entsprechenden Events durch eine PROFINET-Applikation angelegt.

Tasks mit einem "_<x>" im Namen sind an eine Controller- und/oder Device-Instanz gebunden und können daher mehrfach auftreten.

<x> steht hier für eine Ziffer zwischen 0 und 9.

Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 51 verwendet. Dieser Wert kann mit dem Parameter "PNIO_CP16XX_SYSCALLGROUP" geändert werden, falls die Gruppe 51 schon benutzt wird.

Die Syscall-Gruppen können wie folgt angezeigt werden:

```
-> syscallShow
```

```
Group Name          GroupNo    NumRtns    Rtn Tbl Addr
-----
STANDARDGroup      8          64         0x00000000006772a0
PROFIBUS CP5622    13         34         0x000000000044f8d90
PROFINET CP16XX    51         8          0x000000000044f8e90
```

Falls die Syscall-Gruppe geändert wird, muss in der Applikation die Variable "syscallGroup_pnio" mit dem neuen Wert vorbelegt werden.

```
extern unsigned int syscallGroup_pnio;
```

```
...
```

```
syscallGroup_pnio = 51; //new value
```

10.7.2 PN-Treiber konfigurieren

Der Pfad für die Konfigurationsdatei (Default: "/ata0:1/Station_1.PN Driver_1.PNDriverConfiguration.xml"), die Priorität der Interrupttask (Default: 0) und die Syscall-Gruppe (Default: 51) können in der "Kernel Configuration" unter der Komponente "INCLUDE_PROFINET_LAN" angepasst werden.

Description	Name	Type	Value
SGMII interface support for Octeon	INCLUDE_OCTEON_SGMII		
SIMATIC IPC hardware dependent functions	INCLUDE_IPC_HW_FUNCTIONS		
SIMATIC NET CP 16XX PROFINET	INCLUDE_PROFINET_CP16XX		
SIMATIC NET CP 5622 PROFIBUS	INCLUDE_PROFIBUS_CP5622		
SIMATIC NET PN PROFINET	INCLUDE_PROFINET_LAN		
PN driver configuration file	PN_CONFIGURATION_FILE	string	"/ata0:1/Station_1.PN Driver_1.PNDriverConfiguration.xml"
PN interrupt task priority	PN_PRIORITY	uint	0
PN syscall group number	PN_SYSCALLGROUP	uint	51
SMSC LAN9118 VxBus Enhanced Network Driver	INCLUDE_SMSCLAN9118_VXB_END		
SMSC LAN91C111 VxBus Enhanced Network Driver	INCLUDE_SMSCLAN91C111_VXB_END		
SMC FDC37x driver (default)	DRV_SIPFRIO_SMC'FDC'37X		

Die SIMATIC IPCs haben in der Regel zwei Ethernet-Schnittstellen.

VxWorks benutzt die erste Ethernet-Schnittstelle, die am PCI-Bus gefunden wird, falls IFCONFIG_1 konfiguriert ist.

Der PN-Treiber benutzt den zweiten Ethernet-Anschluss.

Wenn der PN-Treiber benutzt wird, muss die zweite Ethernet-Schnittstelle von VxWorks unbenutzt sein, d. h. IFCONFIG_2 darf nicht konfiguriert sein.

Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 51 verwendet. Dieser Wert kann mit dem Parameter "PN_SYSCALLGROUP" geändert werden, falls die Gruppe 51 schon benutzt wird.

Die Syscall-Gruppen können wie folgt angezeigt werden:

-> syscallShow

Group Name	GroupNo	NumRtns	Rtn Tbl Addr
STANDARDGroup	8	64	0x0000000006772a0
PROFIBUS CP5622	13	34	0x00000000044f8d90
PROFINET PN	51	8	0x00000000044f8e90

Falls die Syscall-Gruppe geändert wird, muss in der Applikation die Variable "syscallGroup_pn" mit dem neuen Wert vorbelegt werden.

```
extern unsigned int syscallGroup_pn;
...
syscallGroup_pn = 51; //new value
```

PROFIBUS-Treiber

11.1 PROFIBUS-Funktionen

Der PROFIBUS-Treiber unterstützt den PROFIBUS-Anschluss der SIMATIC IPC. Der PROFIBUS-Anschluss enthält einen Prozessor, auf dem die Firmware "fw_5612.bin" läuft. Die Kommunikation zum Anwender erfolgt über einen gemeinsamen Speicher.

Es wird die Master-Funktionalität nach PROFIBUS-DP-Master Klasse 1, die "aktive" Slave Funktionalität und das PROFIBUS-Protokoll "DP-V0" unterstützt.

Als API-Schnittstelle wird vom Treiber die sogenannte DP-Base-Schnittstelle angeboten.

Es wird nur eine PROFIBUS-Schnittstelle unterstützt. Die API-Schnittstelle ist nicht reentrant fähig, d. h. es darf zu einem Zeitpunkt nur eine PROFIBUS-Funktion aufgerufen werden.

Applikationen können sowohl im Kernel Mode (DKM) als auch im User Mode (Real Time Process) laufen. Die Prototypen der PROFIBUS-Funktionsaufrufe sind in der Datei "vxwdp.h" enthalten.

11.2 DP-Base-Schnittstelle

11.2.1 Programmierhandbuch zur DP-Base-Schnittstelle

Eine genaue Beschreibung der DP-Base-Schnittstelle finden Sie im Programmierhandbuch "SIMATIC NET Programmierschnittstelle DP-Base für CP 5613/CP 5614" (Datei "mn_dp_base_api_0.pdf"), das im Verzeichnis "`<WIND_BASE>\target\3rdparty\siemens\docs`" zu finden ist.

11.2.2 DP-Slave-Daten senden (Aktualisierung der Informationen im Programmierhandbuch)

Die im Programmierhandbuch zur DP-Base-Schnittstelle beschriebenen Informationen bezüglich DP-Slave-Daten senden haben sich geändert, siehe Kapitel 5.3.13 "DP-Slave-Daten senden" im "Programmierhandbuch zur DP-Base-Schnittstelle (Seite 75)".

Die Sendedaten des DP-Slave können nicht mittels Speicherzugriff auf das DP-RAM in das Ausgangsabbild des Slave-Moduls übertragen werden. Es muss stattdessen die nachstehend beschriebene Funktion "DPS_write()" verwendet werden.

Beispiel aus dem Programmierhandbuch

```
#define SLAVE_CTRL_ADDR 127

length = 1;

offset = 0;

memcpy(&dpr_ptr_slave->pi.slave_out[SLAVE_CTRL_ADDR].data[offset], &tmp, length);

// write slave data

dpr_ptr_slave->ctr.D_out_slave_adr=SLAVE_CTRL_ADDR;
```

Beispiel für PROFIBUS-Treiber unter VxWorks

```
length = 1;

offset = 0;

ret=DPS_write(slave_user_handle,length,offset,&tmp,&error);

if(ret!=DP_OK)

{

    print_error_text ("DPS_write failed\n", &error);

}
```

DPS_write()

Die Funktion "DPS_write()" beschreibt den Ausgabebereich des Slave-Moduls mit neu zu sendenden Daten, die der übergeordnete Master über PROFIBUS als Eingänge liest.

Include

```
<dps_5612.h>
```

Syntax

```
DPR_DWORD DP_CODE_ATTR DPS_write

(

    DPR_DWORD user_handle,           /* in */
    DPR_DWORD length,               /* in */
    DPR_DWORD offset,               /* in */
    DPR_BYTE DP_MEM_ATTR* buffer,   /* in */
    DP_ERROR_T DP_MEM_ATTR* error   /* out */
);
```

Parameter

Parameter	Bedeutung
user_handle	User-Handle, das beim Aufruf DPS_open vergeben wurde
length	Länge der zu sendenden Daten
offset	Offset der Daten
buffer	Zeiger auf die zu sendenden Daten
error	Adresse einer vom DP-Anwenderprogramm bereitgestellten Struktur vom Typ DP_ERROR_T. Die Struktur enthält im Fehlerfall Details zur Fehlerursache (siehe DP-Base Beschreibung).

Rückgabewert

Wert	Bedeutung
DP_OK	Funktion erfolgreich ausgeführt
DP_ERROR_REQ_PAR und error->error_code == DP_RET_PAR_USR_HNDL	Der Parameter user_handle ist ungültig
DP_ERROR_REQ_PAR und error->error_code == DPS_RET_PAR_STATE	Der Parameter buffer ist ungültig
DP_ERROR_CI und error->error_code == "andere"	Fehlerhafter Abschluss der Funktion

11.3 PROFIBUS-Aufrufe im Downloadable Kernel Modul (DKM) nutzen

Alle PROFIBUS-Aufrufe können in einem Downloadable Kernel Modul (DKM) genutzt werden. Verwenden Sie dazu die mitgelieferte Headerdatei "vxwdp.h". Es muss keine Bibliothek eingebunden werden.

11.4 PROFIBUS-Aufrufe im Real Time Process (User Mode) nutzen

Sie können alle PROFIBUS-Aufrufe auch in einem Real Time Process (RTP) nutzen. Verwenden Sie dazu die mitgelieferte Headerdatei "vxwdp.h". Es muss keine Bibliothek eingebunden werden.

Da beim Übergang vom User Mode in den Kernel Mode nur maximal 8 Parameter möglich sind, werden folgende Parameter beim Aufruf "DPS_open" nicht ausgewertet:

Parameter	Gesetzter Wert
cp_name	"CP5612_1"
addr_change	0
baud_rate	DPS_BD_AUTO_DETECT

Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 13 verwendet. Informationen dazu, wie dieser Wert konfiguriert wird, finden Sie im Kapitel "PROFIBUS konfigurieren (Seite 79)".

11.5 Meldungen des PROFIBUS-Treibers

Hochlaufmeldung des Treibers

```
VxWorks PROFIBUS DP Driver for CP 5622 V1.0.x  
(C) Copyright 2014, Siemens AG. All rights reserved.
```

Format der Fehlermeldungen

Der PROFIBUS-Treiber gibt Warnungen und Fehlermeldungen an der Systemkonsole aus. Diese Warnungen und Fehlermeldungen haben folgendes Format:

```
*** cp5622: <text>
```

Hinweis

Wenden Sie sich im Fehlerfall an den Technical Support, siehe Kapitel Service und Support (Seite 129).

11.6 PROFIBUS konfigurieren

Der Pfad für die Firmwaredatei (Default: "/ata0:1"), die Priorität der Interrupttask (Default: "0") und die Syscall-Gruppe (Default: "13") können in der "Kernel Configuration" unter der Komponente "INCLUDE_PROFIBUS_CP5622" angepasst werden.

Component Configuration			
Description	Name	Type	Value
SGMII interface support for Octeon	INCLUDE_OCTEON_SGMII		
SIMATIC IPC hardware dependent functions	INCLUDE_IPC_HW_FUNCTIONS		
SIMATIC NET CP 16XX PROFINET	INCLUDE_PROFINET_CP16XX		
SIMATIC NET CP 5622 PROFIBUS	INCLUDE_PROFIBUS_CP5622		
CP 5622 firmware file path	DP_CP5622_FIRMWARE_PATH	string	"/ata0:1"
CP 5622 interrupt task priority	DP_CP5622_PRIORITY	uint	0
CP 5622 syscall group number	DP_CP5622_SYSCALLGROUP	uint	13
SIMATIC NET PN PROFINET	INCLUDE_PROFINET_LAN		
SMSC LAN9118 VxBus Enhanced Network Driver	INCLUDE_SMSCAN9118_VXB_END		

Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 13 verwendet.

Dieser Wert kann mit dem Parameter "DP_CP5622_SYSCALLGROUP" geändert werden, falls die Gruppe 13 schon benutzt wird.

Die Syscall-Gruppen können wie folgt auf dem Zielsystem angezeigt werden:

```
-> syscallShow
```

Group Name	GroupNo	NumRtns	Rtn Tbl Addr
STANDARDGroup	8	64	0x00000000006772a0
PROFIBUS CP5622	13	34	0x00000000044f8d90

Falls die Syscall-Gruppe geändert wird, muss in der Datei "dp_rtp.h" im Verzeichnis "<WIND_BASE>\target\3rdparty\siemens\h" das Define "MY_OWN_SCG" auch entsprechend geändert werden.

Beispiel:

```
#define MY_OWN_SCG 13
```

11.7 Projektierung erstellen

Für den Betrieb der PROFIBUS-Schnittstelle wird von der Firmware eine Datenbasis vorausgesetzt, die grundlegende Informationen über die PROFIBUS-Projektierung des CP enthält.

Die Firmware prüft im Slave-Betrieb die Gültigkeit der Datenbasis und vergleicht die Ist-Konfiguration mit der Soll-Konfiguration.

Für die Projektierung der PROFIBUS-Schnittstelle werden folgende Projektierungstools unterstützt:

Tool	Einschränkungen/Fehler
SIMATIC Manager STEP7 V5.5 (Servicepack 3, Hotfix 3)	Keine Einschränkungen.
STEP 7 (TIA-Portal) ab V13	Keine Einschränkungen.

Damit für die Onboard-Schnittstelle für PROFIBUS (kompatibel mit CP 5622) eine Konfigurationsdatei exportiert werden kann, muss eine Projektierung für die Baugruppe CP 5613-A2 verwendet werden.

Projektierungsbeispiele finden Sie in folgendem Verzeichnis:
<WIND_BASE>\target\3rdparty\siemens\examples.

11.7.1 Mit SIMATIC Manager STEP7 projektieren

Master projektieren

Projektieren Sie den Master als SIMATIC PC-Station.

1. Öffnen Sie den Hardwarekatalog und navigieren Sie zum entsprechenden Prozessor. Diesen finden Sie in folgendem Verzeichnis:

SIMATIC PC-Station->CP-PROFIBUS->CP 5613 A2->SW V7.1.2.

2. Fügen Sie dem Master noch eine Applikation hinzu. Diese finden Sie in folgendem Verzeichnis:

SIMATIC PC-Station->Benutzer_Applikation->Applikation.

Wählen Sie die Version "SW V8.1.2" .

3. Wählen Sie für den CP5613A2 unter "Eigenschaften->Betriebsart" die Einstellung "DP-Master" und "DPV0-kompatibel".
4. Markieren Sie unter "Eigenschaften->Betriebsart" die Option "LDB Datei erzeugen" und vergeben Sie Name und Pfad.

Für den Betrieb als Master wird eine Datenbasis (.ldb-Datei) vorausgesetzt, die grundlegende Informationen über die PROFIBUS-Projektierung des Prozessors enthält.

5. Führen Sie die Aktion "Speichern und Übersetzen" aus.

Die .ldb-Datei wird erzeugt.

Hinweis

Die Onboard-Schnittstelle für PROFIBUS unterstützt maximal 64 DP-Slaves.

Slave projektieren

Projektieren Sie den Slave als SIMATIC PC-Station.

1. Öffnen Sie den Hardwarekatalog und navigieren Sie zum entsprechenden Prozessor. Diesen finden Sie in folgendem Verzeichnis:

SIMATIC PC-Station->CP-PROFIBUS->CP 5613 A2->SW V7.1.2.

2. Wählen Sie für den CP5613A2 unter "Eigenschaften->Betriebsart" die Einstellung "DP-Slave" und markieren Sie die Option "LDB Datei erzeugen".

Für den Betrieb als Slave wird eine Datenbasis (.ldb-Datei) vorausgesetzt, die grundlegende Informationen über die PROFIBUS-Projektierung des Prozessors enthält.

3. Führen Sie die Aktion "Speichern und Übersetzen" aus.

Die .ldb-Datei wird erzeugt.

4. Wechseln Sie in die Master-PC-Station und navigieren Sie im Hardwarekatalog zum neu erstellten Slave in folgendem Verzeichnis:

"PROFIBUS-DP->Bereits projektierte Stationen" als "SIMATIC PC (CP 56x3)".

5. Wählen Sie hier den Eintrag "SIMATIC PC (CP 56x3) / DPV0."

6. Wählen Sie im Dialog "Eigenschaften – DP-Slave->Kopplung" den projektierten CP 5613 A2 Slave aus.

7. Klicken Sie auf "Koppeln".

Der Slave wird in das PROFIBUS-Netz aufgenommen. Sie können nun die benötigten Eingangsmodule und Ausgangsmodule hinzufügen.

Hinweis

Weitere Informationen zur Projektierung des CP 5613 A2 als DP-Slave finden Sie im SIEMENS Industry Online Support in folgendem Handbuch:

"Industrielle Kommunikation SIMATIC NET PC-Stationen in Betrieb nehmen - Anleitung und Schnelleinstieg (<http://support.automation.siemens.com/WW/view/de/13542666>)" (Kapitel 4.2.8 "PC-Station als DP-Slave projektieren").

11.7.2 Mit STEP 7 (TIA-Portal) projektieren

Master projektieren

Projektieren Sie den Master als SIMATIC PC-Station.

1. Wechseln Sie zur Netzansicht und öffnen Sie die Gerätesicht der Master PC-Station.
2. Fügen Sie den CP 5613 A2 in die PC-Station ein.

Den richtigen CP finden Sie im Hardwarekatalog unter "Kommunikationmodule->PROFIBUS->CP 5603, CP 5613 A2."

3. Wählen Sie die Version "SW V7.1.2 SP2."
4. Öffnen Sie die Eigenschaften des CP5613 A2 und wählen Sie das Register "Allgemein".
5. Erstellen Sie unter "PROFIBUS-Adresse" ein neues PROFIBUS-Netz und stellen Sie die Busparameter des Masters ein.

- Wählen Sie bei "Betriebsart" die Einstellung "DP-Master"
- Wählen Sie bei "DP-Mode" die Einstellung "DPV0-kompatibel".

6. Markieren Sie unter "LDB-Konfiguration" die Option "LDB Datei generieren" .

Für den Betrieb als Master wird von der Firmware wird eine Datenbasis (.ldb-Datei) vorausgesetzt, die grundlegende Informationen über die PROFIBUS-Projektierung des CP enthält.

7. Führen Sie in der Netzansicht die Aktion "Übersetzen" aus.

Die .ldb-Datei wird erzeugt.

Hinweis

Die Onboard-Schnittstelle für PROFIBUS unterstützt maximal 64 DP-Slaves.

Slave projektieren

Projektieren Sie den Slave als SIMATIC PC-Station.

1. Wechseln Sie zur Netzansicht und öffnen Sie die Gerätesicht der Slave PC-Station.
2. Fügen Sie den CP 5613 A2 in die PC-Station ein.

Den richtigen CP finden Sie im Hardwarekatalog unter "Kommunikationmodule->PROFIBUS->CP 5603, CP 5613 A2".

Wählen Sie die Version "SW V7.1.2 SP2".

3. Öffnen Sie die Eigenschaften des CP5613 A2 und wählen Sie das Register "Allgemein".
4. Erstellen Sie unter "PROFIBUS-Adresse" das PROFIBUS-Netz des Masters und stellen Sie die Busparameter für den Slave ein.
 - Wählen Sie bei "Betriebsart" die Einstellung "DP-Slave"
 - Wählen Sie bei "DP-Mode" die Einstellung "DPV0-kompatibel".
 - Wählen Sie bei "Zugewiesener DP-Master" den PROFIBUS-Master aus.
 - Fügen Sie, falls nötig, unter "Betriebsart->I-Slave-Kommunikation" Eingabe- und Ausgabemodule hinzu.
5. Markieren Sie unter "LDB-Konfiguration" die Option "LDB Datei generieren" .

Für den Betrieb als Slave wird von der Firmware eine Datenbasis (.ldb-Datei) vorausgesetzt, die grundlegende Informationen über die PROFIBUS-Projektierung des CP enthält.

6. Führen Sie in der Netzansicht die Aktion "Übersetzen" aus.

Die .ldb-Datei wird erzeugt.

Hinweis

Weitere Informationen zur Projektierung des CP 5613 A2 als DP-Slave finden Sie im SIEMENS Industry Online Support in folgendem Handbuch:

"Industrielle Kommunikation SIMATIC NET PC-Stationen in Betrieb nehmen - Anleitung und Schnelleinstieg (<http://support.automation.siemens.com/WW/view/de/13542666>)" (Kapitel 4.3.7 "PC-Station als DP-Slave projektieren").

Hardwarenahe Funktionen

Es stehen die folgenden Funktionen für den Zugriff auf die Hardware-Komponenten des IPC zur Verfügung:

- **CPU-Typ** auslesen
- **Temperaturinformation** auslesen
- **Lüfterstatus** auslesen
- **Batteriestatus** auslesen
- **Watchdog** starten und triggern
- **Betriebsstundenzähler** auslesen und aktualisieren

Der Betriebsstundenzähler erfasst die Betriebszeit seit Inbetriebnahme.

In Verbindung mit einem Schwellwert kann z. B. ein Wartungsintervall eingestellt werden.

- User **LEDs** setzen
- Remanenzspeicher als **RAMDISK** verwenden

Die SIMATIC IPCs verfügen zum Teil über einen gepufferten Remanenzspeicher (NVRAM). Der Speicherinhalt bleibt nach dem Ausschalten erhalten. Ob Ihr SIMATIC IPC mit einem Remanenzspeicher ausgestattet ist und Informationen über die Größe dieses Speichers, entnehmen Sie Ihren Bestellunterlagen.

- **SMART-Status** auslesen
- **Echtzeituhr** auslesen und aktualisieren

12.1 Hardwarenahe Funktionen im Downloadable Kernel Modul nutzen

Alle Aufrufe der hardwarenahen Funktionen können auch in einem Downloadable Kernel Modul (DKM) genutzt werden. Verwenden Sie dazu die mitgelieferte Headerdatei "cpuex.h". Sie müssen keine Bibliothek einbinden.

12.2 Hardwarenahe Funktionen im Real Time Process (User Mode) nutzen

Alle Aufrufe der hardwarenahen Funktionen können auch in einem Real Time Process (User Mode) genutzt werden. Binden Sie dazu die mitgelieferte Headerdatei "cpuex.h" ein. Sie müssen keine Bibliothek einbinden.

Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 14 verwendet. Informationen dazu, wie dieser Wert konfiguriert wird, finden Sie im Kapitel "Hardwarenahe Funktionen konfigurieren (Seite 87)".

12.3 Meldungen des Treibers

Hochlaufmeldung des Treibers am Beispiel IPC427D

Das folgende Beispiel gilt für einen IPC427D.

```
VxWorks SIMATIC IPC hardware dependent functions V1.0.x
```

```
(C) Copyright 2014, Siemens AG. All rights reserved.
```

```
SIMATIC IPC427D (type=13) detected
```

Fehlermeldungen des Treibers

Der Treiber gibt Warnungen und Fehlermeldungen an der Systemkonsole aus. Diese Warnungen und Fehlermeldungen haben folgendes Format:

```
*** cpux: <text>
```

Hinweis

Wenden Sie sich im Fehlerfall an den Technical Support, siehe Kapitel Service und Support (Seite 129).

12.4 Hardwarenahe Funktionen konfigurieren

Einige Parameter für den Betriebsstundenzähler und die Syscall-Gruppe (Default: 14) sind konfigurierbar. Diese Parameter werden bei der Konfiguration der Komponente beim Erzeugen eines VxWorks-Images angezeigt.

Für die Konfiguration des Betriebsstundenzählers und der RAMDISK stehen folgende Werte zur Verfügung:

Components

Component Configuration				
Description	Name	Type	Value	
SGMII interface support for Octeon	INCLUDE_OCTEON_SGMII			
SIMATIC IPC DiagMonitor Agent	INCLUDE_DIAGMONITOR_AGENT			
SIMATIC IPC hardware dependent functions	INCLUDE_IPC_HW_FUNCTIONS			
IPC RAMDISK formatting	CPUEX_RAMDISK_MODE	BOOL	FALSE	
IPC RAMDISK size	CPUEX_RAMDISK_SIZE	uint	0xFFFFFFFF	
IPC operation hours counter cycle	CPUEX_OP_HOURS_COUNTER_CYCLE	uint	60	
IPC operation hours counter file path	CPUEX_OP_HOURS_COUNTER_PATH	string	"/ata0:1"	
IPC operation hours counter priority	CPUEX_OP_HOURS_COUNTER_PRIORITY	uint	191	
IPC syscall group number	CPUEX_SYSCALLGROUP	uint	14	
SIMATIC NET CP 16XX PROFINET	INCLUDE_PROFINET_CP16XX			
SIMATIC NET CP 5622 PROFIBUS	INCLUDE_PROFIBUS_CP5622			
SIMATIC NET PN PROFINET	INCLUDE_PROFINET_LAN			
SMSC LAN9118 VxBus Enhanced Network Driver	INCLUDE_SMSCLAN9118_VXB_END			
SMSC LAN91C111 VxBus Enhanced Network Driver	INCLUDE_SMSCLAN91C111_VXB_END			

Synopsis

Driver for SIEMENS SIMATIC IPC hardware dependent functions (operation hours counter, temperature, fan, battery, watchdog and S.M.A.R.T. status)

Defined at:

<C:\vxworks-6.9\target\config\comps\vxWorks\40drvSimaticIPC.cdf>

Overview | Bundles | Components

Name	Beschreibung	Wertebereich	Default Wert	Bemerkung
CPUEX_OP_HOURS_COUNTER_PRIORITY	Priorität der Task (Betriebsstundenzähler)	0 ... 255	191	Ist die Priorität 255, dann wird kein Betriebsstundenzähler angelegt. Bei Priorität ungleich 255 wird der Betriebsstundenzähler angelegt und initialisiert.
CPUEX_OP_HOURS_COUNTER_CYCLE	Aktualisierungsintervall für das Schreiben des Betriebsstundenzählers in die Datei "ophours.vxw"	1 ... 0xFFFFFFFF	60	Zeit in Minuten
CPUEX_OP_HOURS_COUNTER_PATH	Pfad zur Ablage der Datei ophours.vxw	String (max. 255 Zeichen)	/ata0:1	
CPUEX_RAMDISK_MODE	Formatierung durchführen	TRUE oder FALSE	FALSE	Legt fest, ob RAMDISK beim/nach dem Booten formatiert werden soll.
CPUEX_RAMDISK_SIZE	Größe der RAMDISK	0x20000 ... 0xFFFFFFFF	0xFFFFFFFF	Mit dem Wert 0 wird keine RAMDISK angelegt. Mit dem Wert 0xFFFFFFFF wird das gesamte SRAM für die RAMDISK verwendet.

Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 14 verwendet. Falls die Gruppe 14 schon benutzt wird, kann dieser Wert mit dem Parameter "CPUEX_SYSCALLGROUP" geändert werden.

Die Syscall-Gruppen können wie folgt angezeigt werden.

-> syscallShow

Group Name	GroupNo	NumRtns	Rtn Tbl Addr
STANDARDGroup	8	64	0x00000000006cb080
VXWORKSGroup	9	59	0x00000000006cb480
SIMATIC IPC HW	14	8	0x00000000006d7c60

Falls die Syscall-Gruppe geändert wird, muss in der Datei "cpuex.h" das Define "MY_OWN_SCG" auch entsprechend geändert werden.

Beispiel:

```
#define MY_OWN_SCG 14
```

Remanenzspeicher als RAMDISK verwenden

Der Remanenzspeicher kann als RAMDISK verwendet werden. Die Größe des verwendeten Speicherbereichs wird durch den Konfigurationsparameter CPUEX_RAMDISK_SIZE eingestellt. Wird die RAMDISK durch die Parametrierung einer Größe > 0 konfiguriert, dann erhält sie die Laufwerksbezeichnung "/ram:0". Wenn die RAMDISK noch nicht formatiert ist, wird die Formatierung automatisch durchgeführt. Die Formatierung nach dem Booten kann durch den Konfigurationsparameter "CPUEX_RAMDISK_MODE" auch erzwungen werden.

12.5 Zugriffsfunktionen für hardwarenahe Funktionen

Folgende Zugriffsfunktionen stehen zur Verfügung:

Funktion	Beschreibung
GetCPUTypeDMI()	Ermitteln des Typs der verwendeten CPU aus DMI-Informationen
ReadTemperature()	Auslesen der verschiedenen Temperatur-Sensoren im SIMATIC IPC
ReadFanStatus()	Auslesen des Lüfterstatus
ReadBatteryStatus()	Auslesen des Batteriestatus
SetOpHoursCounter()	Setzen der Schwelle für den Betriebsstundenzähler
GetOpHoursCounter()	Auslesen des Betriebsstundenzählerstands und des Schwellenwertes
UpdateOpHoursCounter()	Aktualisierung des Betriebsstundenzählers
StartWatchdog()	Start des Hardware-Watchdogs
TriggerWatchdog()	Triggern des Hardware-Watchdogs
SetUserLED()	User LED setzen
ResetUserLED()	User LED zurücksetzen
GetSMARTStatus()	Auslesen des S.M.A.R.T.-Status eines Datenträgers
GetCMOSDateTime()	Auslesen der Echtzeit-Uhr (Realtime Clock)
SetCMOSDateTime()	Setzen der Echtzeit-Uhr (Realtime Clock)

12.5.1 GetCPUTypeDMI

Die Funktion "GetCPUTypeDMI()" liefert den Typ der verwendeten CPU zurück.

Include

```
<cpuex.h>
```

Syntax

```
int GetCPUTypeDMI
(
    char *buffer,
    unsigned int bufferLength,
    unsigned int *type
);
```

Parameter

Parameter	Bedeutung	
buffer	Zeiger auf Feld für die Rückgabe der CPU-Typbezeichnung, z. B. SIMATIC IPC427D	
bufferLength	Länge des Felds "buffer"	
type	Zeiger auf Kennung der verwendeten CPU:	
	13:	IPC427D
	25:	IPC627D
	26	IPC827D
	35:	IPC647D
	36:	IPC847D
	41:	IPC227E
	42:	IPC277E

Rückgabewert

Wert	Bedeutung
0 (OK)	Funktion erfolgreich ausgeführt
-1	Zugriffsfehler
-2	Der mit bufferLength übergebene Wert war zu klein; die CPU-Typbezeichnung ist nicht vollständig.

12.5.2 ReadTemperature

Die Funktion "ReadTemperature()" liest abhängig vom Parameter "type" die Temperatur eines bestimmten Sensors des SIMATIC IPC aus.

Mit dieser Funktion kann der Anwender, z. B. von einer Task aus, regelmäßig die Temperatur auslesen und bei Erreichen einer bestimmten Temperaturgrenze eingreifen.

Include

```
<cpuex.h>
```

Syntax

```
int ReadTemperature
(
    unsigned int type,
    unsigned char *buffer
);
```

Parameter

Parameter	Bedeutung	
type	TEMPERATURE_CPU	Temperatur der CPU
	TEMPERATURE_MAINBOARD	Temperatur der Grundplatine
	TEMPERATURE_MEMORY	Temperatur des Speichers
	TEMPERATURE_POWER_SUPPLY	Temperatur des Netzteils
	TEMPERATURE_CHASSIS	Temperatur des Gehäuses
buffer	gelesener Temperaturwert in Grad Celsius	

Rückgabewert

Wert	Bedeutung
0 (OK)	Funktion erfolgreich ausgeführt
-1	Aufruf wird für diese CPU nicht unterstützt
-2	Hardware-Fehler, z. B. Lesefehler am SMB (System Management Bus)
-3	Diesen Temperatursensor gibt es nicht bei der verwendeten CPU

Die SIMATIC IPC verfügen über folgende Temperatursensoren:

Gerät	Temperatursensoren
IPC627D IPC647D IPC827D IPC847D	CPU, Gehäuse, Speicher, Grundplatine
IPC227E IPC277E IPC427D	CPU, Speicher, Grundplatine

Programmierbeispiel

```
unsigned char buffer;  
  
int ret = ReadTemperature (TEMPERATURE_CPU, &buffer );  
  
if (ret == OK) {  
    if (buffer >= 0x80) {  
        printf( "CPU Temperatur = -%u °C\n",  
                (unsigned char) (buffer & 0x7F) );  
    }  
    else {  
        printf( "CPU Temperatur = %u °C\n",buffer );  
    }  
}
```

12.5.3 ReadFanStatus

Die Funktion "ReadFanStatus()" liest den Lüfterstatus über den SMB (System Management Bus) aus. Mit dieser Funktion kann der Anwender, z. B. von einer Task aus, regelmäßig den Lüfterstatus auslesen und bei Lüfterausfall eingreifen.

Diese Funktion ist für die CPUs folgender SIMATIC IPCs verfügbar:

- SIMATIC IPC627D
- SIMATIC IPC647D
- SIMATIC IPC827D
- SIMATIC IPC847D

Include

```
<cpuex.h>
```

Syntax

```
int ReadFanStatus (void);
```

Rückgabewert

Wert	Bedeutung
0 (OK)	Alle Lüfter funktionieren
>0	Bitmaske, die angibt, welche Lüfter ausgefallen sind:
	Bit 0: FAN_ERROR_MASK Mindestens ein Lüfter ist defekt
	Bit 1: FAN_CHASSIS_BACK_MASK Gehäuselüfter hinten ausgefallen
	Bit 2: FAN_POWER_SUPPLY_MASK Netzteil Lüfter ausgefallen
	Bit 3: FAN_CHASSIS_FRONT_L_MASK Gehäuselüfter vorn / vorn links ausgefallen
	Bit 4: FAN_HDD_MASK Gehäuselüfter innen ausgefallen
	Bit 5: FAN_CHASSIS_FRONT_R_MASK Gehäuselüfter vorn rechts ausgefallen
-1	Aufruf wird für diese CPU nicht unterstützt
-2	Hardware-Fehler (z. B. Lesefehler am SMB)

Hinweis

Bei Ausfall mehrerer Lüfter können im Rückgabewert mehrere Bits gesetzt sein. Ist mindestens ein Lüfter ausgefallen, dann ist Bit 0 (FAN_ERROR_MASK) immer zusätzlich zum Bit des ausgefallenen Lüfters gesetzt.

Die SIMATIC IPCs verfügen über folgende Lüfter:

Gerät	Lüfter
IPC647D	<ul style="list-style-type: none">• Gehäuselüfter vorne links• Gehäuselüfter vorne rechts• Netzteil Lüfter (abhängig von der Gerätevariante)• Gehäuselüfter innen
IPC847D	<ul style="list-style-type: none">• Gehäuselüfter vorne• Netzteil Lüfter (abhängig von der Gerätevariante)• Gehäuselüfter innen (abhängig von der Gerätevariante)
IPC627D, IPC827D	<ul style="list-style-type: none">• Netzteil Lüfter• Gehäuselüfter hinten

Beispiel: IPC647D

Gehäuselüfter vorn links ist defekt:

- Rückgabewert ist 0x9

Gehäuselüfter vorn rechts und Netzteil Lüfter sind defekt:

- Rückgabewert ist 0x25

12.5.4 ReadBatteryStatus

Die Funktion "ReadBatteryStatus()" liest den CMOS-Batteriestatus aus einem Register aus. Mit dieser Funktion kann der Anwender, z. B. von einer Task aus, regelmäßig den CMOS-Batteriestatus auslesen und einem Batterieausfall vorbeugen.

Include

```
<cpuex.h>
```

Syntax

```
int ReadBatteryStatus (void);
```

Rückgabewert

Wert	Bedeutung
0 (OK)	CMOS-Batterie ist in Ordnung
1	CMOS-Batterie ist leer
2	CMOS-Batterie ist erschöpft (Restkapazität reicht noch für höchstens einen Monat)
4	Ein redundantes Netzteil ist ausgefallen
-1	Aufruf wird für diese CPU nicht unterstützt
-2	Batteriestatus konnte nicht gelesen werden

12.5.5 Betriebsstundenzähler

Die Funktion "Betriebsstundenzähler" bietet die Möglichkeit, eine spezielle Task als Betriebsstundenzähler des Systems zu aktivieren. Damit können Sie z. B. Wartungsmaßnahmen am SIMATIC IPC festlegen und die Zeit in Minuten seit Inbetriebnahme des SIMATIC IPC ermitteln. Der Betriebsstundenzähler wird zyklisch je nach eingestelltem Aktualisierungsintervall aktualisiert. Bei Systemhochlauf-Intervallen, die kürzer als das Aktualisierungsintervall sind, können Sie die Aktualisierung des Betriebsstundenzählers mit Aufruf der Funktion "UpdateOpHoursCounter()" anstoßen.

Es ist eine Schwelle für den Betriebsstundenzähler mit der Funktion "SetOpHoursCounter()" einstellbar. Die gesetzte Schwelle wird in der Datei "ophours.vxw" gespeichert und bleibt somit auch nach einem Neustart erhalten. Bei Erreichen der Schwelle wird das Semaphor "/IPC_HW" gesetzt. Anwendungen können dieses Semaphor sowohl auf Kernel-Ebene als auch auf User-Ebene abfragen.

12.5.5.1 SetOpHoursCounter

Die Funktion "SetOpHoursCounter()" setzt den Schwellwert des Betriebsstundenzählers in der Datei "ophours.vxs". Bei Überschreiten des Schwellwerts wird das Semaphor "/IPC_HW" gesetzt.

Include

```
<cpuex.h>
```

Syntax

```
int SetOpHoursCounter  
(  
    unsigned int limitHoursCounter  
);
```

Parameter

Parameter	Bedeutung
limitHoursCounter	Schwellwert in Minuten, bei dessen Überschreiten die Überwachungstask das Semaphor "/IPC_HW" setzt.

Rückgabewert

Wert	Bedeutung
0 (OK)	Funktion wurde erfolgreich ausgeführt.
-1	Der Betriebsstundenzähler wurde noch nicht initialisiert.
-3	Die Datei "ophours.vxw" kann nicht geöffnet werden.
-4	Die Datei "ophours.vxw" kann nicht geschrieben werden.
-6	Die Länge der Datei "ophours.vxw" ist ungültig.

12.5.5.2 GetOpHoursCounter

Die Funktion "GetOpHoursCounter()" liest den Schwellwert und den aktuellen Stand des Betriebsstundenzählers aus der Datei "ophours.vxw".

Include

```
<cpuex.h>
```

Syntax

```
int GetOpHoursCounter
(
    unsigned int *hoursCounter,
    unsigned int *limitHoursCounter
);
```

Parameter

Parameter	Bedeutung
hoursCounter	Zeiger auf den aktuellen Stand des Betriebsstundenzählers in Minuten.
limitHoursCounter	Zeiger auf den Schwellwert in Minuten, bei dessen Überschreiten die Überwachungs-Task das Semaphor "/IPC_HW" setzt.

Hinweis

Wenn ein Zeiger den Wert NULL hat, dann wird der zugehörige Parameter nicht ausgewertet.

Rückgabewert

Wert	Bedeutung
0 (OK)	Funktion wurde erfolgreich ausgeführt.
-1	Der Betriebsstundenzähler wurde noch nicht initialisiert.
-3	Die Datei "ophours.vxw" kann nicht geöffnet werden.
-6	Die Länge der Datei "ophours.vxw" ungültig.

12.5.5.3 UpdateOpHoursCounter

Die Funktion "UpdateOpHoursCounter()" stößt die Aktualisierung des Betriebsstundenzählerstands in der Datei "ophours.vxw" manuell an, d. h. außerhalb des eingestellten automatischen Aktualisierungszyklus. Bei Überschreiten des Schwellwerts setzt die Überwachungstask das Semaphor.

Include

```
<cpuex.h>
```

Syntax

```
int UpdateOpHoursCounter (void);
```

Rückgabewert

Wert	Bedeutung
0 (OK)	Funktion wurde erfolgreich ausgeführt.
-1	Der Betriebsstundenzähler wurde noch nicht initialisiert.
-3	Die Datei "ophours.vxw" kann nicht geöffnet werden.
-4	Die Datei "ophours.vxw" kann nicht geschrieben werden.
-6	Die Länge der Datei "ophours.vxw" ist ungültig.

12.5.6 Watchdog

12.5.6.1 StartWatchdog

Die Funktion "StartWatchdog()" parametrisiert die Überwachungszeit und startet den Watchdog. Der Watchdog dient der Erkennung von Programmabstürzen. Nach der Initialisierung des Watchdogs mit "StartWatchdog()" leuchtet die Watchdog-LED grün.

Solange der Watchdog regelmäßig vor Ablauf der eingestellten Überwachungszeit getriggert wird, leuchtet die Watchdog-LED grün. Wenn der Watchdog nicht vor Ablauf der eingestellten Überwachungszeit getriggert wird, dann leuchtet die Watchdog-LED rot und der Rechner bootet neu.

Include

```
<cpuex.h>
```

Syntax

```
int StartWatchdog
(
    unsigned int msec
);
```

Parameter

Parameter	Bedeutung
msec	<p>0: Watchdog anhalten >0: Watchdog nach msec ms auslösen</p> <p>Für IPC227E, IPC277E und IPC427D sind nur die Werte 94, 210, 340, 460, 590, 710, 840, 960, 2000, 4000, 6000, 8000, 16000, 32000, 48000 und 64000 ms gültig. Für IPC627D/IPC827D/IPC647D/IPC847D sind nur die Werte 4000, 6000, 8000, 16000, 32000, 48000 und 64000 ms gültig.</p>

Rückgabewert

Wert	Bedeutung
0 (OK)	Watchdog wurde gestartet.
1	Watchdog wurde angehalten.
-1	Aufruf wird für diese CPU nicht unterstützt.
-2	Parameterfehler (z. B. ungültiger Wert für "msec").

12.5.6.2 TriggerWatchdog

Die Funktion "TriggerWatchdog()" triggert den Watchdog.

Include

```
<cpuex.h>
```

Syntax

```
int TriggerWatchdog (void);
```

Rückgabewert

Wert	Bedeutung
0 (OK)	Watchdog wurde getriggert.
1	Watchdog kann nicht getriggert werden, da Watchdog nicht gestartet wurde.
-1	Aufruf wird für diese CPU nicht unterstützt.

12.5.7 LED

Bei allen SIMATIC IPCs ist es möglich, von der Anwendung aus mit bestimmten Aufrufen User-LEDs zu setzen.

- IPC227E, IPC427D
Die User-LEDs "L1" (zweifarbige, grün und orange), "L2" und "L3" (zweifarbige, rot und orange) können gesetzt werden.
- IPC627D/IPC827D
Die User-LEDs "L1", "L2" und "L3" (dreifarbige, grün, rot und orange) können gesetzt werden.
- IPC647D/IPC847D
Die User-LEDs "HDD0 ALARM" und "HDD1 ALARM" können auf rot gesetzt werden.

12.5.7.1 LED setzen

Bei allen SIMATIC IPCs ist es möglich, mit dem Aufruf "SetUserLED()" die User-LEDs zu setzen. Die Leuchtfarbe der LED kann bei jedem IPC unterschiedlich sein. Weitere Informationen finden Sie im Beispielprogramm.

Folgender Auszug aus dem Beispielprogramm zeigt exemplarisch die Defines für IPC427D:

```
#define LED_IPC427D_L1_ORANGE    0x0001
#define LED_IPC427D_L2_ORANGE    0x0002
#define LED_IPC427D_L3_ORANGE    0x0004
#define LED_IPC427D_L2_RED       0x0010
#define LED_IPC427D_L3_RED       0x0020
```

Include

```
<cpuex.h>
```

Syntax

```
int SetUserLED
(
    unsigned int led
);
```

Parameter

Parameter	Bedeutung
led	Die Ansteuerung der LEDs erfolgt bitweise (siehe Beispielprogramm).

Rückgabewert

Wert	Bedeutung
0 (OK)	LED wurde gesetzt.

12.5.7.2 LED zurücksetzen

Bei allen SIMATIC IPCs ist es möglich, mit dem Aufruf "ResetUserLED()" die User-LEDs zurückzusetzen.

Include

```
<cpuex.h>
```

Syntax

```
int ResetUserLED  
(  
    unsigned int led  
);
```

Parameter

Parameter	Bedeutung
led	Die Ansteuerung der LEDs erfolgt bitweise (siehe Beispielprogramm).

Rückgabewert

Wert	Bedeutung
0 (OK)	LED wurde zurückgesetzt.

12.5.8 GetSMARTStatus

Beschreibung

Die Funktion "GetSMARTStatus()" liest die S.M.A.R.T.-Statusinformation eines bestimmten Laufwerks aus. Das Laufwerk kann eine Festplatte (nur SATA), ein Solid State Drive oder eine SIMATIC CF-Card sein.

Hinweis

Die Funktion "GetSMARTStatus()" liefert nur dann gültige Werte, wenn die zu prüfende Hardware die S.M.A.R.T.-Funktionalität unterstützt.

Include

```
<cpuex.h>
```

Syntax

```
int GetSMARTStatus
(
    const char *volumeName
);
```

Parameter

Parameter	Bedeutung
volumeName	Laufwerksbezeichnung, z. B. "/ata0"

Rückgabewert

Wert	Bedeutung
0 (OK)	Festplatte / Flashcard ist in Ordnung
1	Eine der vom Hersteller vorgegebenen Schwellen wurde überschritten. Festplatte / Flashcard sollte nicht mehr benutzt werden.
-1	Festplatte / Flashcard unterstützt keine S.M.A.R.T.-Funktionen
-2	Der Massenspeicher-Treiber ist nicht gestartet
-3	Ungültiger Laufwerksname bzw. Nullzeiger für Parameter "volumeName"
-4	Im Massenspeicher-Treiber ist ein Fehler aufgetreten

12.5.9 GetCMOSDateTime

Beschreibung

Die Funktion "GetCMOSDateTime()" liest das Datum und die Uhrzeit der Echtzeituhr (Realtime Clock) aus und speichert das Ergebnis in der "Struktur tm" ab.

Die "Struktur tm" ist in der Headerdatei "time.h" folgendermaßen definiert:

```
struct tm
{
    int tm_sec; /* seconds after the minute - [0, 59] */
    int tm_min; /* minutes after the hour - [0, 59] */
    int tm_hour; /* hours after midnight - [0, 23] */
    int tm_mday; /* day of the month - [1, 31] */
    int tm_mon; /* months since January - [0, 11] */
    int tm_year; /* years since 1900 */
    int tm_wday; /* days since Sunday - [0, 6] */
    int tm_yday; /* days since January 1 - [0, 365] */
    int tm_isdst; /* Daylight Saving Time flag */
};
```

Das Element `tm_isdst` ist ohne Bedeutung.

Include

```
<cpuex.h>
```

Syntax

```
int GetCMOSDateTime
(
    struct tm *pTime
);
```

Parameter

Parameter	Bedeutung
<code>pTime</code>	Zeiger auf Struktur <code>tm</code>

Rückgabewert

Wert	Bedeutung
0 (OK)	Funktion erfolgreich

12.5.10 SetCMOSDateTime

Beschreibung

Die Funktion "SetCMOSDateTime()" setzt das Datum und die Uhrzeit der Echtzeituhr (Realtime Clock) anhand der "Struktur tm". Es wird nicht überprüft, ob es sich dabei um eine sinnvolle Datums- bzw. Uhrzeitangabe handelt.

Die "Struktur tm" ist in der Headerdatei "time.h" folgendermaßen definiert:

```
struct tm
{
    int tm_sec; /* seconds after the minute - [0, 59] */
    int tm_min; /* minutes after the hour - [0, 59] */
    int tm_hour; /* hours after midnight - [0, 23] */
    int tm_mday; /* day of the month - [1, 31] */
    int tm_mon; /* months since January - [0, 11] */
    int tm_year; /* years since 1900 */
    int tm_wday; /* days since Sunday - [0, 6] */
    int tm_yday; /* days since January 1 - [0, 365] */
    int tm_isdst; /* Daylight Saving Time flag */
};
```

Die Elemente `tm_wday`, `tm_yday` und `tm_isdst` sind ohne Bedeutung und werden nicht ausgewertet.

Include

```
<cpuex.h>
```

Syntax

```
int SetCMOSDateTime
(
    struct tm *pTime
);
```

Parameter

Parameter	Bedeutung
pTime	Zeiger auf Struktur tm

Rückgabewert

Wert	Bedeutung
0 (OK)	Funktion erfolgreich

DiagMonitor-Agent für VxWorks

13.1 Übersicht

Die SIMATIC IPC DiagMonitor Software ist ein optionales bestellbares Produkt zur Fernüberwachung der SIMATIC IPCs unter Windows. Voraussetzung für den Einsatz zusammen mit Vxworks ist die SIMATIC IPC DiagMonitor Software ab Version 4.4.x.

SIMATIC Industrie PCs bieten bereits in der Grundausbaustufe die Möglichkeit, Überwachungsfunktionen einzusetzen. Es stehen in Verbindung mit der DiagMonitor Software folgende Anzeige-, Überwachungs- und Kontrollfunktionen zur Verfügung:

- Temperaturüberwachung (Über- oder Untertemperatur, Kabelbruch am Temperatursensor)
- Lüfterüberwachung (Unterdrehzahl, Ausfall eines Lüfters, Kabelbruch der Tachometerleitung)
- Überwachung der Batteriespannung
- Überwachung der Festplatten mit S.M.A.R.T.-Funktionalität
- Watchdog (Hardware- oder Software-Reset des Rechners)
- Betriebsstundenzähler (Informationen zur Gesamtlaufzeit)

Mit der SIMATIC IPC DiagMonitor Software können Sie diese Funktionen zur Überwachung nutzen. SIMATIC IPC DiagMonitor besteht aus zwei Teilen:

- DiagMonitor Management Explorer:
läuft unter Windows
- DiagMonitor-Agent:
liefert unter VxWorks über SNMP die Daten zum Windows System

Der DiagMonitor Management Explorer listet die überwachten Gerätekomponenten und deren Zustand in einer Übersicht auf.

Zur Identifikation der Station wird der VxWorks Target Name verwendet. Dieser wird über die "boot line" mit dem Parameter "tn" (z. B. "tn=IPC427D") gesetzt und muss im Netzwerk eindeutig sein, siehe Kapitel Ethernet-Schnittstelle konfigurieren (Seite 28).

Die Festplatten-Ansicht zeigt Statusinformationen zu allen angeschlossenen Datenträgern (HDD, SSD, CFAST und USB) an. Zur Laufzeit eingesteckte oder entfernte USB-Datenträger werden erkannt und zur Liste der Laufwerke in der Festplatten-Ansicht hinzugefügt oder aus der Ansicht gelöscht.

RAID-Konfigurationen werden von DiagMonitor-Agent für VxWorks nicht unterstützt.

Um die S.M.A.R.T.-Funktion zur Fehlererkennung der Festplatten zu unterstützen, wird der AHCI-Sata-Treiber (INCLUDE_DRV_STORAGE_AHCI) ab VxWorks V6.9.4.3 benötigt. Der Intel AHCI-Sata-Treiber ist nicht mit den SIMATIC IPCs kompatibel.

DiagMonitor-Agent bietet einer Station die Möglichkeit, sich per Watchdog selbst zu überwachen.

Für den Fall eines Ablaufs der Überwachungszeit kann im Management Explorer oder über die DMAPI-Schnittstelle eine der folgenden Aktionen ausgewählt werden:

Reset Typ	Aktion
Reset ein	Hardware-Reset
Reset aus	keine Aktion
Neustarten	Neustart
Herunterfahren	Neustart

Hinweis

Sowohl die Option "Neustart" als auch die Option "Herunterfahren" lösen einen Neustart des Systems aus.

ACHTUNG

Keine korrekte Überwachung des System gewährleistet

Der Watchdog des DiagMonitor-Agent und die Watchdog-Funktionalität der hardwarenahen Funktionen können nicht gleichzeitig benutzt werden. Um eine korrekte Überwachung des Systems zu gewährleisten, wenden Sie nur eine der beiden Funktionalitäten an.

Einige Einstellungen des DiagMonitor-Agent wie z. B. Revisionsalarm oder Alarmzuordnung müssen auch nach dem Ausschalten erhalten bleiben. Diese Information werden in Dateien abgespeichert. Das dabei verwendete Verzeichnis wird durch den Konfigurationsparameter CPUEX_OP_HOURS_COUNTER_PATH von den Hardwarenahen Funktionen festgelegt.

13.2 DMAPI-Schnittstelle

Die DMAPI-Schnittstelle (Diagnose Management API) ist die Programmierschnittstelle zur Verwaltung der Überwachungsdaten der lokalen Station oder mehrerer Stationen im Netzwerk in der Sprache C. Unter VxWorks ist nur die lokale Überwachung des Systems über die DMAPI-Schnittstelle des DiagMonitor möglich. Die externe Überwachung eines Systems über SNMP ist nicht möglich.

Unter Windows können Sie mit der DMAPI-Schnittstelle auch auf ein VxWorks System mit DiagMonitor-Agent zugreifen.

Die Management-Funktionen bestehen aus:

- Abfragen von lesbaren Daten
- Setzen von beschreibbaren Daten
- Hinzufügen oder Entfernen von Elementen

Die Überwachungsdaten einer Station sind in Form von Elementen und Gruppen zusammengefasst. Einzelne Elemente können über den Group Identifier sowie den Index innerhalb einer Gruppe identifiziert werden. Eine Station selbst wird bei der Abfrage oder beim Setzen von beschreibbaren Daten als Element angesehen.

Folgende Dateien werden für die Programmentwicklung benötigt:

Dateiname	Beschreibung
DMAPI.h	Header-Datei der DMAPI-Schnittstelle
DMMCLConstants.h	Allgemeine Konstanten
DMMCLErrors.h	Fehlerkonstanten
DMMCLStatus.h	Status der Überwachungsobjekte
DMMCLIdentifier.h	Konstanten zur Identifikation der Überwachungsobjekte (Group Identifier, Type Identifier)
DMMCLHWDS structs.h	Strukturen der Überwachungsobjekte
DMColors.h	Konstanten für die LED-Farben
DMAPITypeDefs.h	Typdefinitionen für Funktionszeiger

Die mitgelieferten Beispielprogramme beschreiben die Anwendung der DMAPI-Schnittstelle.

Hinweis

Die Aufrufe `DMInitSNMP`, `DMAddStation` und `DMDeleteStation` sind unter VxWorks nicht verwendbar.

13.3 Betriebsstundenzähler des DiagMonitor-Agent

Der Betriebsstundenzähler der Hardwarenahen Funktionen ist identisch mit dem Betriebsstundenzähler des DiagMonitor-Agent und wird remanent im SIMATIC IPC abgelegt.

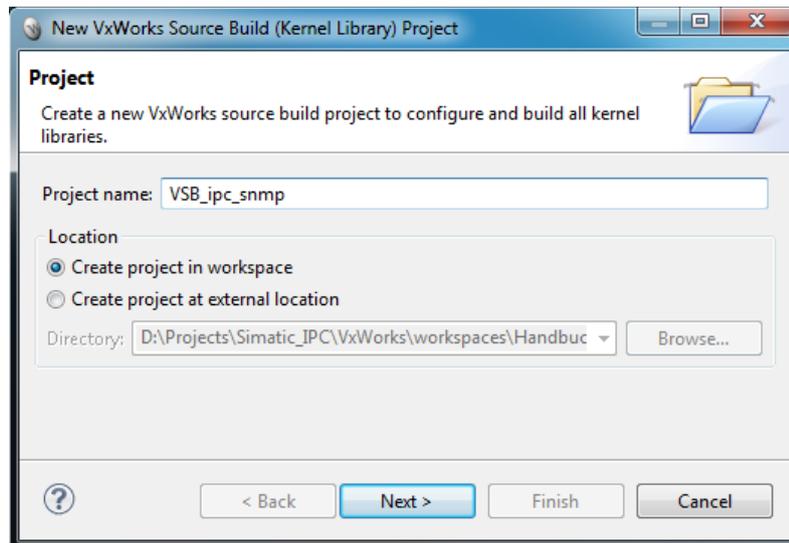
13.4 VxWorks-Image erstellen

DiagMonitor-Agent verwendet zum Bereitstellen der Stationsinformationen das Netzwerkprotokoll SNMP. Die VxWorks-SNMP-Unterstützung wird jedoch nicht als vorkompilierte Bibliothek ausgeliefert, sondern muss aus dem Sourcecode erzeugt werden. Dies geschieht mit einem "VxWorks Source Build (VSB) Project", auf dessen Basis anschließend ein VxWorks-Image-Projekt angelegt werden kann.

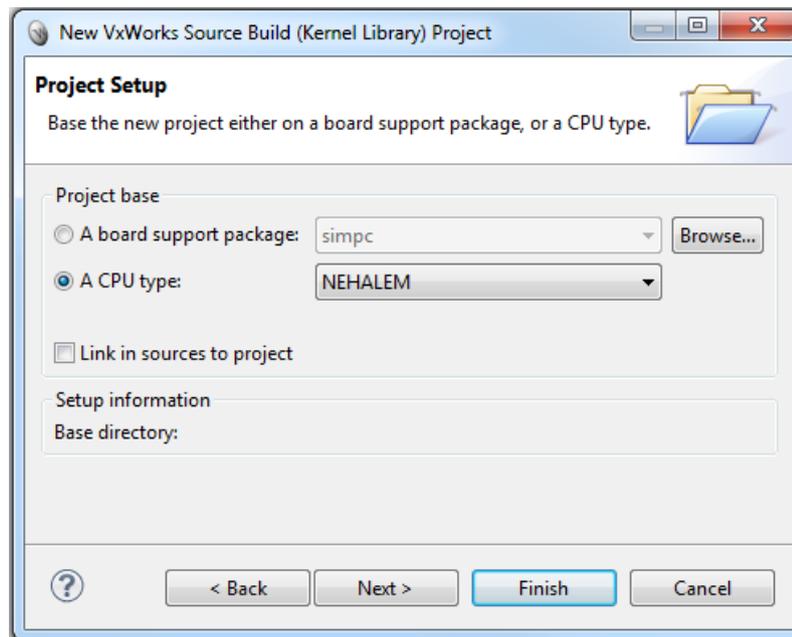
VxWorks Source Build Projekt anlegen

1. Öffnen Sie die Wind River Workbench. Sie finden diese im Windows Startmenü unter "Start > All programs > Wind River > Workbench".
2. Wählen Sie einen Workspace aus. Wählen Sie "File > New > Project".
3. Öffnen Sie im folgenden Dialogfenster im Listenfeld den Ordner "VxWorks 6.x" und wählen Sie dann die Option "VxWorks Source Build (Kernel Library) Project".

4. Geben Sie im folgenden Dialogfenster einen "Project name" (z. B. "VSB_ipc_snmp") ein und wählen Sie die Option "Create project in workspace".



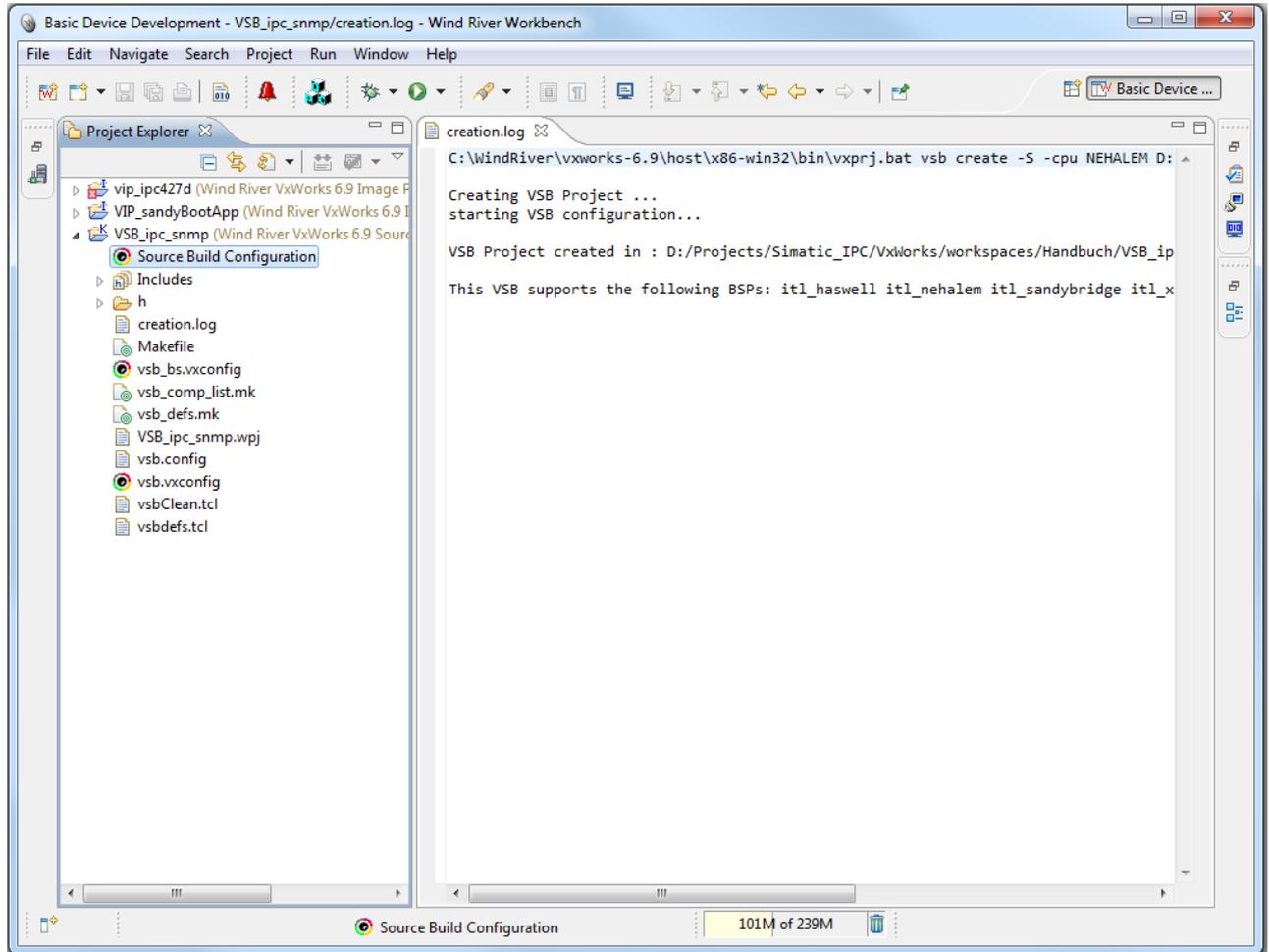
5. Klicken Sie auf "Next >".
6. Klicken Sie im nächsten Fenster auf die Option "A CPU type:" und wählen Sie aus der Liste den Eintrag "NEHALEM".



7. Klicken Sie auf „Finish“.
Ein Projekt ist erstellt.

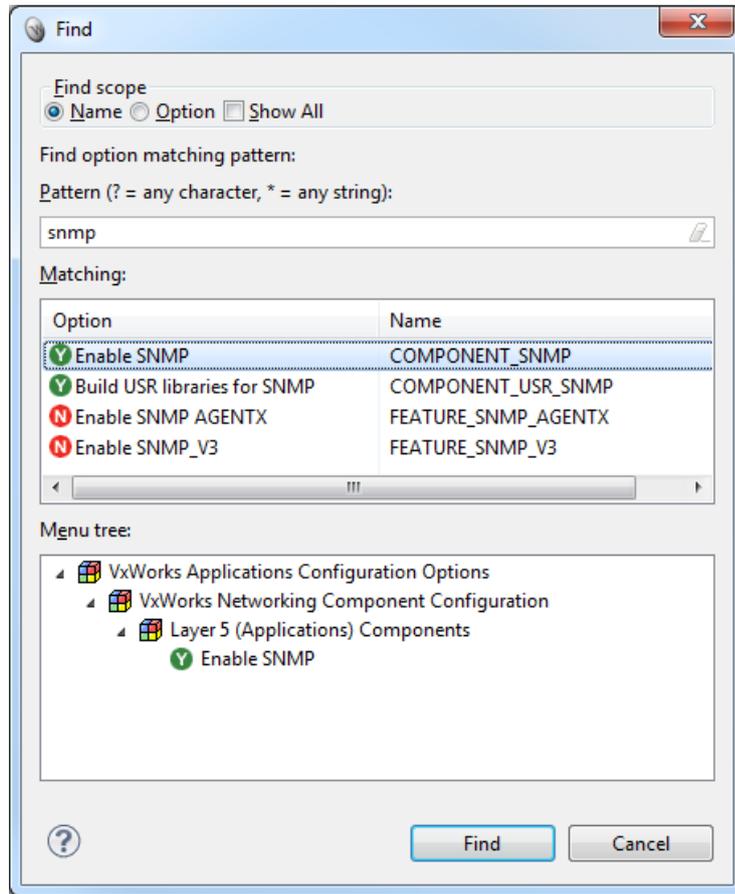
Komponenten einbinden

1. Wählen Sie die Projektansicht.
2. Doppelklicken Sie in der Baumstruktur auf "Source Build Configuration".

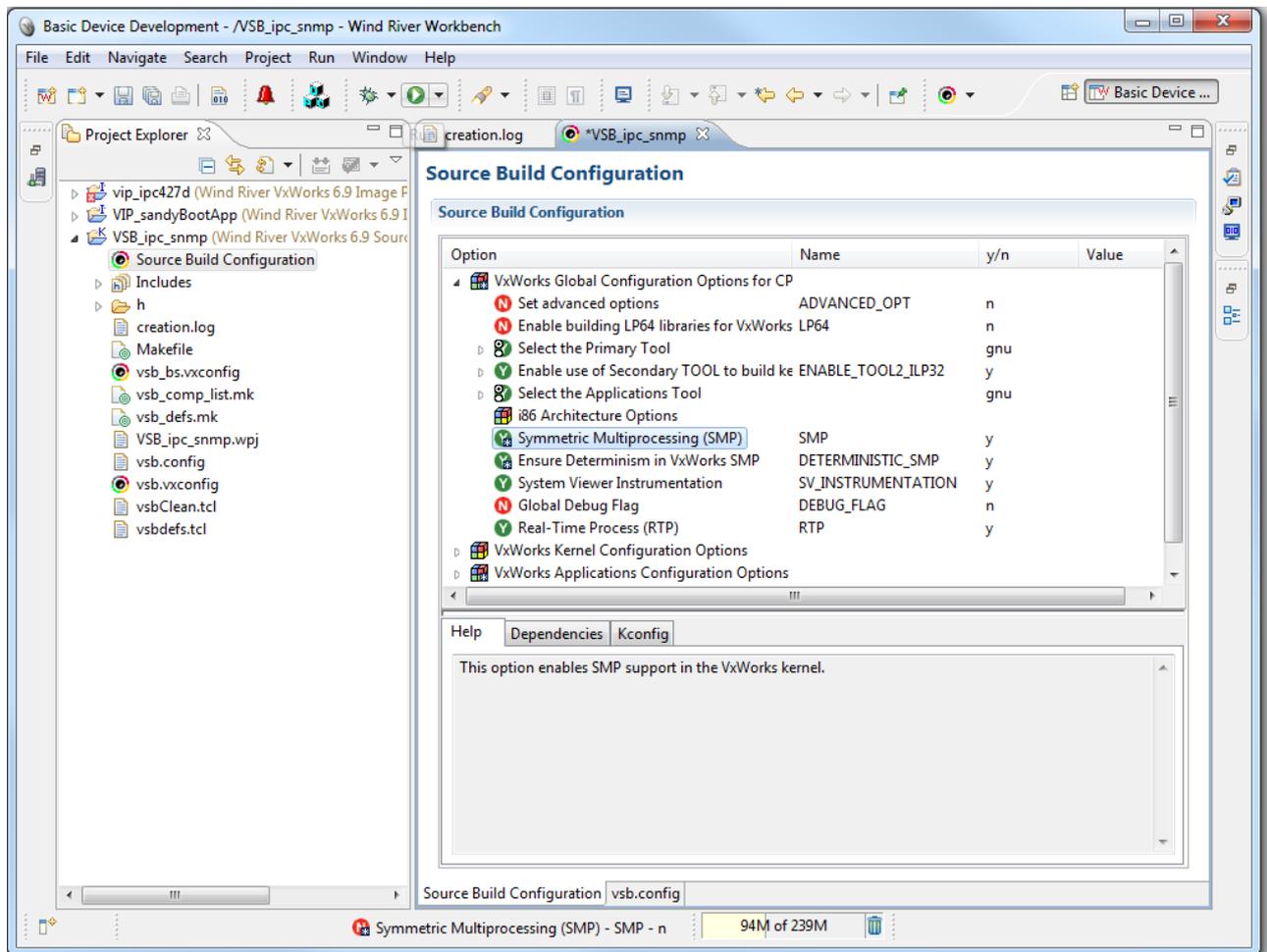


Der Bereich zur "Source Build Configuration" wird geöffnet. In diesem Bereich sind die verschiedenen Komponenten angezeigt.

- Öffnen Sie den Suche-Dialog (STRG + F) und geben Sie als Suchbegriff „snmp“ an. Überprüfen Sie, ob die beiden Komponenten "COMPONENT_SNMP" und "COMPONENT_USR_SNMP" eingebunden sind (grün hinterlegtes Y).



4. Sollten Sie in Ihrem VxWorks-Image Multicore-Unterstützung benötigen, suchen Sie nach der Komponente "SMP" und binden diese mit einem Doppelklick ein.



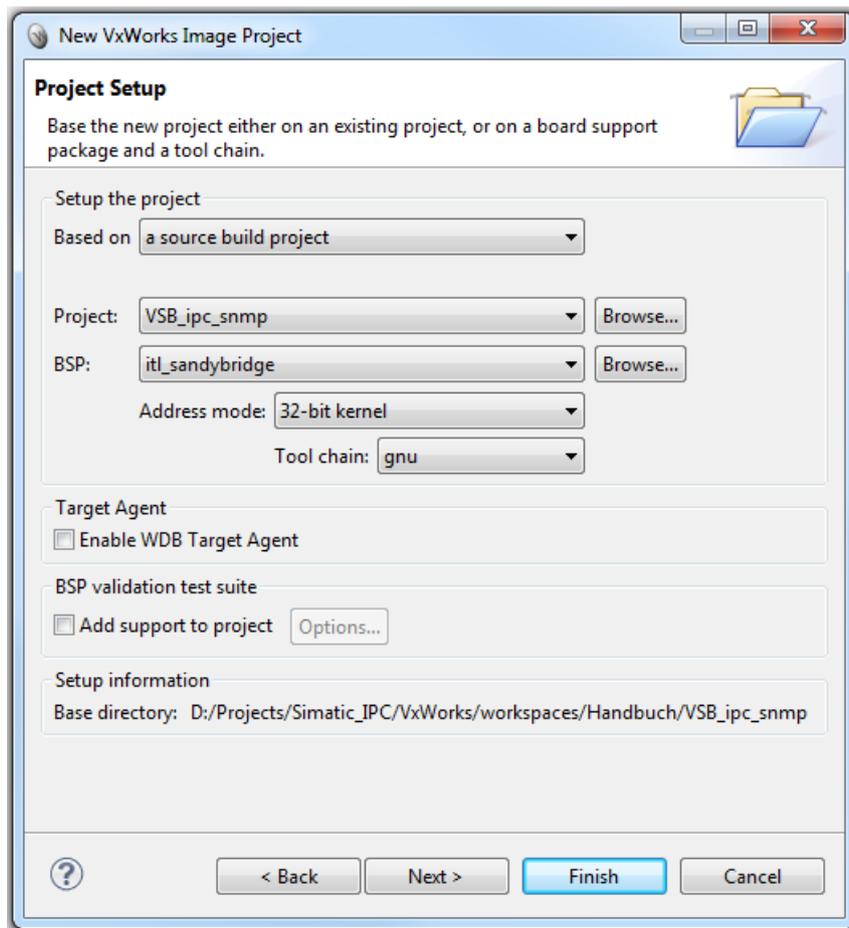
Source Build Projekt erstellen

1. Wählen Sie "Project > Build Project".

Die Erstellung des Projekts ist gestartet.

VxWorks Image Projekt anlegen

1. Erstellen Sie ein neues VxWorks-Image-Projekt, siehe Kapitel VxWorks-Image erzeugen (Seite 15).
2. Wählen Sie folgende Einträge im Dialog "Project Setup", siehe Schritt 8, Kapitel BSPs für die SIMATIC IPCs auswählen (Seite 15):
 - Based on: a source build project
 - Project: ihr Source Build Projekt mit SNMP Unterstützung (hier VSB_ipc_snmp)
 - BSP: das für Ihren IPC passende BSP (hier itl_sandybridge). Informationen hierzu finden Sie in der Tabelle am Anfang von Kapitel: BSPs für die SIMATIC IPCs auswählen (Seite 15).



3. Binden Sie außerdem die Komponente INCLUDE_DIAGMONITOR_AGENT ein und stellen Sie sicher, dass in der "boot line" der Parameter "target name" (tn) gesetzt ist.

13.5 Meldungen

Hochlaufmeldung des Treibers

```
VxWorks SIMATIC IPC Diagmonitor Agent V1.0.x
(C) Copyright 2015, Siemens AG. All rights reserved.
```

Fehlermeldungen des Treibers

In Problemsituationen gibt DiagMonitor-Agent geeignete Warnungen und Fehlermeldungen an der Systemkonsole aus. Diese Ausgaben haben folgendes Format:

```
*** diagmonitor: <text>
```

Hinweis

Wenden Sie sich im Fehlerfall an den Technical Support, siehe Kapitel Service und Support (Seite 129).

13.6 Konfiguration

Die Priorität der von DiagMonitor-Agent erzeugten Tasks (Default: 99) ist konfigurierbar. Erlaubt sind Werte von 10 bis 254. Dieser Parameter wird bei der Konfigurierung der Komponente "INCLUDE_DIAGMONITOR_AGENT" beim Erzeugen eines VxWorks-Images angezeigt und kann editiert werden.

Components

Component Configuration

Description	Name	Type	Value
SD card driver	DRV_SDSTORAGE_CARD		
SGMII interface support for Octeon	INCLUDE_OCTEON_SGMII		
SIMATIC IPC DiagMonitor Agent	INCLUDE_DIAGMONITOR_AGENT		
DiagMonitor task priority	DIAG_PRIORITY	uint	99
SIMATIC IPC hardware dependent functions	INCLUDE_IPC_HW_FUNCTIONS		
SIMATIC NET CP 16XX PROFINET	INCLUDE_PROFINET_CP16XX		
SIMATIC NET CP 5622 PROFIBUS	INCLUDE_PROFIBUS_CP5622		
SIMATIC NET PN PROFINET	INCLUDE_PROFINET_LAN		
SMSC LAN9118 VxBus Enhanced Network Driver	INCLUDE_SMSCLAN9118_VXB_END		

Synopsis

driver that allows remote monitoring of the system with the SIMATIC IPC DiagMonitor diagnostics and signaling software

Defined at:

<

Overview Bundles Components

Die Basis-Priorität wird mit dem Parameter "DIAG_PRIORITY" konfiguriert.

Der Name aller von DiagMonitor-Agent erzeugten Tasks beginnt mit "diag_" (z. B. diag_t1, diag_driver).

In der folgenden Tabelle ist angegeben, welche Tasks DiagMonitor-Agent anlegt und welche Prioritäten diesen Tasks zugeordnet sind.

Task-Name	Beschreibung	Priorität
diag_driver	Task zur Verarbeitung	Basis-Priorität
diag_Timer		Basis-Priorität
diag_tLIMnr	Task zur Verarbeitung von LLMNR-Anfragen	Basis-Priorität
diag_t0	Task zur Statusabfrage des Hardware-Watchdog	1
diag_t1		Basis-Priorität
diag_t2		Basis-Priorität
diag_t3		Basis-Priorität
diag_t4		Basis-Priorität
diag_t5		Basis-Priorität
diag_t18		Basis-Priorität - 1

VxWorks V7 ist das Nachfolgeprodukt zu Vxworks V6.9.

Wind River Workbench 4 ist die Entwicklungsumgebung von Vxworks V7.

Wind River Workbench 3.3 ist die Entwicklungsumgebung von Vxworks V6.9.

In diesem Kapitel sind die von der Beschreibung für VxWorks V6.9 abweichenden Einstellungen und Entwicklungsschritte beschrieben.

Hinweis

Verschiedene Versionen von VxWorks

Die nachfolgenden Kapitel beziehen sich auf die VxWorks Version 7.

Ist ein Thema nicht für VxWorks V7 beschrieben, so sind die vorhergehenden Informationen zu diesem Thema für VxWorks V6.9 unverändert gültig.

Für VxWorks V7 gilt dann zum jeweiligen Thema folgendes:

- Für die Abkürzung "<WIND_BASE>" ist der Pfad "<installDir>\WindRiver\vxworks-7" einzusetzen.
 - Der Ablagepfad von Dateien lautet für die unterschiedlichen Versionen folgendermaßen:
 - VxWorks V6.9: "<WIND_BASE>\target\3rdparty\siemens\"
 - VxWorks V7: "<WIND_BASE>\pkgs\siemens\simatic_ipc_sp-w.x.y.z\"
-

14.1 Installation auf dem Entwicklungsrechner

Vorgehensweise

Um das SIMATIC IPC Support Package für VxWorks V7 zu installieren, gehen Sie wie folgt vor:

1. Legen Sie die Produkt-CD in das Laufwerk Ihres Entwicklungsrechners ein.
2. Starten Sie den Dateimanager (z. B. Explorer).
3. Rufen Sie auf der Produkt-CD im Ordner "IPC_Support_Package_for_VxWorks_7" das Programm "setup.exe" auf und folgen Sie den Anweisungen.

Ergebnis

Die Installation ist abgeschlossen.

Ablage der Dateien im Dateisystem

Nach der Installation sind die Dateien in dem folgenden Verzeichnisbaum abgelegt:

```
(<WIND_BASE>\pkgs\siemens)
|
\--- simatic_ipc_sp-w.x.y.z      SIMATIC IPC Support Package layer
|
|
+---cdf
|   *.cdf                        Component definition files for Siemens components
|
+---docs
|   *.pdf                        Documentation for Siemens components
|
+---examples
|   *.*                          Examples for Siemens components
|
+---h
|   *.h                          Header files for Siemens components
|
+---lib
|   lib*.a                       Library for Siemens components
|
\--- <additional directories and files>
```

Beachten Sie die Readme-Datei mit wichtigen Informationen zum SIMATIC IPC Support Package for VxWorks.

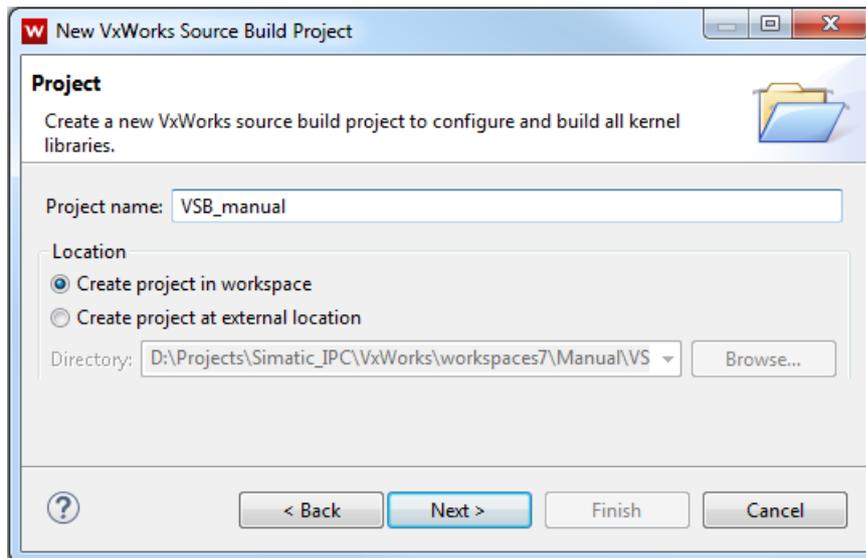
Die Readme-Datei befindet sich im Hauptverzeichnis der CD und im Verzeichnis "`<WIND_BASE>\pkgs\siemens\simatic_ipc_sp-w.x.y.z`".

14.2 Image erzeugen

Source Build Projekt anlegen

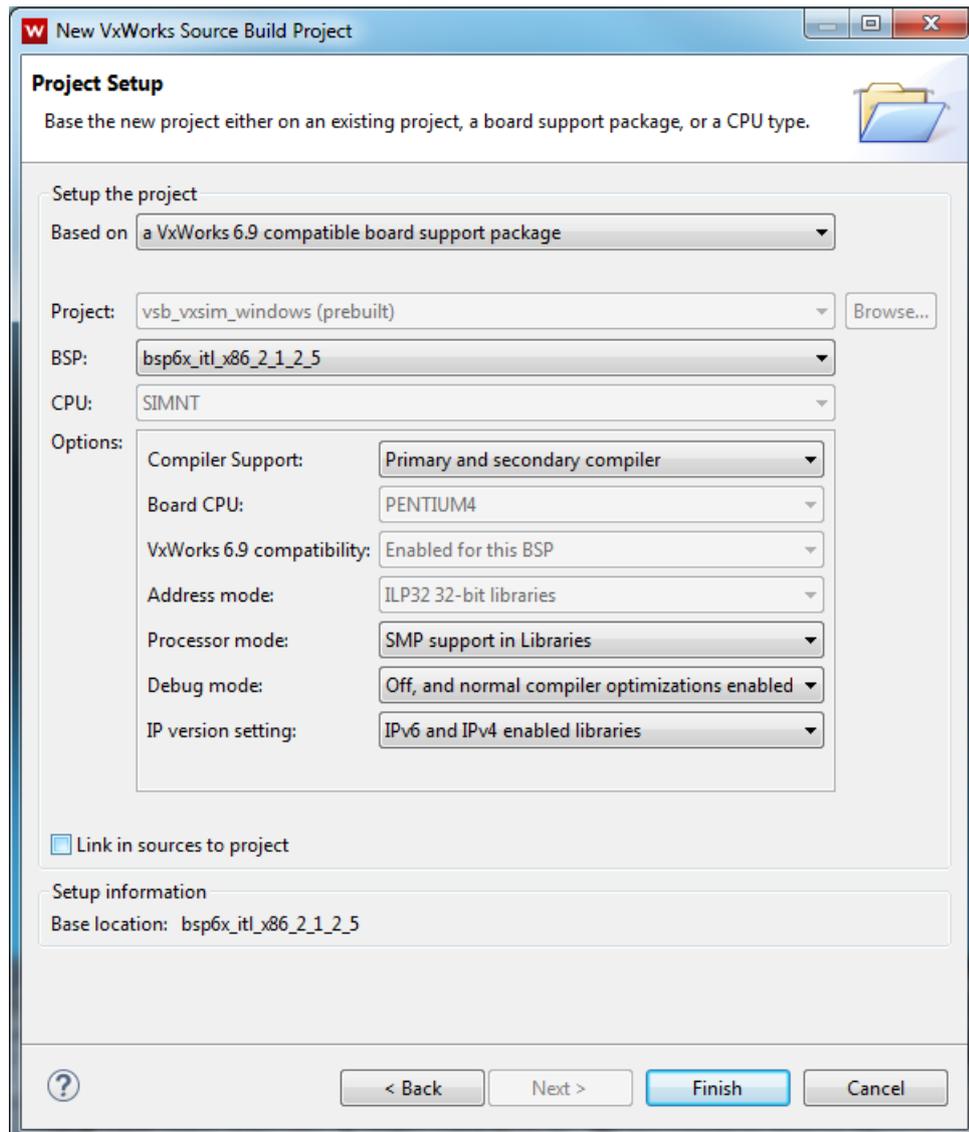
1. Öffnen Sie die Wind River Workbench.
Sie finden diese im Windows Startmenü unter "Start > All programs > Wind River > Workbench".
2. Wählen Sie einen Workspace, in dem Sie Ihr VxWorks-Image Projekt erstellen möchten.
3. Wählen Sie "File > New > Wind River Workbench Project".
4. Wählen Sie im folgenden Dialogfenster die Option "Source Build".
5. Klicken Sie "Next >"

6. Geben Sie im folgenden Dialogfenster einen "Project name" (z. B. "VSB_manual") ein und wählen Sie die Option "Create project in workspace".



7. Klicken Sie "Next >".

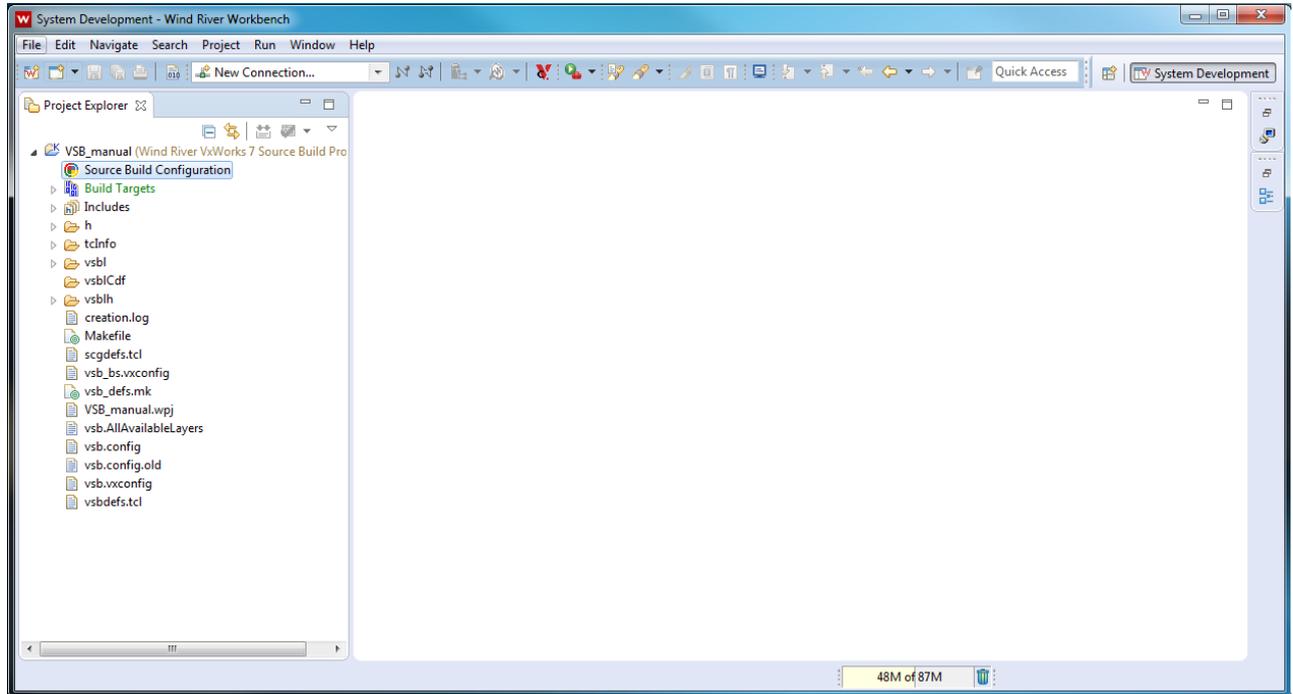
8. Wählen Sie das BSP aus. Wählen Sie aus den entsprechenden Listen folgende Einstellungen:
 - Based on: "a VxWorks 6.9 compatible board support package"
 - BSP: das board support package "bsp6x_itl_x86" in der neuesten Version
 - Options - Processor mode: "SMP support in Libraries".



9. Klicken Sie "Finish".

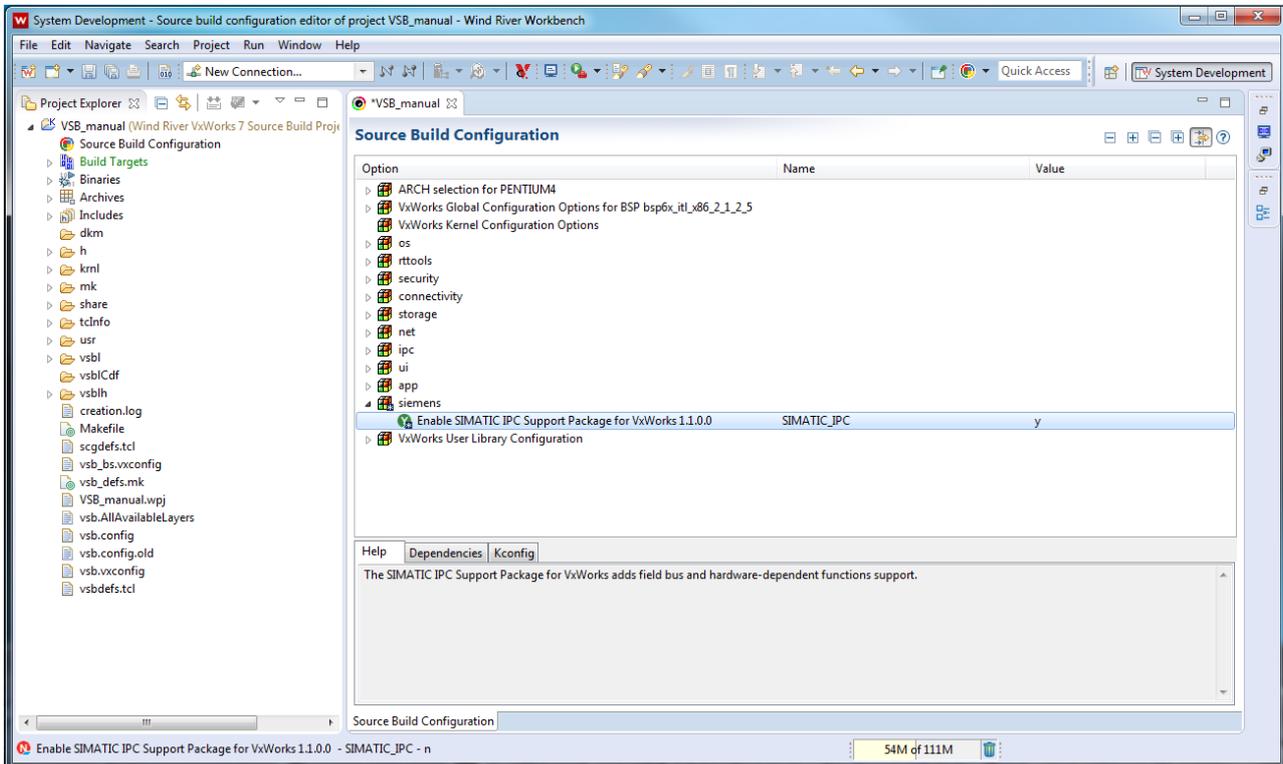
Komponenten auswählen

1. Wählen Sie die Projektansicht.
2. Doppelklicken Sie in der Baumstruktur Ihres Projekts auf "Source Build Configuration".



Der Bereich zur "Source Build Configuration" wird geöffnet. In diesem Bereich werden die verschiedenen Komponenten angezeigt.

- Öffnen Sie in der Ansicht "Source Build Configuration" den Ordner "siemens" und doppelklicken Sie auf den Layer "SIMATIC_IPC".



Hinweis

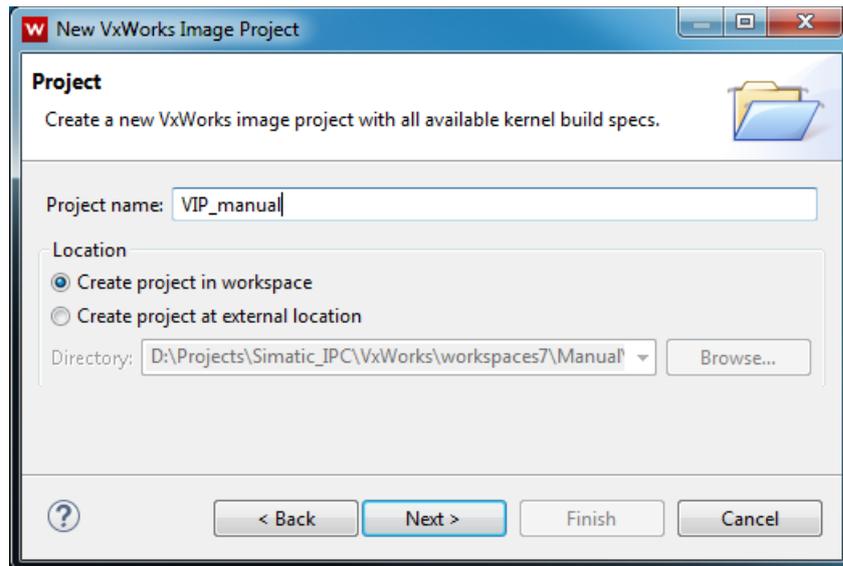
Die im Screenshot gezeigte Version der Komponente ist ein Beispiel. Sollten Sie mehrere Versionen des SIMATIC IPC Support Package for VxWorks installiert haben, werden diese vollständig gezeigt.

Source Build Projekt erstellen

- Wählen Sie "Project > Build Project".
Das Projekt wird erstellt.

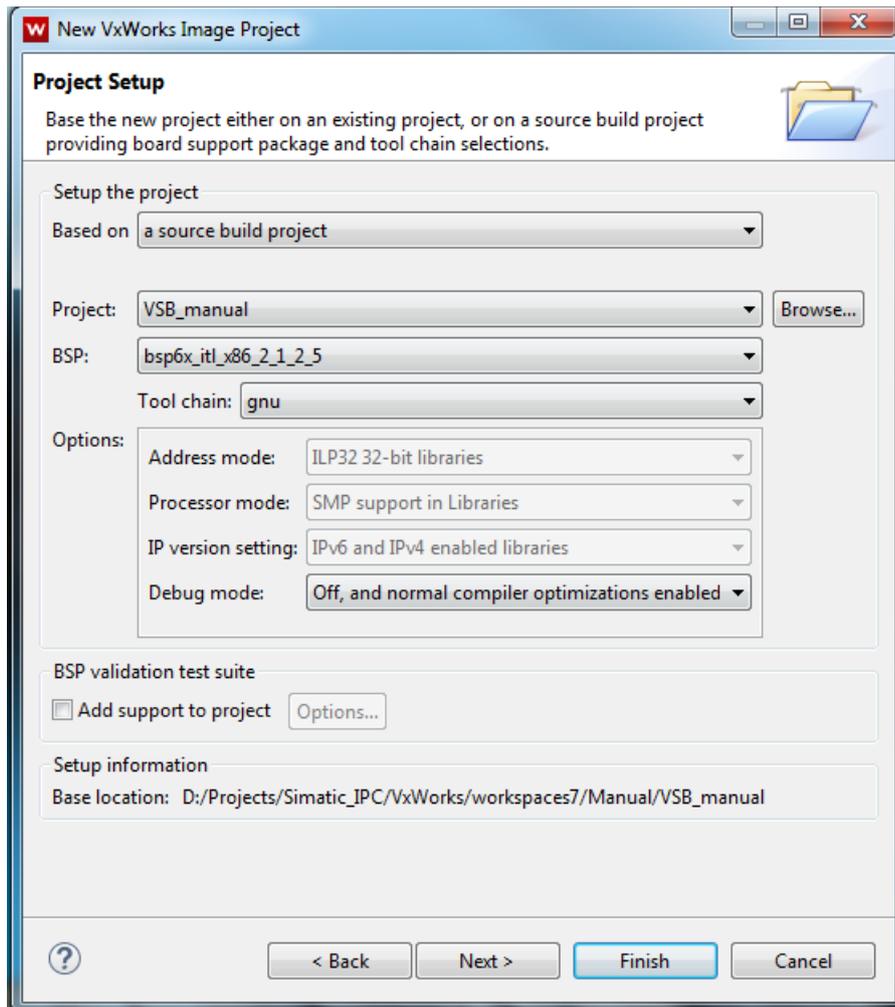
VxWorks Image Projekt anlegen

1. Wählen Sie "File > New > Wind River Workbench Project".
2. Wählen Sie im folgenden Dialogfenster die Option "Kernel Image".
3. Geben Sie im folgenden Dialogfenster einen "Project name" (z. B. "VIP_manual") ein und wählen Sie die Option "Create project in workspace".



4. Klicken Sie "Next >".

5. Wählen Sie unter "Project" das gerade erstellte Source Build Projekt aus.



6. Klicken Sie "Finish".

VxWorks Image konfigurieren und erstellen

Konfigurieren Sie das Image Projekt, siehe Kapitel VxWorks-Image erzeugen (Seite 15).

Hinweis

Möglicherweise sind einige der Komponenten von VxWorks V6.9 in VxWorks V7 nicht mehr vorhanden.

Verbindung zum Debuggen aktivieren

Sie können die Debug-Verbindung zu Workbench aktivieren.

Wählen Sie die folgenden Komponenten:

- VxWorks Stop Mode Debug Agent (INCLUDE_STOP_MODE_AGENT)
- Debug Agent Start (INCLUDE_DEBUG_AGENT_START)

14.3 Bootloader für VxWorks

Für VxWorks V7 mit VxWorks 6.9 kompatiblen BSPs wird der Systemstart mit VxWorks Bootloader nicht unterstützt. Verwenden Sie stattdessen GRUB.

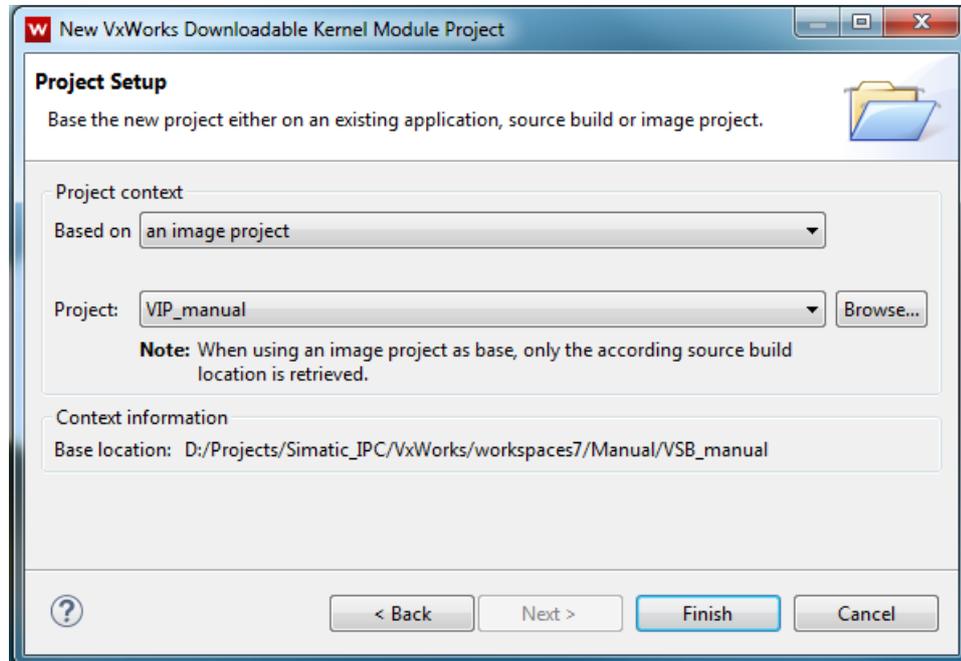
14.4 Nachladbares Kernel Modul erzeugen

Wie Sie ein nachladbares Kernel Module erzeugen, ist grundsätzlich in folgendem Kapitel beschrieben: "Nachladbares Kernel Modul erzeugen (Seite 55).

Nachfolgend werden die Änderungen zu diesem Thema für VxWorks V7 beschrieben.

In VxWorks V7 basieren Kernel Module auf einem VxWorks Image Projekt oder einem VxWorks Source Build Projekt.

1. Wählen Sie im Dialog "Project Setup" ein Image Projekt oder Source Build Projekt aus. Wählen Sie aus den Listen folgende Einstellungen:
 - Based on: "a source build project", wenn das Kernel Modul auf einem Source Build Projekt basieren soll, oder wählen Sie "an image project", wenn das Kernel Modul auf einem Image Projekt basieren soll.
 - Project: das entsprechende Projekt.



Unter VxWorks V7 müssen Sie in den Projekteinstellungen keinen Suchpfad zu den im SIMATIC IPC Support Package for VxWorks mitgelieferten Headerdateien hinzufügen. Lassen Sie Schritt 13 aus folgendem Kapitel aus: Nachladbares Kernel Modul erzeugen (Seite 55).

14.5 Real Time Process Applikation (RTP) erzeugen

Eine Real Time Process Applikation erstellen Sie auf die gleiche Weise wie ein nachladbares Kernel Modul.

Wählen Sie als Projekttyp die Option "VxWorks Real Time Process Project".

Der Name der erzeugten Datei erhält die Erweiterung ".vxe".

14.6 PROFINET-Aufrufe im Real Time Process (User Mode) nutzen

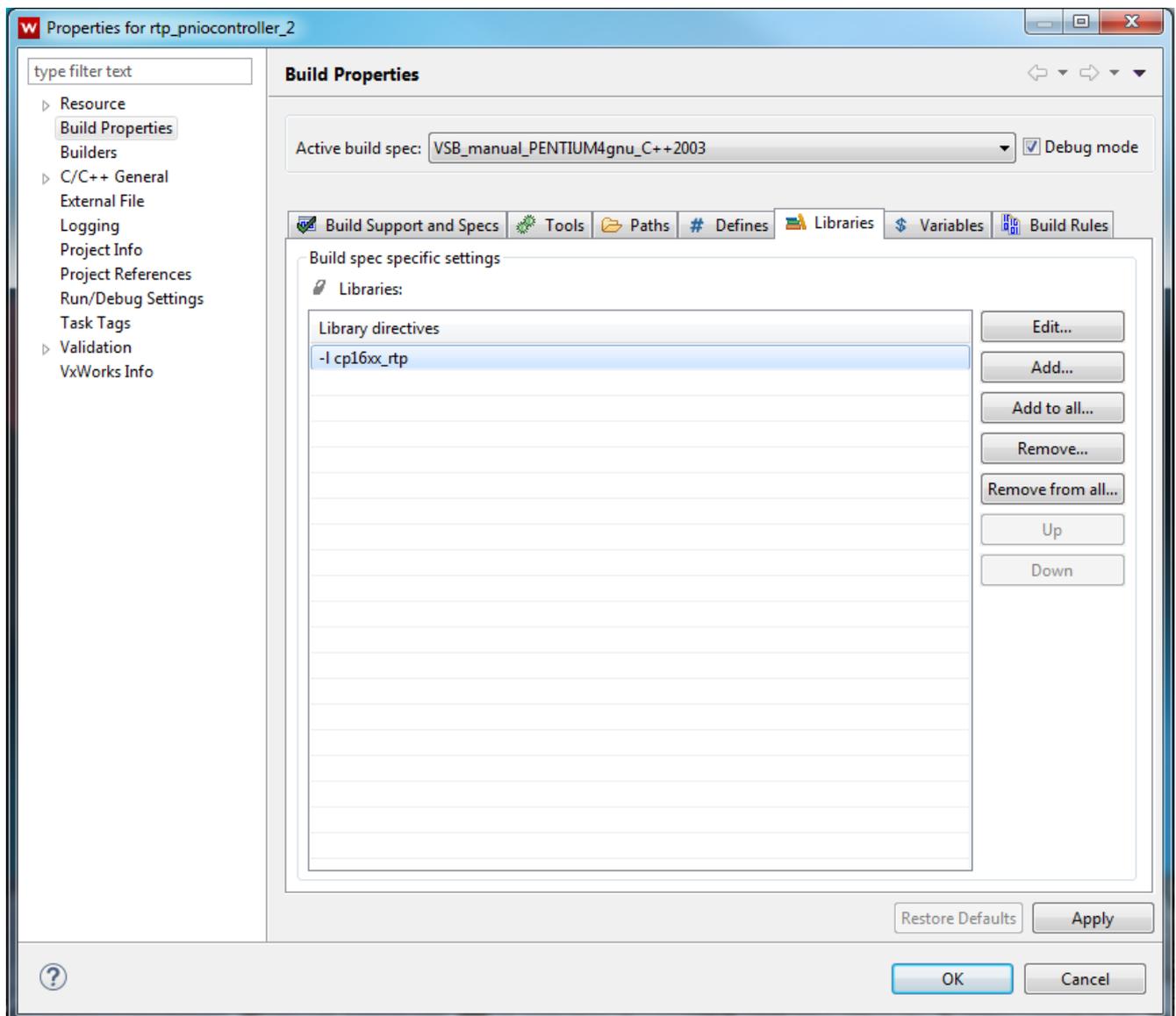
Sie können alle PROFINET-Aufrufe in einem Real Time Process (RTP) nutzen.

Verwenden Sie dazu die mitgelieferte Headerdatei "vxwpio.h".

Binden Sie, je nach Treiber, eine der mitgelieferten Bibliotheken ein:

- CP 16xx-Treiber:
"libcp16xx_rtp.a"
- PN-Treiber:
"libpn_rtp.a"

Im RTP-Mode wird fast der gesamte Treibercode auf User-Ebene (Privilege Level 3) ausgeführt.



14.6 PROFINET-Aufrufe im Real Time Process (User Mode) nutzen

Für die Kommunikation zwischen User Mode und Kernel Mode wird die Syscall-Gruppe 51 verwendet. Informationen zur Konfiguration dieses Werts finden Sie in Kapitel PROFINET-Treiber konfigurieren (Seite 72).

Service und Support

Weiterführende Informationen und Unterstützung zu den beschriebenen Produkten finden Sie im Internet unter folgenden Adressen:

- Technical Support (http://www.siemens.de/automation/csi_de_WW)
- Formular für einen Support-Request (<http://www.siemens.de/automation/support-request>)
- After Sales Information System SIMATIC PC/PG (<http://www.siemens.de/asis>)
- Gesamtdokumentation SIMATIC (<http://www.siemens.de/simatic-tech-doku-portal>)
- Ihr Ansprechpartner vor Ort
(<http://www.automation.siemens.com/mcms/aspa-db/de/Seiten/default.aspx>)
- Trainingscenter (<http://sitrain.automation.siemens.com/sitrainworld/?AppLang=de>)
- Industry Mall (<https://mall.industry.siemens.com>)

Wenn Sie sich mit Ihrem Ansprechpartner vor Ort oder mit dem Technical Support in Verbindung setzen, halten Sie bitte folgende Informationen bereit:

- Bestellnummer des Geräts (MLFB)
- BIOS-Version (Industrie-PC) bzw. Image-Version (Bediengerät)
- Installierte Zusatzhardware
- Installierte Zusatzsoftware

Tools & Downloads

Überprüfen Sie regelmäßig, ob Updates und Hotfixes für Ihr Gerät zum Download bereitstehen. Die Downloads finden Sie im Internet unter "After Sales Information System SIMATIC PC/PG" (siehe oben).

Index

B

Batterieüberwachung, 7
Baudrate, 29, 39
Beispielprogramm, 101
Betriebsstundenzähler, 7, 87, 89, 95, 96, 97, 98
Bootloader, 43

C

CMD Shell, 65
CPUEX_OP_HOURS_COUNTER_CYCLE, 88
CPUEX_OP_HOURS_COUNTER_PATH, 88
CPUEX_OP_HOURS_COUNTER_PRIORITY, 88
CPUEX_RAMDISK_MODE, 88
CPUEX_RAMDISK_SIZE, 88
CPUEX_SYSCALLGROUP, 88
CPU-Grundeinstellung, 7

D

debuggen, 29, 62
DiagMonitor, 107
DiagMonitor-Agent, 107
DKM, 67, 69, 75, 85
DMAPI, 108
Downloadable Kernel Module, 55, 59, 67
DP_CP5622_SYSCALLGROUP, 79
DP-Base-Schnittstelle, 7, 75

E

Ethernet-Schnittstelle, 28, 67, 74

G

GetCMOSDateTime(), 89
GetCPUTypeDMI(), 89, 90
GetOpHoursCounter(), 89, 97
GetSMARTStatus(), 89
GRUB, 46, 48, 50

H

hardwarenahe Funktion, 7
Hardwarenahe Funktion, 11
Hardwarenahe Funktionen, 85

I

Installation, 13
IO-Base-Schnittstelle, 7, 69

K

Kernel Mode, 67, 70, 73, 74, 75, 78, 79, 85, 88
Kernel Modul, 55, 57, 59, 77, 85
Konfiguration, 7, 68, 79, 82, 87
konfigurieren, 29, 43, 70, 78, 85

L

LED
 setzen, 101
 zurücksetzen, 102
Lüfterüberwachung, 7

M

Meldung, 27, 71, 78, 86

P

PNIO_CP16XX_PRIORITY, 72
PNIO_CP16XX_SYSCALLGROUP, 73, 74
PROFIBUS, 7, 11, 12, 75, 76, 82, 83
 konfigurieren, 79
 projektieren, 80, 82, 83
PROFINET, 7, 11, 12, 18, 67, 71, 72, 73, 74
 Beispiel, 22
 CP 16xx-Treiber, 67
 IRT, 67
 PN-Treiber, 67
 PN-Treiber projektieren, 69
 PROFenergy, 69
 RT, 67
Projektierungssoftware, 12

R

RAMDISK, 85, 87, 88
ReadBatteryStatus(), 89, 95
ReadFanStatus(), 89, 93
ReadTemperature(), 89, 91
Real Time Process, 59, 61, 67, 70, 75, 78, 85
Real Time Process Project, 29
Remanenzspeicher, 7, 85
ResetUserLED(), 89, 102
RTP, 29, 59, 61, 67, 70, 78

S

SetCMOSDateTime(), 89
SetOpHoursCounter(), 89, 96
SetUserLED(), 89, 101
SNMP, 108, 109
Source Build Project, 114, 126
StartWatchdog(), 89, 99
Syscall-Gruppe, 70, 72, 73, 74, 78, 79, 85, 87

T

TCP/IP Debug, 29
Temperaturüberwachung, 7
TriggerWatchdog(), 89, 100

U

UpdateOpHoursCounter(), 89, 95, 98
User Mode, 67, 70, 73, 74, 75, 78, 79, 85, 88
User-LED, 7, 85, (LED), (LED)

V

VxWorks Source Build Project, 114, 126
VxWorks V6.9, 12, 117
VxWorks V7, 12, 117
VxWorks-Image, 15, 17, 27, 32, 43, 87

W

Watchdog, 7, 72, 99, 100
Workbench, 11, 12, 15, 16, 17, 29, 32, 55, 57
Workspace, 15, 32