

# **phyCORE-ST10F168/ phyCORE-ST10F269**

## **Hardware-Manual**

**Ausgabe Juli 2002**

Im Buch verwendete Bezeichnungen für Erzeugnisse, die zugleich ein eingetragenes Warenzeichen darstellen, wurden nicht besonders gekennzeichnet. Das Fehlen der © Markierung ist demzufolge nicht gleichbedeutend mit der Tatsache, daß die Bezeichnung als freier Warenname gilt. Ebenso wenig kann anhand der verwendeten Bezeichnung auf eventuell vorliegende Patente oder einen Gebrauchsmusterschutz geschlossen werden.

Die Informationen in diesem Handbuch wurden sorgfältig überprüft und können als zutreffend angenommen werden. Dennoch sei ausdrücklich darauf verwiesen, daß die Firma PHYTEC Meßtechnik GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf den Gebrauch oder den Inhalt dieses Handbuches zurückzuführen sind. Die in diesem Handbuch enthaltenen Angaben können ohne vorherige Ankündigung geändert werden. Die Firma PHYTEC Meßtechnik GmbH geht damit keinerlei Verpflichtungen ein.

Ferner sei ausdrücklich darauf verwiesen, daß PHYTEC Meßtechnik GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf falschen Gebrauch oder falschen Einsatz der Hard- bzw. Software zurückzuführen sind. Ebenso können ohne vorherige Ankündigung Layout oder Design der Hardware geändert werden. PHYTEC Meßtechnik GmbH geht damit keinerlei Verpflichtungen ein.

© Copyright 2002 PHYTEC Meßtechnik GmbH, D-55129 Mainz.

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form ohne schriftliche Genehmigung der Firma PHYTEC Meßtechnik GmbH unter Einsatz entsprechender Systeme reproduziert, verarbeitet, vervielfältigt oder verbreitet werden.

Informieren Sie sich:

	EUROPA	NORD AMERIKA
Adresse:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 D-55129 Mainz GERMANY	PHYTEC America LLC 203 Parfitt Way SW, Suite G100 Bainbridge Island, WA 98110 USA
Angebots Hotline:	+49 (800) 0749832 <a href="mailto:order@phytec.de">order@phytec.de</a>	+1 (800) 278-9913 <a href="mailto:info@phytec.com">info@phytec.com</a>
Technische Hotline:	+49 (6131) 9221-31 <a href="mailto:support@phytec.de">support@phytec.de</a>	+1 (800) 278-9913 <a href="mailto:support@phytec.com">support@phytec.com</a>
Fax:	+49 (6131) 9221-33	+1 (206) 780-9135
Web Seite:	<a href="http://www.phytec.de">http://www.phytec.de</a>	<a href="http://www.phytec.com">http://www.phytec.com</a>

3. Auflage Juli 2002

---

<b>Einleitung .....</b>	<b>1</b>
<b>1 Kurzübersicht über das phyCORE-ST10F168/ST10F269 .....</b>	<b>3</b>
1.1 Blockschaltbild .....	6
1.2 Ansicht des phyCORE-ST10F168/ST10F269 .....	6
<b>2 Anschlußbelegung.....</b>	<b>7</b>
<b>3 Jumper .....</b>	<b>15</b>
3.1 J1 Verwendung von Pin 30 des SRAM .....	19
3.2 J2 Interner oder externer Programmspeicher.....	20
3.3 J3 Flash-Adressierung .....	20
3.4 J4, J5 A/D-Wandler .....	21
3.5 J6 On-Chip Flash Programmierspannung / RDP .....	22
3.6 J7, J8 Nutzung des externen UART .....	23
3.7 J9, J10 Konfiguration von P3.3 und P3.4 für I <sup>2</sup> C-Bus.....	24
3.8 J11 Interrupt-Ausgang der RTC .....	24
3.9 J12 Schreibschutz des E <sup>2</sup> PROM /FRAM.....	25
3.10 J13, J14 Konfiguration der zweiten seriellen Schnittstelle.....	25
3.11 J15 Adresse des seriellen E <sup>2</sup> PROM /FRAM.....	26
3.12 J16, J17, J18, J19 CAN-Schnittstellen.....	26
3.13 J20 Remote Download Quelle .....	28
3.14 J21, J22 Serielle Schnittstelle .....	28
3.15 J23, J24 Versorgungsspannung des Microcontrollers .....	29
<b>4 System-Konfiguration .....</b>	<b>31</b>
4.1 System-Startup-Konfiguration.....	31
<b>5 Speichermodell.....</b>	<b>35</b>
5.1 Bus-Timing .....	38
<b>6 Serielle Schnittstellen .....</b>	<b>39</b>
6.1 RS-232 Schnittstelle .....	39
6.2 CAN-Schnittstelle.....	40
<b>7 Echtzeituhr RTC-8563 (U10).....</b>	<b>41</b>
<b>8 Serielles EEPROM/FRAM (U9).....</b>	<b>43</b>
<b>9 Remote Supervisor Chip (U8) .....</b>	<b>45</b>
<b>10 On-Board Flash Speicher (U1) .....</b>	<b>46</b>
<b>11 Batteriepufferung .....</b>	<b>47</b>
<b>12 Technische Daten.....</b>	<b>49</b>

---

<b>13</b>	<b>Hinweise zum Umgang mit dem phyCORE-ST10F168/ST10F269.....</b>	<b>51</b>
<b>14</b>	<b>Das phyCORE-ST10F168/ST10F269 auf dem Development Board phyCORE-HD200 .....</b>	<b>53</b>
14.1	Das Konzept des Development Board phyCORE-HD200 .....	53
14.2	Anschlüsse und Jumper des Development Board phyCORE-HD200 .....	55
14.2.1	Anschlüsse.....	55
14.2.2	Jumper des Development Boardes phyCORE-HD200 .....	57
14.2.3	Nicht unterstützte Features und unzulässige Jumper-Stellungen.....	59
14.3	Funktionsgruppen des Development Board .....	60
14.3.1	Spannungsversorgung an X1.....	60
14.3.2	Aktivierung des Bootstrap Loaders.....	62
14.3.3	Erste serielle Schnittstelle an Buchse P1A.....	64
14.3.4	Zweite serielle Schnittstelle an Buchse P1B .....	66
14.3.5	Erste CAN-Schnittstelle an Stecker P2A .....	71
14.3.6	Zweite CAN-Schnittstelle an Stecker P2B.....	76
14.3.7	Visualisierungs LED D3 .....	81
14.3.8	Die Belegungen des phyCORE, des Expansion-Bus und des Patchfeldes im Überblick.....	81
14.3.9	Der Batterieanschluß BAT1 .....	89
14.3.10	Auslösung des Interrupts /NMI .....	89
14.3.11	Nummernchip.....	89
14.3.12	Stiftleiste X4.....	90
<b>15</b>	<b>debugCORE-ST10F168/ST10F269.....</b>	<b>91</b>
15.1	Zusatzfunktionen des debugCORE .....	91
15.2	debugADAPTER-167 .....	92
15.2.1	Quad-Connector auf dem debugADAPTER-167.....	93
15.3	Mechanische Abmessungen debugCORE/debugADAPTER.....	95
<b>Index</b>	<b>.....</b>	<b>97</b>

## **Bildverzeichnis**

Bild 1:	Blockschaltbild phyCORE-ST10F168/ST10F269 .....	6
Bild 2:	Ansicht des phyCORE-ST10F168/ST10F269.....	6
Bild 3:	Pinout des phyCORE-Connectors (Drauf-/Durchsicht von oben) .....	9
Bild 4:	Zählweise der Jumper .....	15
Bild 5:	Lage der Jumper (Ansicht Microcontrollerseite / Connectorseite).....	15
Bild 6:	Beispiele für Speichermodelle.....	38
Bild 7:	Mechanische Abmaße.....	49
Bild 8:	Das Mehrplatinenkonzept mit phyCORE-ST10F168/ST10F269, Development Board und Erweiterungsplatine .....	54
Bild 9:	Lage der Anschlüsse auf dem Development Board phyCORE-HD200.....	55
Bild 10:	Zählweise der Jumper .....	57
Bild 11:	Lage der Jumper (Ansicht Bestückungsseite).....	57
Bild 12:	Default Jumper-Stellungen des Development Board HD200 für Standard phyCORE-ST10F168/ST10F269 (nicht maßstabsgetreu).....	58
Bild 13:	Anschluß der Versorgungsspannung an X1 .....	61
Bild 14:	Belegung der DB-9 Buchse P1A als erste RS-232 (Ansicht Vorderseite).....	64
Bild 15:	Belegung der DB-9 Buchse P1B als 2. RS-232 (UART bestückt, Ansicht Vorderseite) .....	68
Bild 16:	Pinbelegung der DB9-Buchsen P1B bei emulierter RS-232 Schnittstelle (Ansicht Vorderseite).....	69
Bild 17:	Belegung des DB9-Steckers P2A (CAN-Treiber auf phyCORE-ST10F168/ST10F269) .....	71
Bild 18:	Belegung des DB-9 Steckers P2A (CAN Treiber auf Development Board).....	72
Bild 19:	Belegung des DB-9 Steckers P2A (CAN-Treiber auf Development Board und galvanische Trennung) .....	75

Bild 20:	Belegung des DB-9 Steckers P2B (CAN-Treiber auf phyCORE-ST10F269, nur mit ST10F269) .....	76
Bild 21:	Belegung des DB-9 Steckers P2B (CAN Treiber auf Development Board, nur mit ST10F269) .....	77
Bild 22:	Belegung des DB-9 Steckers P2B (CAN-Treiber auf Development Board und galvanische Trennung, nur mit ST10F269) .....	80
Bild 23:	Zählweise für den Expansion-Bus .....	82
Bild 24:	Zählweise des Patchfeldes .....	82
Bild 25:	Anschluß des Nummernchips .....	90
Bild 26:	Pinbelegung Nummernchip .....	90
Bild 27:	Lage der Anschlüsse auf dem debugCORE-ST10F168/ST10F269 .....	91
Bild 28:	Mechanische Abmessungen debugCORE-ST10F168/ST10F269 .....	95

## **Tabellenverzeichnis**

Tabelle 1:	Pinout des phyCORE-Connectors X1 .....	13
Tabelle 2:	Jumperbelegung .....	18
Tabelle 3:	J1 SRAM Pin 30 Beschaltung .....	19
Tabelle 4:	J2 Auswahl interner oder externer Programmspeicher .....	20
Tabelle 5:	J3 Flash-Adressierung .....	21
Tabelle 6:	J4, J5 Referenzspannung des A/D-Wandlers .....	21
Tabelle 7:	J6 Aktivierung Oszillator Watchdog .....	22
Tabelle 8:	J7, J8 Nutzung optionale externe UART .....	23
Tabelle 9:	J9, J10 Konfiguration P3.3 und P3.4 / I <sup>2</sup> C-Bus .....	24
Tabelle 10:	J11 Konfiguration P2.9 / RTC Interrupt .....	24
Tabelle 11:	J12 Konfiguration E <sup>2</sup> PROM/FRAM Schreibschutz .....	25
Tabelle 12:	J13, J14 Konfiguration zweite serielle Schnittstelle .....	25
Tabelle 13:	J15 Konfiguration Adresse seriell E <sup>2</sup> PROM/FRAM .....	26
Tabelle 14:	J16, J17, J18 und J19 Konfiguration CAN-Schnittstellen .....	27

---

Tabelle 15: J20 Konfiguration Remote Download Quelle .....	28
Tabelle 16: J21, J22 Konfiguration erste serielle Schnittstelle .....	29
Tabelle 17: J23, J24 Konfiguration VCC Microcontroller .....	29
Tabelle 18: Belegung des Port 0 für die System-Startup-Konfiguration.....	32
Tabelle 19: System-Startup-Konfiguration.....	34
Tabelle 20: Bestückungsoptionen für U9 .....	43
Tabelle 21: E <sup>2</sup> PROM/FRAM Schreibschutz .....	43
Tabelle 22: E <sup>2</sup> PROM/FRAM Adresse.....	44
Tabelle 23: Remote Download Quelle .....	45
Tabelle 24: Unzulässige Jumper-Stellung JP16 auf dem Development Board.....	59
Tabelle 25: Unzulässige Jumper-Stellung JP30/JP33 auf dem Development Board.....	59
Tabelle 26: JP9 Konfiguration der Hauptversorgungsspannung VCC.....	61
Tabelle 27: JP9 Unzulässige Jumper-Stellungen der Hauptspannungsversorgung .....	61
Tabelle 28: JP28 Konfiguration des BOOT-Tasters.....	62
Tabelle 29: JP28 Konfiguration für permanenten Start des Bootstrap Loaders .....	63
Tabelle 30: JP22, JP23, JP10 Konfiguration von BOOT über RS-232.....	63
Tabelle 31: Unzulässige Jumper-Stellung beim BOOT über RS-232.....	63
Tabelle 32: Jumper-Konfiguration der DB-9 Buchse P1A (1. RS-232).....	64
Tabelle 33: Unzulässige Jumper-Stellungen bei der Konfiguration von P1A als erste RS-232 Schnittstelle .....	65
Tabelle 34: Jumper-Konfiguration der DB-9 Buchse P1B (ohne 2. RS-232) .....	66
Tabelle 35: Unzulässige Jumper-Stellungen bei der Konfiguration von P1B (ohne 2. RS-232).....	67
Tabelle 36: Jumper-Konfiguration der DB-9 Buchse P1B (UART, 2. RS-232) .....	68
Tabelle 37: Jumper-Konfiguration der DB-9 Buchse P1B (Software-Emulation der 2. RS-232).....	69

---

Tabelle 38: Unzulässige Jumper-Stellungen bei der Konfiguration von P1B (Software-Emulation 2. RS-232).....	70
Tabelle 39: Jumper-Konfiguration des CAN-Steckers P2A mit CAN-Treiber auf phyCORE-ST10F168/ST10F269 .....	71
Tabelle 40: Jumper-Konfiguration des CAN-Steckers P2A mit CAN-Treiber auf Development Board.....	72
Tabelle 41: Unzulässige Jumper-Stellungen CAN-Stecker P2A (CAN-Treiber auf Development Board) .....	73
Tabelle 42: Jumper-Konfiguration des CAN-Steckers P2A mit CAN-Treiber auf Development Board und galvanischer Trennung .....	74
Tabelle 43: Unzulässige Jumper-Stellungen bei galvanisch getrenntem CAN-Bus (CAN-Treiber auf Development Board) .....	75
Tabelle 44: Jumper-Konfiguration des CAN-Steckers P2B mit CAN-Treiber auf phyCORE-ST10F269 (nur mit ST10F269) .	76
Tabelle 45: Jumper-Konfiguration des CAN-Steckers P2B mit CAN-Treiber auf Development Board (nur mit ST10F269)....	77
Tabelle 46: Unzulässige Jumper-Stellungen CAN-Stecker P2B (CAN-Treiber auf Development Board, nur mit ST10F269)...	78
Tabelle 47: Jumper-Konfiguration des CAN-Steckers P2B mit CAN-Treiber auf Development Board und galvanischer Trennung (nur mit ST10F269) .....	79
Tabelle 48: Unzulässige Jumper-Stellungen bei galvanisch getrenntem CAN-Bus (CAN-Treiber auf Development Board) .....	80
Tabelle 49: JP17 Konfiguration der Visualisierungs-LED D3 .....	81
Tabelle 50: Daten/Adressbus-Pinzuordnung phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine.....	83
Tabelle 51: Pinzuordnung Port P2, P3, P4 phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine.....	84
Tabelle 52: Pinzuordnung Port P5, P6, P7, P8 phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine.....	85



---

Tabelle 53: Schnittstellen-Pinzuordnung phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine .....	86
Tabelle 54: Schnittstellen-Pinzuordnung phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine .....	86
Tabelle 55: Versorgungsspannungs-Zuordnung phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine .....	87
Tabelle 56: Nicht verwendete Pins des phyCORE-ST10F168/ST10F269 beim Development Board und der Erweiterungsplatine .....	88
Tabelle 57: JP28 Aktivierung /NMI Interrupt .....	89
Tabelle 58: JP19 Konfiguration des Nummernchip .....	89
Tabelle 59: Belegung Stiftleiste X3 des debugCORE-ST10F168/ST10F269 .....	92
Tabelle 60: Anschlußbelegung des Quad-Connectors (X6) auf dem debugADAPTER-167 .....	94



## Einleitung

Dieses Handbuch beschreibt nur die Schaltung und Funktionen des phyCORE-ST10F168/ST10F269, nicht aber die verschiedenen, bestückbaren Microcontroller selbst. Es wird ergänzt durch das entsprechende Microcontrollerhandbuch z.B. "ST10F168" von ST Microelectronics sowie die Dokumentation zu gegebenenfalls mitgelieferter Software. Bitte beachten Sie daher auch diese Dokumentationen.

In diesem Handbuch sowie im dazugehörigen Schaltplan werden low-aktive Signale durch einen Schrägstrich "/" vor dem Signalnamen gekennzeichnet (z.B. "/RD"). Die Darstellung "0" deutet auf eine logische Null oder low-Pegel hin, während "1" für eine logische Eins oder high-Pegel steht.

### Anmerkungen zum EMV-Gesetz für das phyCORE-ST10F168/ST10F269



Das phyCORE-ST10F168/ST10F269 (im Folgenden Produkt genannt) ist als Zulieferteil für den Einbau in ein Gerät (Weiterverarbeitung durch Industrie (siehe § 5 Abs. 5 EMVG)) bzw. als Evaluierungsboard für den Laborbetrieb (zur Hardware- und Softwareentwicklung) bestimmt.

#### **Achtung!**

Das Produkt ist ESD empfindlich und darf nur an ESD geschützten Arbeitsplätzen von geschultem Fachpersonal ausgepackt und gehandhabt bzw. verarbeitet werden. Im Betrieb dürfen ohne weitere Schutzbeschaltung und Prüfung keine Leitungen von mehr als 3 m Länge an die Verbinder angeschlossen werden.

Das Produkt erfüllt die Anforderungen des EMVG (CE-Konformität) nur für den in diesem Handbuch beschriebenen Anwendungsbereich unter Einhaltung der gegebenen Hinweise zur Inbetriebnahme.

Nach dem Einbau in ein Gerät oder bei Änderungen/Erweiterungen an diesem Produkt muß die Konformität nach dem EMV-Gesetz neu festgestellt und bescheinigt werden. Erst danach dürfen solche Geräte in Verkehr gebracht werden.

Auszug aus dem EMVG § 5 Abs. 5

Geräte, die ausschließlich zur Verwendung in eigenen Laboratorien, Werkstätten und Räumen hergestellt, Anlagen, die erst am Betriebsort zusammengesetzt werden, und Netze bedürfen keiner EG-Konformitätserklärung und CE-Kennzeichnung.

Dies gilt auch für Bausätze, die ausschließlich für Funkamateure im Sinne des § 1 Abs. 2 hergestellt und bestimmt sind.

Geräte, die ausschließlich als Zulieferteile oder Ersatzteile zur Weiterverarbeitung durch Industrie, Handwerk oder sonstige auf dem Gebiet der elektromagnetischen Verträglichkeit fachkundige Betriebe hergestellt und bereitgehalten werden, brauchen weder die Schutzanforderungen gemäß § 4 Abs. 1 einzuhalten noch bedürfen sie einer EG-Konformitätserklärung oder CE-Kennzeichnung, vorausgesetzt, es handelt sich dabei nicht um selbständig betreibbare Geräte.

Das phyCORE-ST10F168/ST10F269 ist ein Modul aus der Serie der nano-/micro-/mini-/phyCORE-Module der Firma PHYTEC, die eine Bestückung mit verschiedenen Microcontrollern erlauben, und dadurch eine Vielzahl von Funktionen und Konfigurationen ermöglichen.

PHYTEC unterstützt alle gängigen 8- und 16-bit-Microcontroller auf zwei Arten:

- (1) als Grundlage für Starter Kits, die die Kombination mit benutzer-eigenen Schaltungen auf einem eigens dafür vorgesehenen Wrap-Feld erlauben und
- (2) als universelle, sofort einsetzbare, voll funktionsfähige micro-, mini- und phyCORE-Module, die direkt in die benutzereigene Peripherie-Schaltung eingesteckt werden können.

Mit dem Konzept der Microcontroller-Module von PHYTEC ist es Entwicklungsingenieuren möglich, Entwicklungszeiten zu verkürzen, Entwicklungskosten zu reduzieren, und die Durchführung eines Projektes von der Idee bis zur Markteinführung wesentlich zu beschleunigen. Für weitere Informationen wenden Sie sich bitte an folgende Adressen:

	EUROPA	NORD AMERIKA
Adresse:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 D-55129 Mainz GERMANY	PHYTEC America LLC 203 Parfitt Way SW, Suite G100 Bainbridge Island, WA 98110 USA
Web Seite:	<a href="http://www.phytec.de">http://www.phytec.de</a>	<a href="http://www.phytec.com">http://www.phytec.com</a>
e-mail:	<a href="mailto:info@phytec.de">info@phytec.de</a>	<a href="mailto:info@phytec.com">info@phytec.com</a>
Tel.:	+49 (6131) 9221-0	+1 (800) 278-9913
Fax:	+49 (6131) 9221-33	+1 (206) 780-9135

## **1 Kurzübersicht über das phyCORE-ST10F168/ST10F269**

Das phyCORE-ST10F168/ST10F269 gehört zur Familie von PHYTECs phyCORE-Modulen. Die phyCORE Modulfamilie stellt die konsequente Weiterentwicklung der bekannten mini-, micro- und nanoMODULE von PHYTEC dar. Genauso wie mini-, micro- und nanoMODULE vereinen die phyCORE-Module alle notwendigen Komponenten eines Embedded Microcontroller Systems auf kleinster Fläche. Sie sind für die einfache Erweiterung bzw. Implementierung in unterschiedliche periphere Schaltungen und kundenspezifische Applikationen designed.

Die phyCORE Modulfamilie zeichnet sich unter anderem durch eine wesentlich erhöhte Anzahl von Anschlüssen aus. Dabei sind ca. 20 % aller Anschlüsse Masseverbindungen. Dies ist ein Resultat unabhängiger Untersuchungen, die gezeigt haben, daß ca. 70 % aller EMV-Probleme (elektromagnetische Verträglichkeit) auf unzureichende Masseverbindungen zurückzuführen sind. Dies trifft insbesondere in Umgebungen mit hochfrequenter Störstrahlung zu. Aufgrund der hohen Anzahl von Masseanschlüssen bei der phyCORE-Modulfamilie werden die EMV-Eigenschaften erheblich verbessert, wodurch der Einsatz von phyCORE-Modulen die einfache Einhaltung von EMV-Richtlinien selbst in stark gestörten Umgebungen ermöglicht.

phyCORE-Module sind durch die konsequente Anwendung moderner SMD-Technik, Multilayer-Design, kleinsten Bauteileformen und lasergebohrten Microvias extrem kompakt. Damit können Benutzer von phyCORE-Modulen diese 'state of the art' Techniken in ihrer eigenen Applikation implementieren und nutzen.

Unser Haus sieht sich in diesem Zusammenhang in der Verantwortung als zuverlässiger Partner und ISO 9001 zertifizierter, termintreuer Lieferant, den Einsatz unserer Microcontroller-Module in Ihrer Produktserie zu ermöglichen. Dies ist eine wichtige Voraussetzung für den Erfolg Ihres Produktes.

Das phyCORE-ST10F168/ST10F269 ist ein universelles Microcontrollerboard im Format 60 x 53 mm. Es kann mit den Microcontrollern ST10F168 und ST10F269 von ST Microelectronics bestückt werden.

Das universelle Design ermöglicht die einfache Integration des phyCORE-ST10F168/ST10F269 in eine Vielzahl von Applikationen. Da alle Microcontrollersignale sowie Ein- und Ausgänge auf die beiden Stiftleisten im Rastermaß 0,635 mm auf der Unterseite des Boards geführt sind, kann das phyCORE-ST10F168/ST10F269 wie ein großer Chip in eigene Zielhardware gesteckt werden.

Die controllerspezifischen Eigenschaften entnehmen Sie bitte dem Handbuch zum Microcontroller; im Hardware-Manual zum phyCORE-ST10F168/ST10F269 wird auf keinerlei Besonderheiten des Microcontrollers näher eingegangen, da diese für die grundlegende Funktion des phyCORE-ST10F168/ST10F269 ohne Belang sind.

### **Das phyCORE-ST10F168/ST10F269 bietet folgende Features:**

- Microcontroller-Board im Format 60 x 53 mm durch Einsatz moderner SMD-Technik
- bestückt mit ST Microelectronics ST10F168 or ST10F269 Microcontroller (QFP-144), unterstützt bis zu 2<sup>1</sup> on-chip CAN-Schnittstellen, Version 2.0B
- verbesserte Störsicherheit durch Multilayer-Technik sowie verringerte Störabstrahlung durch verbesserte Masseanbindung
- alle Ports sowie Daten- und Adressleitungen auf Platinenunterseite über zwei 100-polige high-density (0,635 mm) Molex-Verbinder verfügbar
- aufsetzbar auf die Anwendungsschaltung wie ein großer Chip

---

<sup>1</sup>: Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 zur Verfügung.

---

- 16-bit, demultiplexed Busmode
- 20 MHz CPU-Takt (100 ns Befehlszyklus)
- 16 MByte Adressraum
- 256 kByte on-chip Flash Speicher
- 256 kByte – 2 MByte Flash on-board<sup>1</sup>
- on-board Flash-Programmierung
- keine separate Programmierspannung durch Verwendung von 5 V Flash-Bausteinen
- 12 V VPP Programmierspannung nur für Programmierung des on-chip Flash beim ST10F168 erforderlich
- 256 kByte – 1 MByte RAM on-board<sup>1</sup>
- bis zu 2<sup>1</sup> CAN-Schnittstellen mit CAN-Treiber Philips 82C251 bzw. Temic Si9200EY
- I<sup>2</sup>C-Real-Time Clock mit internem Quarz
- 4 bis 8 kByte I<sup>2</sup>C-EEPROM<sup>1</sup>, altern. 512 Byte – 8 kByte FRAM<sup>1</sup>
- Voltage Supervisor Chip für Reset-Logik und Batterieüberwachung
- Remote - Supervisory Circuit<sup>2</sup>
- bis zu 2 freie Chip-Select Signale für einfachen Anschluß externer Peripherie
- einzige Versorgungsspannung 5 V, typ. < 220 mA
- RS-232 Transceiver für zwei serielle Schnittstellen
- optionaler UART für zweite asynchrone serielle Schnittstelle optional
- Unterstützung der Modemsignale CTS, RTS ,DTR und DSR über zweite serielle Schnittstelle (nur wenn optionaler UART bestückt)

---

<sup>1</sup> : Lassen Sie sich von PHYTEC über weitere Bestückungsvarianten beraten.

<sup>2</sup> : Dieses Feature befindet sich noch in der Entwicklung und steht noch nicht zur Verfügung.

---

## 1.1 Blockschaltbild

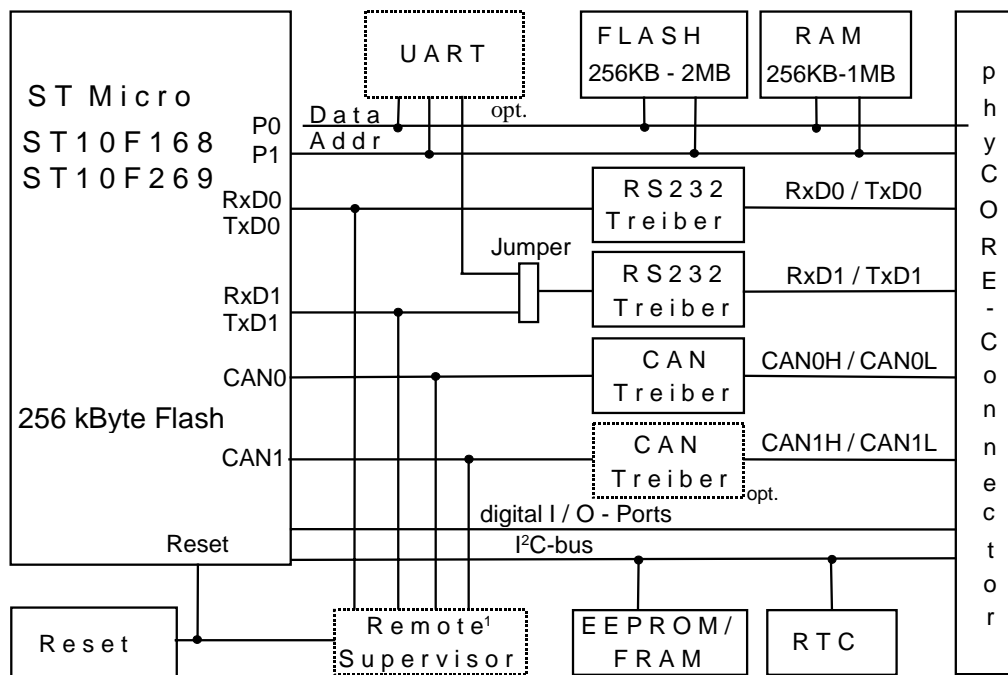


Bild 1: Blockschaltbild phyCORE-ST10F168/ST10F269

## 1.2 Ansicht des phyCORE-ST10F168/ST10F269

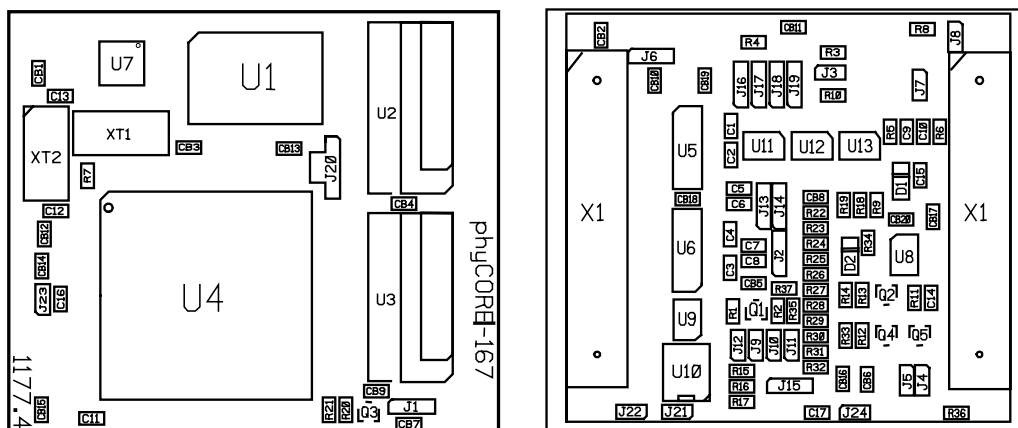


Bild 2: Ansicht des phyCORE-ST10F168/ST10F269



## 2 Anschlußbelegung

Es sei ausdrücklich darauf hingewiesen, daß bei allen Modulanschlüssen unbedingt die Maximalspannungen und -ströme nicht überschritten werden dürfen. Die Grenzwerte hierfür können Sie dem jeweiligen Controller-Handbuch entnehmen. Da eventuell auftretende Störungen stark vom Einsatzgebiet bzw. Anwendungsfall abhängen, obliegt es der Verantwortung des Anwenders, in entsprechend kritischer Umgebung geeignete Schutzmaßnahmen zu treffen.

Wie in *Bild 3* dargestellt, werden alle relevanten Signale an zwei Seiten des Moduls auf SMT-Stiftleisten im Rastermaß 0,635 mm (im folgenden phyCORE-Connector genannt) an den Platinenrand geführt. Damit kann das phyCORE-ST10F168/ST10F269 wie ein großer Chip in verschiedene Applikationen integriert werden.

Im Rahmen der phyCORE-Spezifikation wurde eine neue Zählweise für die Pins des phyCORE-Connectors eingeführt. Diese Zählweise erleichtert das Auffinden der Position von gesuchten Pins und beugt Mißverständnissen und Fehlern bei der Übertragung der Pinbelegung von den phyCORE-Modulen auf die entsprechenden Sockel der Anwendungsschaltung oder des Development Boardes vor.

Die Numerierung des phyCORE-Connectors in dieser neuen Zählweise ergibt sich aus einer zweidimensionalen Zählmatrix bei der den Spalten Buchstaben und den Zeilen Zahlen zugeordnet sind. Dabei befindet sich Pin 1A immer in der linken, oberen Ecke der Zählmatrix. Die Ziffern werden von dieser Ecke nach unten aufsteigend gezählt, während die Buchstaben nach rechts aufsteigen (*siehe Bild 3*).

Die Zählmatrix wird gedanklich von oben auf das phyCORE-ST10F168/ST10F269 (Draufsicht; phyCORE-Connector zeigt nach unten) oder auf den entsprechenden Sockel des phyCORE Development Boardes/der Anwendungsschaltung gelegt. Dabei wird die linke obere Ecke der Zählmatrix (Pin 1A) mit der durch ein weißes Dreieck markierten Ecke des phyCORE-ST10F168/ST10F269 zur Deckung gebracht. Selbst wenn ein phyCORE-Modul nur Kontakte auf der Unterseite führt, wird die Numerierung immer gleichermaßen durch das "Auflegen" der Zählmatrix auf die Oberseite bzw. Draufsicht definiert.

Der Vorteil dieser Vorgehensweise besteht darin, daß jeder Pin des phyCORE-Moduls die gleiche Pin-Nummer führt wie der korrespondierende Pin des entsprechenden Sockels auf dem phyCORE Development Board. Die Verwechslungsgefahr ist damit ausgesprochen gering.

Da die Pins über die oben beschriebene Zählmatrix exakt definiert sind, werden die Steckverbinder des phyCORE-Connectors jeweils nur unter einem einzelnen Bezeichner (beispielsweise X1) zusammengefaßt. Sie bilden somit immer eine logische Einheit, ungeachtet der Tatsache, daß es sich um mehrere physikalische Steckverbinder handeln kann. Die mit einem weißen Dreieck versehene Ecke im Bestückungsdruck gibt die Position der Zeile 1 an und erleichtert damit die Orientierung.

Das folgende Bild (siehe Bild 3) dient der Verdeutlichung der Zählmatrix. Es zeigt ein auf dem phyCORE Development Board steckendes phyCORE-ST10F168/ST10F269 mit seinen gestrichelt angedeuteten SMD-Steckverbindern auf der Unterseite. Des besseren Verständnisses wegen sind Teile des phyCORE-Moduls "weggebrochen", um die darunterliegenden Steckverbinder des Development Boardes anzudeuten.

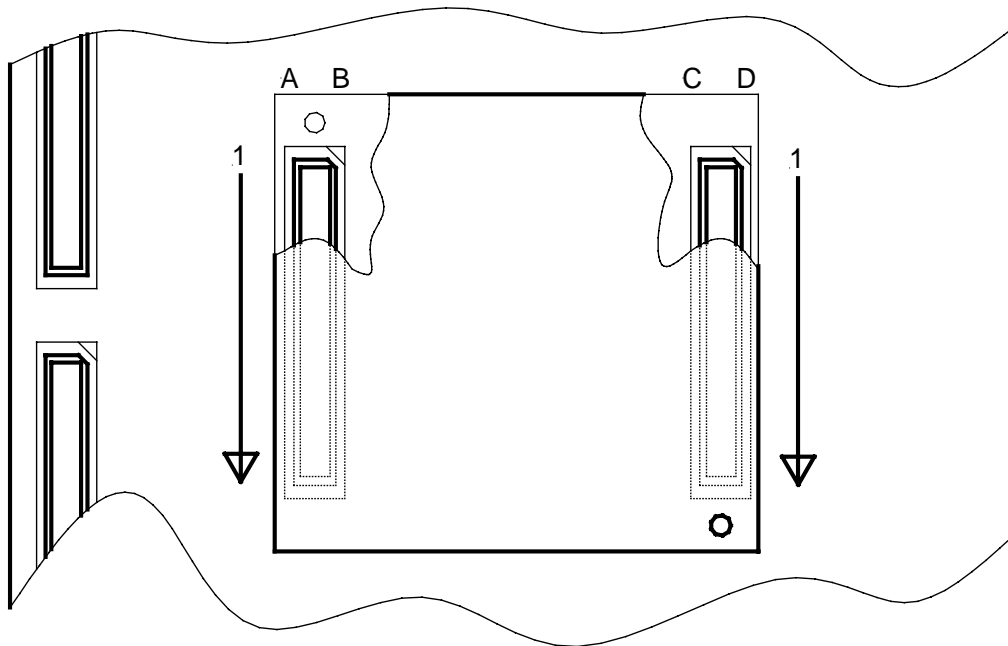


Bild 3: Pinout des phyCORE-Connectors (Drauf-/Durchsicht von oben)

Viele der am phyCORE-Connector verfügbaren Portpins des Microcontrollers sind mit alternativen Funktionen versehen, die in der Regel durch die Software entsprechend aktiviert werden müssen.

*Tabelle 1* gibt eine Übersicht über die Belegung des phyCORE-Connectors, sowie Hinweise auf mögliche Alternativfunktionen einiger Portpins. *Bitte ziehen Sie im Zweifelsfall das Data Sheet des auf dem phyCORE-ST10F168/ST10F269 bestückten Microcontrollers zu Rate.*

Pin-Nr.	Bezeichnung	E/A	Beschreibung
<b>Leiste X1A</b>			
1A	CLKIN	E	optionaler externer Taktgenerator
2A, 7A, 12A, 17A, 22A, 27A, 32A, 37A, 42A, 47A	GND	-	Schaltungsmasse 0 V
3A	P2.9	E/A	CAPCOM1:CC9 Capture Eing./Compare Ausgang Fast external Interrupt 1 Eingang (E)
4A	/NMI	E	Nicht maskierbarer Interrupt-Eingang
5A	P6.4, /CS4	A	Chip-Select #4
6A	ALE	A	Address-Latch-Enable
8A	/WRL	A	/WRL-Signal des $\mu$ C
9A, 10A, 11A, 13A, 14A, 15A, 16A, 18A, 24A, 25A, 26A, 28A	A1, A2, A4, A7, A9, A10, A12, A15, A17, A18, A20, A23	A	Adressleitungen des $\mu$ C
19A, 20A, 21A, 23A, 29A, 30A, 31A, 33A	D1, D2, D4, D7, D9, D10, D12, D15	E/A	Datenleitung des $\mu$ C
34A	/RDY	E	/READY-Signal des $\mu$ C
35A	NC	-	Unbelegt Diese Kontakte sollten auf der Applikationsseite ebenfalls unbelegt bleiben.
36A	P6.6, /HLDA	E/A	Acknowledge Ausgang (master mode)/ Eingang (slave mode)
38A, 39A	P7.1, P7.3	A	POUT1 PWM Kanal 1 POUT3 PWM Kanal 3
40A, 41A	P7.4, P7.6	E/A	CAPCOM2:CC28 Capture Eing./Compare Ausg. CAPCOM2:CC30 Capture Eing./Compare Ausg.
43A	P3.9	E/A	SSC Master Transmit / Slave Receive
44A	P3.0	E	CAPCOM1 Timer T0 Zähler-Eingang
45A	P3.1	A	GPT2 Timer T6 Latch-Ausgang
46A	P3.3	A	GPT1 Timer T3 Latch-Ausgang
48A	P3.6	E	GPT1 Timer T3 Zähler-Eingang
49A	P6.0, /CS0	A	Chip-Select #0
50A	P6.1, /CS1	A	Chip-Select #1

Leiste X1B			
1B	P3.15	A	CLKOUT System Clock Ausgang
2B, 3B	P2.8, P2.10	E/A	CAPCOM1: CC8 Capture Eing./Compare Ausg. Fast ext. Interrupt 0 Eingang (E) CAPCOM1: CC10 Capture Eing./Compare Ausg. Fast ext Interrupt 2 Eingang (E)
4B, 9B, 14B, 19B, 24B, 29B, 34B, 39B, 44B, 49B	GND	-	Schaltungsmasse
5B	P6.3, /CS3	A	Chip-Select #3
6B	P6.2, /CS2	A	Chip-Select #2
7B	/RD	A	/RD-Signal des $\mu$ C
8B, 10B, 11B, 12B, 13B, 15B, 16B, 17B, 23B, 25B, 26B, 27B,	A0, A3, A5, A6, A8, A11, A13, A14, A16, A19, A21, A22	A	Adressleitungen des $\mu$ C
18B, 20B, 21B, 22B, 28B, 30B, 31B, 32B	D0, D3, D5, D6, D8, D11, D13, D14	E/A	Datenleitungen des $\mu$ C
33B	P3.12, /WRH	A	/WRH-Signal des $\mu$ C
35B	P6.5, /HOLD	E	/HOLD-Signal des $\mu$ C
36B	P6.7, /BREQ	A	/BREQ-Signal des $\mu$ C
37B, 38B	P7.0, P7.2	A	POUT0 PWM Kanal 0 POUT2 PWM Kanal 2
40B, 41B	P7.5, P7.7	E/A	CAPCOM2:CC29 Capture Eing./Compare Ausg. CAPCOM2:CC31 Capture Eing./Compare Ausg.
42B, 43B, 45B, 46B, 47B, 48B	P3.8, P3.13, P3.2, P3.4, P3.5, P3.7	E/A	Port 3 des $\mu$ C ( <i>siehe Controller-Handbuch</i> )
50B	NC	-	Unbelegt Diese Kontakte sollten auf der Applikationsseite ebenfalls unbelegt bleiben.

Leiste X1C			
1C, 2C	VCC	-	Versorgungsspannung +5 V =
3C, 7C, 12C, 17C, 22C, 27C, 32C, 37C	GND	-	Schaltungsmasse 0 V
4C, 5C	NC	-	Unbelegt Diese Kontakte sollten auf der Applikationsseite ebenfalls unbelegt bleiben.
6C	VBAT	E	Batterie-Eingang zur Pufferung von RTC und RAM
8C	/PFO	A	MAX 690 /Power-Fail-Ausgang
9C	BOOT	E	Eingang zum Starten der FlashTools
10C	/RESET	E	/RESET-Eingang des phyCORE-ST10F168/F269
11C	/RESOUT	A	/RESOUT-Signal des $\mu$ C
13C, 14C, 15C, 16C, 19C, 20C	P2.2, P2.4, P2.5, P2.7, P2.11, P2.12	E/A	Port 2 des $\mu$ C ( <i>siehe Controller-Handbuch</i> )
18C	CAN-H1 <sup>1</sup>	E/A	Differentielle CANH-Leitung des zweiten CAN-Treibers
21C	RxD1_RS-232	E	Eingang der zweiten seriellen Schnittstelle des phyCORE-ST10F168/ST10F269, RS-232-Pegel
23C	TxD1_RS-232	A	Ausgang der zweiten seriellen Schnittstelle des phyCORE-ST10F168/ST10F269, RS-232-Pegel
24C	/RTS1_RS-232	A	/RTS Signal des UART U7, RS-232-Pegel
25C	/CTS1_RS-232	E	/CTS Signal des UART U7, RS-232-Pegel
26C	/DSR1_RS-232	E	/DSR Signal des UART U7, RS-232-Pegel
28C	/DTR1_RS-232	A	/DTR Signal des UART U7, RS-232-Pegel
29C	/RI1_TTL	E	/RI Signal des UART U7, TTL-Pegel
30C	/CD1_TTL	E	/CD Signal des UART U7, TTL-Pegel
31C	SCL	A	CLK-Leitung I <sup>2</sup> C-Bus
33C	IRQ_UART	A	Interrupt-Ausgang des UART U7
34C	/CS_UART	E	Chip-Select Signal des UART U7
35C, 36C	P8.4, P8.6	E/A	CAPCOM2:CC20 Capture Eing./Compare Ausg. CAPCOM2:CC22 Capture Eing./Compare Ausg.
38C, 39C, 40C	NC	-	Unbelegt Diese Kontakte sollten auf der Applikationsseite ebenfalls unbelegt bleiben.
41C, 43C, 44C, 45C, 46C, 48C, 49C, 50C	P5.14, P5.11, P5.9, P5.8, P5.6, P5.3, P5.1, P5.0	E/A	Port 5 des $\mu$ C ( <i>siehe Controller-Handbuch</i> )
42C, 47C	VAGND	-	Analog-Masses des $\mu$ C

Leiste X1D			
1D, 2D	VCC	-	Versorgungsspannung +5 V =
3D, 9D, 14D, 19D, 24D, 29D, 34D	GND	-	Schaltungsmasse 0 V
4D, 5D	VPP	-	Programmierspannung für internes Flash. Nur verwendbar, wenn ST10F168 Controller bestückt! Die Verwendung des RDP Pins für verschiedene Reset-Funktionen beim ST10F269 erfordert eine zusätzliche externe Beschaltung am Anschluss VPP ( <i>siehe Controller Handbuch</i> ).
6D	VPD	A	Ausgang der gepufferten Versorgungsspannung zur Pufferung externer Komponenten
7D	PFI	E	MAX 690 Power-Fail-Eingang. Dieser Eingang muß, falls ungenutzt, an VCC oder GND angeschlossen werden
8D	WDI	E	MAX 690 Watchdog-Eingang
10D	/RESET	E	/RESET-Eingang des phyCORE-ST10F168/ST10F269
11D, 12D, 13D, 15D	P2.0, P2.1, P2.3, P2.6	E/ A	Port 2 des µC ( <i>siehe Controller-Handbuch</i> )
16D	P3.11, RxD0	E	Eingang der ersten seriellen Schnittstelle, TTL-Pegel
17D	P3.10, TxD0	A	Ausgang der ersten seriellen Schnittstelle, TTL-Pegel
18D	CAN-L1 <sup>1</sup>	E/ A	Differentielle CANL-Leitung des zweiten CAN-Treibers
20D	CAN-L0	E/ A	Differentielle CANL-Leitung des ersten CAN-Treibers
21D	CAN-H0	E/ A	Differentielle CANH-Leitung des ersten CAN-Treibers
22D	RxD0_RS-232	E	Eingang der ersten seriellen Schnittstelle, RS-232-Pegel
23D	TxD0_RS-232	A	Ausgang der ersten seriellen Schnittstelle, RS-232-Pegel
25D, 26D	P2.14, P2.15	E/ A	CAPCOM1: CC14 Capture Eing./Compare Ausg. Fast ext. Interrupt 6 Eingang (E) CAPCOM1: CC15 Capture Eing./Compare Ausg. Fast ext Interrupt 7 Eingang (E) T7IN Timer T7 Count Eingang (E)
27D, 28D, 30D, 31D	P8.0, P8.1, P8.2, P8.3	E/ A	CAPCOM2:CC16 Capture Eing./Compare Ausg. CAPCOM2:CC17 Capture Eing./Compare Ausg. CAPCOM2:CC18 Capture Eing./Compare Ausg. CAPCOM2:CC19 Capture Eing./Compare Ausg.
32D	SDA	A	Datenleitung I <sup>2</sup> C-Bus
33D	/IRQ_RTC	A	Interrupt-Ausgang der RTC
35D, 36D	P8.5, P8.7	E/ A	CAPCOM2:CC21 Capture Eing./Compare Ausg. CAPCOM2:CC23 Capture Eing./Compare Ausg.
37D	P2.13	E/ A	CAPCOM1:CC13 Capture Eing./Compare Ausg. Fast ext Interrupt 5 Eingang (E)
38D	NC	-	Unbelegt Diese Kontakte sollten auf der Applikationsseite ebenfalls unbelegt bleiben.
39D, 44D, 49D	VAGND	-	Analog-Masse des µC
40D, 41D, 42D, 43D, 45D, 46D, 47D, 48D,	P5.15,P5.13,P5.12, P5.10,P5.7, P5.5, P5.4, P5.2	E	Port 5 des µC ( <i>siehe Controller-Handbuch</i> )
50D	VAREF	E	Referenzspannungs-Eingang für A/D-Wandler

Tabelle 1: Pinout des phyCORE-Connectors X1

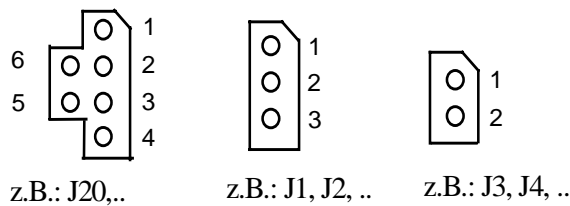
<sup>1</sup>: Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 Microcontroller zur Verfügung



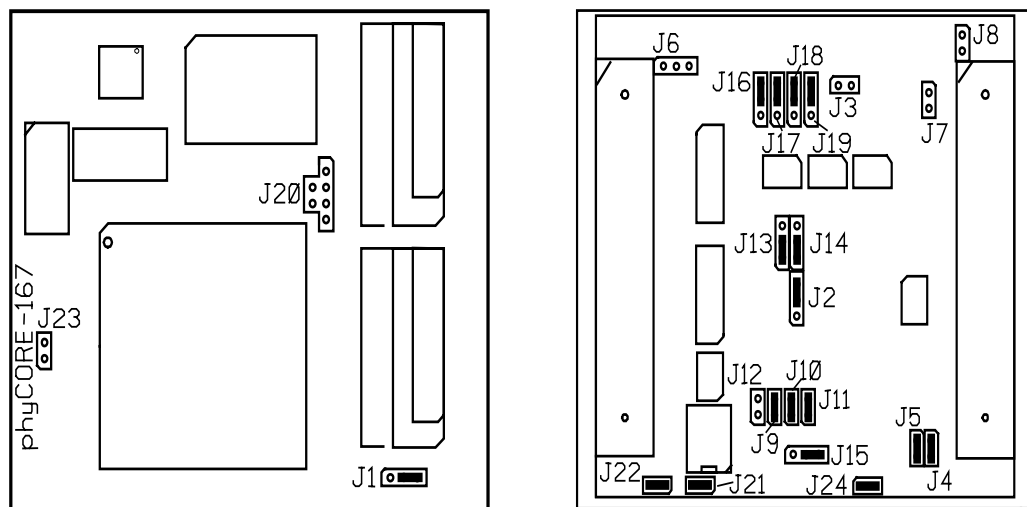


### 3 Jumper

Das phyCORE-ST10F168/ST10F269 besitzt zur Konfiguration 24 Lötjumper, die teilweise bereits bei der Auslieferung vorverbunden sind. *Bild 4* verdeutlicht die verwendete Zählweise bei den Jumpern, *Bild 5* die Lage der Jumper auf dem Board. Auf dem phyCORE-ST10F168/ST10F269 befinden sich alle Lötjumper bis auf J1, J20 und J23 auf der Platinenunterseite.



*Bild 4: Zählweise der Jumper*



*Bild 5: Lage der Jumper (Ansicht Microcontrollerseite / Connectorseite)*

Die Jumper (J = Lötjumper) haben folgende Funktionen:

	<b>Default-Einstellung<sup>1</sup></b>		<b>Alternative Einstellung</b>	
<b>J1</b>	(2+3)	VPD an Pin 30 des RAM (für RAM-Speicher < 512 kByte)	(1+2)	A18 an Pin 30 des RAM (für RAM-Speicher > 512 kByte)
<b>J2</b>	(2+3)	externes ROM/ Flash aktiv	(1+2)	internes ROM/Flash-EEPROM aktiv
<b>J3</b>	(offen)	Port P4.4 (A20) zur Verwendung als CAN2-Empfangsleitung verwendbar <sup>2</sup>	(geschl.)	Adressleitung A20 an externem Flash
<b>J4</b>	(geschl.)	VAREF aus Versorgungsspannung VCC abgeleitet	(offen)	VAREF von externer Spannungsquelle über Pin X1D50
<b>J5</b>	(geschl.)	VAGND aus Digitalmasse GND abgeleitet	(offen)	VAGND von externer Masse an Pins X1C42, X1C47, X1D39, X1D44 und X1D49 <sup>3</sup>
<b>J6</b>	ST10F168: (1+2) ST10F269: (offen)	VPP an Pin 84 des Microcontrollers keine Funktion, RDP Signal nicht abgreifbar <sup>4</sup>	ST10F168: (offen) (2+3) ST10F269: (1+2) (2+3)	VPP nicht mit Pin 84 des Microcontrollers verbunden unzulässig RPD des Microcontrollers an VPP-Anschluss verfügbar <sup>5</sup> RPD des Microcontrollers an GND
<b>J7</b>	(offen)	P2.8 des $\mu$ C als Standard I/O an Pin X1B2 frei verfügbar	(geschl.)	IRQ des UART an Pin P2.8 des $\mu$ C angeschlossen
<b>J8</b>	(offen)	/CS2 des $\mu$ C an Pin X1B6 frei verfügbar	(geschl.)	/CS2 des $\mu$ C an externen UART angeschlossen
<b>J9</b>	(geschl.)	P3.4 des $\mu$ C an SCL des I <sup>2</sup> C-Busses angeschlossen	(offen)	P3.4 des $\mu$ C als Standard I/O an Pin X1B46 verfügbar
<b>J10</b>	(geschl.)	P3.3 des $\mu$ C an SDA des I <sup>2</sup> C-Busses angeschlossen	(offen)	P3.3 des $\mu$ C als Standard I/O an Pin X1A46 verfügbar

<sup>1</sup>: Gültig für Standardmodul ohne Zusatzoption.

<sup>2</sup>: Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 Microcontroller zur Verfügung.

<sup>3</sup>: Bei der Verwendung des Development Board phyCORE-HD200 5V sind diese Pins fest mit der Masse des Development Boardes verbunden. D.h. der Anschluß einer externen Masse ist nicht möglich.

<sup>4</sup>: Weitere Informationen zur Verwendung des RDP Pins beim ST10F269 finden Sie im *Controller Handbuch*.

<sup>5</sup>: Die Verwendung des RDP Pins für verschiedene Reset-Funktionen beim ST10F269 erfordert eine zusätzliche externe Beschaltung am Anschluss VPP (*siehe Controller Handbuch*).

<b>J11</b>	(geschl.)	IRQ der RTC an Pin P2.9 des $\mu$ C angeschlossen	(offen)	P2.9 des $\mu$ C als Standard I/O an Pin X1A3 verfügbar
<b>J12</b>	(offen)	Schreibschutz des E <sup>2</sup> PROM/FRAM-Bausteins deaktiviert	(geschl.)	evtl. vorhandene Schreibschutz des E <sup>2</sup> PROM/FRAM-Bausteins aktiviert ( <i>siehe Datenblatt</i> )
<b>J13</b>	(1+2)	RS-232 Treiber (TxD) für zweite serielle Schnittstelle an P3.0 des $\mu$ C's angeschlossen	(2+3)	RS-232 Treiber (TxD) für zweite serielle Schnittstelle an UART angeschlossen (von Modulkonfiguration abhängig)
<b>J14</b>	(1+2)	RS-232 Treiber (RxD) für zweite serielle Schnittstelle an P3.1 des $\mu$ C's angeschlossen	(2+3)	RS-232 Treiber (RxD) für zweite serielle Schnittstelle an UART angeschlossen (von Modulkonfiguration abhängig)
<b>J15</b>	(2+3)	Adresse des seriellen Speicherbausteins U9 auf 0xA8 eingestellt ( <i>siehe Datenblatt</i> )	(1+2)	Adresse des seriellen Speicherbausteins U9 auf 0xAC eingestellt ( <i>siehe Datenblatt</i> )
<b>J16<sup>1</sup></b>	(2+3)	CAN0-Sendesleitung (CANTx) des CAN-Bus-treiber U11 verbunden mit Port P4.6 (A22) des $\mu$ C's ( <i>siehe Controller-Handbuch</i> )	(1+2)	ST10F168/ST10F269: unzulässig <sup>2</sup>
<b>J17<sup>1</sup></b>	(2+3)	CAN0-Empfangsleitung (CANRx) des CAN-Bus-treiber U11 verbunden mit Port P4.5 (A21) des $\mu$ C's ( <i>siehe Controller-Handbuch</i> )	(1+2)	ST10F168/ST10F269: unzulässig <sup>2</sup>
<b>J18<sup>3</sup></b>	(2+3)	CAN1-Sendeleitung (CANTx) des CAN-Bus-treiber U12 verbunden mit Port P4.7 (A23) des $\mu$ C's ( <i>siehe Controller-Handbuch</i> )	(1+2)	ST10F168/ST10F269: unzulässig <sup>2</sup>
<b>J19<sup>3</sup></b>	(2+3)	CAN1-Empfangsleitung (CANRx) des CAN-Bus-treiber U12 verbunden mit Port P4.4 (A20) des $\mu$ C's ( <i>siehe Controller-Handbuch</i> )	(1+2)	ST10F168/ST10F269: unzulässig <sup>2</sup>

<sup>1</sup> : **Achtung!** Bei der Verwendung des ST10F168/F269 reduziert die Nutzung der CAN-Schnittstelle den verfügbaren Adreßraum auf 1 MByte pro /CS-Signal.

<sup>2</sup> : Die Benutzung von Port P8 für die CAN-Schnittstelle ist mit dem ST10F168/ST10F269 nicht möglich.

<sup>3</sup> : Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 Microcontroller zur Verfügung.

<b>J20</b>	(offen)	keine Remote Download Quelle angeschlossen	(1+2) (3+5) (3+4) (2+6)	Remote Download Quelle P3.1 Remote Download Quelle CAN1Rx Remote Download Quelle CAN2Rx Remote Download Quelle P3.11
<b>J21</b>	(geschl.) <sup>1</sup>	P3.11 als RXD0 mit RS-232 Treiber verbunden	(offen)	P3.11 des µC als Standard I/O an Pin X1D16 frei verfügbar
<b>J22</b>	(geschl.) <sup>1</sup>	P3.10 als TXD0 mit RS-232 Treiber verbunden	(offen)	P3.10 des µC als Standard I/O an Pin X1D17 frei verfügbar
<b>J23</b>	(geschl.)	Pin 17 des µC's an VCC	(offen) <sup>2</sup>	Pin 17 des µC's über Abblockkondensator an GND
<b>J24</b>	(geschl.)	Pin 56 de µC's an VCC	(offen) <sup>2</sup>	Pin 56 des µC's über Abblockkondensator an GND

Tabelle 2: Jumperbelegung

<sup>1</sup> : **Achtung!** Sollten diese Jumper auf dem phyCORE-ST10F168/ST10F269 unbestückt sein, so ist es nicht möglich, mit den PHYTEC FlashTools oder mit einem BOOT-Monitor auf das Modul zuzugreifen.

<sup>2</sup> : **Achtung !** Bei Verwendung des ST10F269 ist unbedingt darauf zu achten, daß die Jumper geöffnet sind.

### 3.1 J1 Verwendung von Pin 30 des SRAM

Über Jumper J1 wird die Verwendung des Pins 30 des SRAM (U2, U3) festgelegt. Bei SRAM-Bausteinen mit einer Größe von 128 kByte ist das Pin 30 als high-aktives CS-Signal definiert und daher mit einem Highpegel zu beschalten, um den Zugriff auf den SRAM zu ermöglichen. Bei größeren SRAM-Bausteinen ab 512 kByte wird Pin 30 als Adressleitung A17 verwendet und ist im Falle einer 16-bit-Organisation des SRAM's mit der Adressleitung A18 des Microcontrollers zu verbinden.

**Auf dem phyCORE-ST10F168/ST10F269 ist J1 wie folgt zu verwenden:**

Besitzt der externe RAM-Speicher eine Größe von 2 x 128 kByte, muß Jumper J1 auf Position 2+3 geschlossen werden. Damit liegt VPD an den Pins 30 der SRAMs an. Bei einem Speicherausbau von 2 x 512 kByte SRAM muß A17 des SRAM mit der Adresse A18 des Microcontrollers verbunden werden (Jumper J1 auf 1+2). Dies resultiert aus der Tatsache, daß sich der Speicherausbau aus zwei 8-bit SRAM-Bausteinen zusammensetzt. Die Adresse A0 des Microcontrollers dient intern zur Auswahl des Speicherbausteins und damit zur Generierung der Write-Signale /WRL und /WRH.

Es ergeben sich folgende Konfigurationen:

Konfiguration	J1
2 x 128 kByte SRAM (U2, U3)	2 + 3*
2 x 512 kByte SRAM (U2, U3)	1 + 2

\* = Default-Einstellung

*Tabelle 3: J1 SRAM Pin 30 Beschaltung*

### 3.2 J2 Interner oder externer Programmspeicher

Der Jumper J2 ist bei der Auslieferung auf Position 2+3 geschlossen. Dadurch wird nach einem Hardware-Reset das im externen Programmspeicher abgelegte Programm abgearbeitet. Um eine Abarbeitung des internen Programmspeichers im ST10F168/ST10F269 Controller zu ermöglichen, muß Jumper J2 auf Position 1+2 geschlossen werden.

Es ergeben sich folgende Konfigurationen:

Code-Zugriff	J2
externer Programmspeicher	2 + 3*
interner Programmspeicher	1 + 2

\* = Default-Einstellung

Tabelle 4: J2 Auswahl interner oder externer Programmspeicher

### 3.3 J3 Flash-Adressierung

Jumper J3 verbindet die Adressleitung A20 des Controllers mit der Adresse A19 des Flash-Baustein U1. Bei einem Flash Speicher mit einer Größe unter 2 MByte muß Jumper J3 generell geöffnet sein, um einen Konflikt mit der zweiten CAN-Schnittstelle<sup>1</sup> zu vermeiden. Ist ein 2 MByte Flash-Baustein auf dem phyCORE-ST10F168/ST10F269 bestückt, so muß Jumper J3 geschlossen werden. Die Nutzung der CAN-Schnittstelle(n)<sup>1</sup> ist in diesem Fall nicht möglich<sup>2</sup>.

---

<sup>1</sup>: Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 Microcontroller zur Verfügung.

<sup>2</sup>: Durch Verwendung der CAN-Schnittstelle(n) verringert sich der Adressraum pro /CS Signal auf 1 MByte (siehe Controller Handbuch).

Es ergeben sich folgende Konfigurationen:

<b>Flash-Adressierung</b>	<b>J3</b>
Flash Speicher < 2 MByte, P4.4 als Standard I/O oder als CAN1 Empfangsleitung nutzbar <sup>1</sup>	offen*
Flash Speicher = 2 MByte, P4.4 dient als A20 zur Adressierung des Flash (CAN-Schnittstelle(n) nicht nutzbar) <sup>2</sup>	geschlossen

\* = Default-Einstellung

Tabelle 5: J3 Flash-Adressierung

### 3.4 J4, J5 A/D-Wandler

Der A/D-Wandler des phyCORE-ST10F168/ST10F269 benötigt an den Pins 37 und 38 eine obere und untere Referenzspannung ( $V_{AREF}$ ,  $V_{AGND}$ ). Mit den Jumpern J4 und J5 wird die Referenzspannungsquelle ausgewählt.

<b>Funktion</b>	<b>J4</b>	<b>J5</b>
externe Referenzspannungsquelle ( $V_{AREF}$ an X1D50, $V_{AGND}$ an X1C42, X1C47, X1D39, X1D44 und X1D49)	offen	offen <sup>3</sup>
$V_{AREF}$ aus Versorgungsspannung VCC abgeleitet	geschlossen*	
$V_{AGND}$ aus Digitalmasse GND abgeleitet		geschlossen*

\* = Default-Einstellung

Tabelle 6: J4, J5 Referenzspannung des A/D-Wandlers

- 
- <sup>1</sup>: Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 Microcontroller zur Verfügung.
  - <sup>2</sup>: Durch Verwendung der CAN-Schnittstelle(n) verringert sich der Adressraum pro /CS Signal auf 1 MByte (siehe Controller Handbuch).
  - <sup>3</sup>: Bei der Verwendung des Development Board phyCORE-HD200 5V sind diese Pins fest mit der Masse des Development Boardes verbunden. Der Anschluß einer externen Masse ist daher nicht möglich.
-

### 3.5 J6 On-Chip Flash Programmierspannung / RDP

Pin 84 hat in Abhängigkeit des verwendeten Microcontrollers unterschiedliche Funktionen. Bei Verwendung eines ST Microelectronics ST10F168 wird Pin 84 für die Einspeisung der 12 Volt Programmierspannung VPP verwendet. Ist das phyCORE-ST10F168/ST10F269 mit einem ST10F269 bestückt, dann wird Pin 84 als Timing-Pin RPD für das Verlassen des Power-Down-Modes benutzt. Jumper J6 aktiviert die unterschiedlichen Funktionen.

Folgende Konfigurationen sind möglich:

<b>Funktion des Pin 84</b>	<b>J6</b>
ST10F168: VPP (12 V) zur Programmierung des on-chip Flash an Pin 84 angeschlossen	1 + 2*
ST10F269: RPD Anschluss (Pin 84) ist offen	offen*
ST10F168: VPP Anschluss des phyCORE-Connectors ist von Pin 84 getrennt	offen
ST10F269: RPD Anschluss (Pin 84) ist am VPP-Anschluss des phyCORE-Connectors verfügbar <sup>1</sup>	1 + 2
ST10F168: unzulässig! ST10F269: RPD Anschluss (Pin 84) an GND angeschlossen	2 + 3

\* = Default-Einstellung

Tabelle 7: J6 On-Chip Flash Programmierspannung / RDP Signal

---

<sup>1</sup>: Bei Verwendung des RDP Pins für verschiedene Reset-Funktionen beim ST10F269 erfordert eine zusätzliche externe Beschaltung am Anschluss VPP (siehe *Controller Handbuch*).

---



### 3.6 J7, J8 Nutzung des externen UART

Das phyCORE-ST10F168/ST10F269 kann optional mit einem UART auf U7 bestückt werden. Dieser stellt eine zweite serielle RS-232 Schnittstelle zur Verfügung. Die Aktivierung des externen UART erfolgt mit Hilfe von /CS2. Wie mit dem on-chip UART des ST10F168/ST10F269 kann auch die serielle Kommunikation über den externen UART interruptgesteuert erfolgen. Hierzu wird der externe Interrupt 0 an Port P2.8 verwendet. Der Anschluß von /CS2 und Port P2.8 erfolgt über die Jumper J7 und J8.

Es ergeben sich folgende Konfigurationen:

<b>Verwendung Port P2.8</b>	<b>J7</b>
P2.8 des $\mu$ C als Standard I/O an Pin X1B2 frei verfügbar	offen*
IRQ des optionalen externen UART an Pin P2.8 des $\mu$ C angeschlossen	geschlossen

<b>Chip-Select /CS2</b>	<b>J8</b>
/CS2 des $\mu$ C an Pin X1B6 frei verfügbar	offen*
/CS2 des $\mu$ C an optionalen externen UART angeschlossen	geschlossen

\* = Default-Einstellung

Tabelle 8: J7, J8 Nutzung optionale externe UART

### 3.7 J9, J10 Konfiguration von P3.3 und P3.4 für I<sup>2</sup>C-Bus

Auf dem phyCORE-ST10F168/ST10F269 ist eine Real-Time Clock (RTC, U10) und ein serielles E<sup>2</sup>PROM/FRAM (U9) mit I<sup>2</sup>C-Schnittstelle vorhanden. Diese werden über die Jumper J9 und J10 mit den Portpins P3.3 und P3.4 verbunden. Um die Pins als Standard I/O-Pins an X1A46 (P3.3) und X1B46 (P3.4) verwenden zu können, müssen die Jumper J9 und J10 geöffnet werden.

Port P3.3 und P3.4 Konfiguration	J10	J9
Port P3.3 als I/O-Pin an X1A46	offen	
Port P3.3 als I <sup>2</sup> C-SDA	geschlossen*	
Port P3.4 als I/O-Pin an X1B46		offen
Port P3.4 als I <sup>2</sup> C-SCL		geschlossen*

\* = Default-Einstellung

Tabelle 9: J9, J10 Konfiguration P3.3 und P3.4 / I<sup>2</sup>C-Bus

### 3.8 J11 Interrupt-Ausgang der RTC

Mit Jumper J11 kann ausgewählt werden, ob der Interrupt-Ausgang der RTC (U10) mit Port 2.9 verbunden ist. Wenn der Jumper offen bleibt, kann P2.9 als Portpin an X1A3 verwendet werden.

Port P2.9 Konfiguration	J11
Port P2.9 als I/O-Pin an X1A3	offen
Port P2.9 als /INT-Eingang für RTC	geschlossen*

\* = Default-Einstellung

Tabelle 10: J11 Konfiguration P2.9 / RTC Interrupt

### 3.9 J12 Schreibschutz des E<sup>2</sup>PROM /FRAM

Verschiedene auf U9 bestückbare E<sup>2</sup>PROM/FRAM Bausteine verfügen über eine Schreibschutzfunktion<sup>1</sup>. Diese kann über Jumper J12 aktiviert werden. Dazu wird durch Schließen des Jumpers Pin 7 des seriellen E<sup>2</sup>PROM/FRAM mit VCC verbunden.

<b>E<sup>2</sup>PROM/FRAM Schreibschutzfunktion</b>	<b>J12</b>
Schreibschutz des E <sup>2</sup> PROM/FRAM deaktiviert	offen*
Schreibschutz des E <sup>2</sup> PROM/FRAM aktiviert	geschlossen

\* = Default-Einstellung

*Tabelle 11: J12 Konfiguration E<sup>2</sup>PROM/FRAM Schreibschutz*

### 3.10 J13, J14 Konfiguration der zweiten seriellen Schnittstelle

Mit den Jumpern J13 und J14 kann die Signalquelle der zweiten seriellen Schnittstelle gewählt werden. Alternativ zu einer software-emulierten Schnittstelle an den Portpins P3.0 und P3.1 des Controllers besteht die Möglichkeit, einen externen UART auf dem phyCORE-ST10F168/ST10F269 zu bestücken. Bei Verwendung der optionalen externen UART steht eine vollwertige zweite serielle Schnittstelle zur Verfügung. In diesem Fall können die Portpins P3.0 und P3.1 als Standard-I/O an X1A44 (P3.0) und X1A45 (P3.1) genutzt werden.

Es ergeben sich folgende Konfigurationen:

<b>Schnittstelle</b>	<b>J13</b>	<b>J14</b>
P3.0/P3.1 an RS-232 Treiber angeschlossen (Software-emulierte zweite Schnittstelle)	1 + 2*	1 + 2*
optionale UART U7 an RS-232 Treiber angeschlossen (echte zweite Schnittstelle)	2 + 3	2 + 3

\* = Default-Einstellung

*Tabelle 12: J13, J14 Konfiguration zweite serielle Schnittstelle*

---

<sup>1</sup>: Bitte informieren Sie sich über das Vorhandensein der Schreibschutzfunktion im Datenblatt des installierten E<sup>2</sup>PROM/FRAM.

### 3.11 J15 Adresse des seriellen E<sup>2</sup>PROM /FRAM

Mit Jumper J15 wird die Adresse des seriellen E<sup>2</sup>PROM/FRAM konfiguriert. Standardmäßig ist die Adresse des E<sup>2</sup>PROM/FRAM auf 0xA8 eingestellt.

Folgende Alternativen sind möglich:

Adresse des E <sup>2</sup> PROM/FRAM	J15
0xA8	2 + 3*
0xAC	1 + 2

\* = Default-Einstellung

*Tabelle 13: J15 Konfiguration Adresse serielles E<sup>2</sup>PROM/FRAM*

### 3.12 J16, J17, J18, J19 CAN-Schnittstellen

Die CAN-Schnittstellen des phyCORE-ST10F168/ST10F269 befinden sich an den Portleitungen P4.5 (CAN1Rx) und P4.6 (CAN1Tx) für die erste CAN-Schnittstelle und den Portleitungen P4.4 (CAN2Rx) und P4.7 (CAN2Tx) für die zweite CAN-Schnittstelle<sup>1</sup>. Diese Signale sind entsprechend an die beiden CAN-Transceiver U11 und U12 (PCA82C251 bzw. Si9200EY) geführt und dienen der Generierung der Signale CANH0, CANL0, CANH1 und CANL1, die direkt mit einem CAN-Zweidrahtbus verbunden werden können. Hierfür sind die Lötjumper J16, J17, J18 und J19 entsprechend zu konfigurieren.

Um einen externen Transceiver zu nutzen, können die Signale CAN1Rx, CAN1Tx, CAN2Rx und CAN2Tx auch direkt am phyCORE-Connector X1 abgegriffen werden. Hierbei müssen die Jumper J16, J17, J18 und J19 geöffnet werden.

---

<sup>1</sup>: Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 Microcontroller zur Verfügung.

Die Verwendung der CAN-Schnittstelle(n) verhindert die volle Nutzung des kompletten Adressbereiches des Controllers von bis zu 16 MByte. Bitte beachten Sie weiterhin, das die Jumper J16 - J19 nicht in der Stellung (1+2) konfiguriert werden. *Bitte entnehmen Sie detaillierte Hinweise zur Bedienung der CAN-Schnittstellen der Controllerbeschreibung von ST Microelectronics bzw. entsprechenden Publikationen zum CAN-Bus.*

Es ergeben sich folgende Konfigurationen:

<b>CAN1-Port</b>	<b>J16</b>	<b>J17</b>
P4.5 (CAN1Rx) P4.6 (CAN1Tx)	2 + 3*	2 + 3*
Nicht zulässig mit ST10F168/ST10F269	1 + 2	1 + 2

<b>CAN2-Port<sup>1</sup></b>	<b>J18</b>	<b>J19</b>
P4.4 (CAN2Rx) P4.7 (CAN2Tx)	2 + 3*	2 + 3*
Nicht zulässig mit ST10F168/ST10F269	1 + 2	1 + 2

\* = Default-Einstellung

*Tabelle 14: J16, J17, J18 und J19 Konfiguration CAN-Schnittstellen*

### **Achtung!**

Bei der Verwendung des ST10F168/ST10F269 reduziert die Nutzung der CAN-Schnittstelle den verfügbaren Adreßraum auf 1 MByte pro /CS-Signal.

<sup>1</sup>: Die zweite CAN-Schnittstelle steht nur bei Bestückung mit einem ST Microelectronics ST10F269 Microcontroller zur Verfügung.

### 3.13 J20 Remote Download Quelle

Der auf dem Modul vorhandene Remote Supervisor Chip<sup>1</sup> (U8) dient dem ferngesteuerten Einleiten der Bootsequenz (*siehe Kapitel 9*). Er kann über verschiedene Schnittstellen angesteuert werden. Default-seitig ist Jumper J20 offen, da diese Funktion zur Zeit noch nicht zur Verfügung steht.

Es ergeben sich folgende Konfigurationen:

Download Quelle	J20
keine	offen*
Port P3.11 / RxD0	2 + 6
Port P3.1 / RxD1	1 + 2
Port CAN1Rx	3 + 5
Port CAN2Rx	3 + 4

\* = Default-Einstellung

Tabelle 15: J20 Konfiguration Remote Download Quelle

### 3.14 J21, J22 Serielle Schnittstelle

Jumper J21 und J22 dienen dazu, die Schnittstellen-Signale der asynchronen seriellen Schnittstelle über den RS-232-Treiber nach außen an die Pins X1D22 (RxD0) und X1D23 (TxD0) der Stiftleiste zu führen. Sollten die Jumper unbestückt sein, so besteht die Möglichkeit, die entsprechenden Controller-Pins (P3.10 und P3.11) mit alternativen Funktionen zu versehen oder die Schnittstellensignale mit TTL-Pegel an die entsprechenden Pins des phyCORE-Connectors zu legen (X1D17, X1D16).

#### **Achtung!**

Sollten diese Jumper auf dem phyCORE-ST10F168/ST10F269 unbestückt sein, so ist es nicht möglich mit den PHYTEC FlashTools oder mit einem BOOT-Monitor auf das Modul zuzugreifen.

---

<sup>1</sup>: Dieses Feature befindet sich noch in der Entwicklung und steht noch nicht zur Verfügung.

Sollten die Jumper bestückt sein, so ist es zu vermeiden die Signale mit ihren TTL-Pegel zu verwenden, da dies zu Zerstörung von Bauteilen führen kann.

Signal	J21	J22
TxD0 und RxD0 mit RS-232 Pegel	geschlossen*	geschlossen*
P3.10 und P3.11 als I/O Pin oder TxD0 und RxD0 als Schnittstellen-signale mit TTL Pegel	offen	offen

\* = Default-Einstellung

Tabelle 16: J21, J22 Konfiguration erste serielle Schnittstelle

### 3.15 J23, J24 Versorgungsspannung des Microcontrollers

Die beiden Jumper J23 und J24 sind speziell zur Bestückung mit einem ST Microelectronics ST10F269 Microcontroller vorgesehen. Diese Jumper sind notwendig, da der ST10F269 an den entsprechenden Controller-Pins (Pin 17 und Pin 56) keine VCC-Spannung (5 V) erhalten darf. Die betreffenden Pins des Controllers werden zur Abblockung der intern im Controller erzeugten CORE-Spannung benötigt.

#### **Achtung!**

Bei Verwendung des ST10F269 ist unbedingt darauf zu achten, dass die Jumper geöffnet sind.

Bei Verwendung des ST10F168 ist unbedingt darauf zu achten, dass die Jumper geschlossen sind.

Anschluss VCC Pins	J23	J24
VCC an Controller angeschlossen	geschlossen <sup>*1</sup>	geschlossen <sup>*1</sup>
C16 und C17 wirken als Abblockkondensatoren	offen <sup>2</sup>	offen <sup>2</sup>

<sup>\*1</sup> Default-Einstellung bei phyCORE-ST10F168

<sup>2</sup> Default-Einstellung bei phyCORE-ST10F269

Tabelle 17: J23, J24 Konfiguration VCC Microcontroller





## 4 System-Konfiguration

Nach einem Reset (Hardware- oder Software-Reset) startet der ST10F168/ST10F269 die Programmausführung ab der Adresse 00:0000H. In der Regel wird dort ein Sprung auf eine applikationsspezifische Initialisierungsroutine zu finden sein, in der bestimmte Features des Controllers konfiguriert werden. Diese Routine wird in einem privilegierten Modus ausgeführt, der durch Ausführung der EINIT-Instruktion abgeschlossen wird. Danach ist der Zugriff auf bestimmte Register sowie die Ausführung bestimmter Instruktionen eingeschränkt.

Obwohl die meisten Features des ST10F168/ST10F269 während der genannten Initialisierungsroutine konfiguriert bzw. programmiert werden, sind einige bestimmte Features früher zu konfigurieren, da sie direkten Einfluß auf die Programmausführung haben.

### 4.1 System-Startup-Konfiguration

Die System-Startup-Konfiguration dient dazu, diejenigen Features des ST10F168/ST10F269 zu konfigurieren, die einen direkten Einfluß auf die Programmausführung und somit auf die korrekte Ausführung der Initialisierungsroutine haben. Hierzu zählt vor allem die Vorgabe einiger grundlegender Eigenschaften der externen Bus-Schnittstelle zum Speicher des Moduls hin (z.B. Datenbreite, Multiplex- oder Demultiplex-Modus).

Zur erwähnten Einstellung werden Teile des Ports P0 während Reset durch den Controller eingelesen und wichtige Features gemäß den logischen Eingangspegeln an den entsprechenden Pins gesetzt. Durch Verbindung der gewünschten Pins des Port 0 mit Pull-Down-Widerständen (ergibt logischen Zustand 0) bzw. Nichtbeschaltung (ergibt logischen Zustand 1) kann die sogenannte System-Startup-Konfiguration definiert werden.

Wir empfehlen 4,7 k $\Omega$ -Widerstände als Pull-Downs, wobei die Größenangabe lediglich als Richtwert dienen kann, da je nach externer Datenbus-Beschaltung des Moduls abweichende Werte zu verwenden sind.

Die folgende *Tabelle 18* zeigt die Funktion der einzelnen Bits des Port P0 während des Systemstarts und ihre Lage auf dem phyCORE-Connector.

Funktion des Port P0 während des System Reset (High-Byte)							
Bit H7	H6	H5	H4	H3	H2	H1	Bit H0
CLKCFG <i>R31, R29, R28</i> <i>1 1 1</i>			SALSEL <i>R27, R26</i> <i>0(1<sup>1</sup>) 0</i>		CSSEL <i>R25, R24</i> <i>0 0</i>		WRC <i>R23</i> <i>0</i>

Funktion des Port P0 während des System Reset (Low-Byte)							
Bit L7	L6	L5	L4	L3	L2	L1	Bit L0
BUSTYP <i>R22</i> <i>1 0</i>		SMOD <i>R21</i> <i>Pin 21B 0 Pin 20B Pin20A</i>				ADP <i>Pin 19A</i>	EMU <i>Pin 18B</i>

Tabelle 18: Belegung des Port 0 für die System-Startup-Konfiguration

Reservierte Pins müssen auf logischem 1 Pegel gehalten werden, um die korrekte Funktion des ST10F168/ST10F269 zu gewährleisten  
Die Konfiguration dieser Pins darf nicht verändert werden

<sup>1</sup>: Bei einem Speicherausbau mit 2 MByte Flash Speicher (PCM-009-x3x) muß das Register SALSEL mit den Bitwerten 1 0 konfiguriert sein.

Die Funktionen im Detail:

Name	Wert	Bedeutung	Bemerkung
CLKCFG	1 1 1	CPU Takt = Ext. Takt * 4	definiert CPU Takt
	1 1 0	CPU Takt = Ext. Takt * 3	
	1 0 1	CPU Takt = Ext. Takt * 2	
	1 0 0	CPU Takt = Ext. Takt * 5	
	0 1 1	CPU Takt = Ext. Takt * 1	
	0 1 0	CPU Takt = Ext. Takt * 1.5	
	0 0 1	CPU Takt = Ext. Takt * 0.5	
	0 0 0	CPU Takt = Ext. Takt * 2.5	
SALSEL	1 1	Adreßleitungen A16..A17, I/O-Pins P4.2..P4.7	definiert Verhalten der Portpins P4.0..P4.7
	1 0 <sup>12</sup>	Adreßleitungen A16..A23, keine I/O-Pins	
	0 1	keine Adreßleitungen, I/O-Pins P4.0..P4.7	
	0 0 <sup>2</sup>	Adreßleitungen A16..A19, I/O-Pins P4.4..P4.7	
CSSEL	1 1	Chip-Selects /CS0../CS4, keine I/O-Pins	definiert Verhalten der Portpins P6.0..P6.4
	1 0	Keine Chip-Selects, I/O-Pins P6.0..P6.4	
	0 1	Chip-Selects /CS0../CS1, I/O-Pins P6.2..P6.4	
	0 0	Chip-Selects /CS0../CS2, I/O-Pins P6.3..P6.4	
WRC	1	/WR und /BHE	definiert Verhalten der Pins /WR und
	0	/WRL und /WRH	P3.12
BUSTYP	1 1	16-bit Multiplex Bus	definiert Businterface
	1 0	16-bit Demultiplex Bus	für /CS0 (BUSCON0)
	0 1	8-bit Multiplex Bus	
	0 0	8-bit Demultiplex Bus	

<sup>1</sup>: Bei einem Speicherausbau mit 2 MByte Flash Speicher (PCM-009-x3x) muß das Register SALSEL mit den Bitwerten 1 0 konfiguriert sein.

<sup>2</sup>: **Achtung!** Bei der Verwendung des ST10F168/ST10F269 reduziert die Nutzung der CAN-Schnittstelle(n) den verfügbaren Adreßraum auf 1 MByte pro /CS-Signal.

Name	Wert	Bedeutung	Bemerkung
BSL	1	Bootstrap-Loader inaktiv	
	0	Bootstrap-Loader aktiv	
ADP	1	Adapt-Modus inaktiv	
	0	Adapt-Modus aktiv	
EMU	1	Emulation-Modus inaktiv	
	0	Emulation-Modus aktiv	

Tabelle 19: System-Startup-Konfiguration

Default-System-Startup-Konfiguration auf dem  
phyCORE-ST10F168/ST10F269

Die grundlegenden Einstellungen der System-Startup-Konfiguration sollten unbedingt anhand einer entsprechenden Initialisierungsroutine verfeinert werden, da bestimmte Einstellungen auf diesem Wege nicht konfiguriert werden können. Hierzu zählen unter anderem die Anzahl der Waitstates für die einzelnen Speicherbereiche bzw. Chip-Selects sowie deren Lage im Adreßraum des Controllers.

Einige Entwicklungsumgebungen benutzen eine spezielle Datei, welche zur Definition derartiger Systemeinstellungen dient und beim Übersetzungs- bzw. Linkprozess mit einzubeziehen ist (z.B. *start167.a66* bei der Entwicklungsumgebung der Fa. KEIL).

## 5 Speichermodell

Für den einfachen Anschluß externer Peripheriebausteine oder zusätzlicher Speicherbänke stellt der ST10F168/ST10F269 Controller bis zu fünf Chip-Select Signale an Port P6 zur Verfügung. Abhängig von der Anzahl der bestückten Peripheriebausteine werden bis zu drei dieser Signale intern verwendet. Die Flash-Bank (U1), die entweder mit 256 kByte, 512 kByte, 1 MByte oder 2 MByte Flash Speicher bestückt ist, wird über /CS0 (P6.0) des Controllers selektiert. Auf der RAM Bank U2/U3 können 128 kByte oder 512 kByte Speicherbausteine im SO28-32 Gehäuse eingesetzt werden. Dies ergibt bei Minimalbestückung einen Speicherausbau von 256 kByte, bei Maximalbestückung 1 MByte. Die RAM Bank U2/U3 wird über /CS1 (P6.1) des Controllers selektiert. Der optional bestückbare UART U7 wird bei Verwendung über /CS2 (P6.2) angesprochen.

Um die Chip-Select Signale zu verwenden, müssen sie während des Reset aktiviert werden (*siehe Kapitel 4.1*). Die Zuordnung der Chip-Select Signale zu bestimmten Speicherbereichen wird durch die Konfiguration der entsprechenden ADDRESELx- und BUSCONx-Register durchgeführt. Beachten Sie, daß zunächst das entsprechende ADDRESELx-Register auf den gewünschten Bereich eingestellt werden muß, bevor mit dem BUSCONx-Register der Zugriff auf diesen Bereich mit dem /CSx-Signalen aktiviert wird. Es ist darauf zu achten, daß sich keine /CS-Signale überschneiden und der Bereich des Programmcodes bei /CS0 nicht durch ein anderes /CSx-Signal ausgeblendet wird.

Nach Reset ist zunächst nur /CS0 über dem gesamten Adreßraum des Controllers aktiv (d.h. P6.0 an Flash-Bank = U1 ). /CS0 ist immer in allen Bereichen aktiv, in denen kein anderes /CSx-Signal aktiviert wurde.

Durch Einstellung von einem Waitstate ( $T_c = 50 \text{ ns}$ ) und dem R/W-Delay, lassen sich alle Speicher mit Zugriffszeiten von bis zu 100 ns bei einem Buszyklus von 150 ns ansprechen. Um den Controller ohne Waitstates betreiben zu können, müssen 55 ns Speicher bestückt sein. Die Buszykluszeit beträgt dann 100 ns. Das R/W-Delay sollte immer aktiv sein (*siehe ST10F168/ST10F269 User's Manual für weitere Informationen*).

Im folgenden finden Sie wichtige Signalzeiten. Alle Informationen beziehen sich auf ST10F168/ST10F269 Controller mit 16-bit Bus, im Demultiplexed Modus und 20 MHz CPU-Takt ( $F_{osz}$ ):

$T_c = 50 \text{ ns} * \text{Waitstates Control (MCTC in BUSCON)}$

$T_f = 50 \text{ ns} * \text{Tri-State Control (MTTC in BUSCON)}$ .

Adressen stabil bis Daten gültig:	max.	$70 \text{ ns} + T_c$	$SR^1$
/RD low bis Daten gültig:	max.	$55 \text{ ns} + T_c$	$SR^1$
/RD low bis Daten gültig (R/W-Delay):	max.	$30 \text{ ns} + T_c$	$SR^1$
/RD high bis Datenbus high-Z:	max.	$15 \text{ ns} + T_f$	$SR^1$
/RD high bis Daten high-Z (R/W-Delay):	max.	$35 \text{ ns} + T_f$	$SR^1$
/CSx bis Daten gültig:	max.	$55 \text{ ns} + T_c$	$SR^1$
/RD und /WR low:	min.	$65 \text{ ns} + T_c$	$CC^2$
/RD und /WR low (R/W-Delay):	min.	$40 \text{ ns} + T_c$	$CC^2$
Daten gültig bis /WR high:	min.	$25 \text{ ns} + T_c$	$CC^2$
/WR high bis Daten ungültig:	min.	$15 \text{ ns} + T_f$	$CC^2$

Im weiteren finden Sie zwei Beispiele für die Einteilung der Speicher des phyCORE-ST10F168/ST10F269. Diese Beispiele sind für die meisten Anwendungen nutzbar.

---

<sup>1</sup>:  $SR$  = System Zeit (Zeit muß von der Beschaltung des ST10F168/ST10F269 eingehalten werden)

<sup>2</sup>:  $CC$  = Controller Charakteristik (Der Controller gewährleistet diese Zeiten für seine Außenbeschaltung)

---

### Beispiel a)

ADDRESEL1: 0406h = Bereich 04:0000h - 07:FFFFh  
(256 kByte RAM Bank auf U2/3)

ADDRESEL2: 0800h = Bereich 08:0000h - 08:0FFFh  
(4 kByte Adressbereich für ext. UART)

ADDRESEL3<sup>1</sup>: 0816h = Bereich 08:1000h – 0C:0FFFh  
(256 kByte freier I/O Bereich)

ADDRESEL4<sup>1</sup>: 0C16h = Bereich 0C:1000h - 10:0FFFh  
(256 kByte freier I/O Bereich)

BUSCON0: 04AFh:Bus aktiv für /CS0 (Flash-Bank U1)

BUSCON1: 04AFh:Bus aktiv für /CS1 (RAM-Bank U2/3)

BUSCON2: 042Fh:Bus aktiv für /CS2 (ext. UART)

BUSCON3<sup>1</sup>: 068Ch:Bus aktiv für /CS3 (freier I/O)

BUSCON4<sup>1</sup>: 068Ch:Bus aktiv für /CS4 (freier I/O )

BUSCON0-2: für alle 55 ns Speicher aktiv (0 Waitstate, RW-Delay, kein Tri-state, kurzes ALE, 16-bit Demultiplexed)

BUSCON3,4: für freien I/O Bereich (3 Waitstate, R/W-Delay, Tri-state Wait 300 ns, langes ALE, 16-bit Demultiplexed)

### Beispiel b)

ADDRESEL1: 0006h = Bereich 00:0000h - 03:FFFFh  
(256 kByte RAM-Bank auf U2/3)

ADDRESEL2: 0806h = Bereich 08:0000h - 08:0FFFh  
(4 kByte Adressbereich für ext. UART)

ADDRESEL3<sup>1</sup>: 0816h = Bereich 08:1000h – 0C:0FFFh  
(256 kByte freier I/O Bereich)

ADDRESEL4<sup>1</sup>: 0C16h = Bereich 0C:1000h - 10:0FFFh  
(256 kByte freier I/O Bereich)

BUSCON0: 04AFh:Bus aktiv für /CS0 (Flash Bank U1)

BUSCON1: 04AFh:Bus aktiv für /CS1 (RAM Bank U2/3)

BUSCON2: 042Fh:Bus aktiv für /CS2 (ext. UART)

BUSCON3<sup>1</sup>: 068Ch:Bus aktiv für /CS3 (freier I/O)

BUSCON4<sup>1</sup>: 068Ch:Bus aktiv für /CS4 (freier I/O)

BUSCON0-2: für alle 55 ns Speicher aktiv (0 Waitstate, RW-Delay, kein Tri-state, kurzes ALE, 16-bit Demultiplexed)

BUSCON3,4: für freien I/O Bereich (3 Waitstate, RW-Delay, Tri-state, langes ALE, 16-bit Demultiplexed)

---

<sup>1</sup>: /CS3 und /CS4 sind bei der Standard-Konfiguration des phyCORE-ST10F168/ST10F269 nicht aktiviert. Um diese Signale zu nutzen, müssen die Widerstände R24 und R25 entfernt werden (siehe Kapitel 4).

Beispiel a)		Beispiel b)	
FF:FFFFh	P6.0 (/CS0) FLASH Bank Spiegelungen	FF:FFFFh	P6.0 (/CS0) FLASH Bank Spiegelungen
10:1000h 10:0FFFh	256 kByte I/O	10:1000h 10:0FFFh	256 kByte I/O
0C:1000h 0C:0FFFh	P6.4 (/CS4)	0C:1000h 0C:0FFFh	P6.4 (/CS4)
08:1000h 08:0FFFh	256 kByte I/O	08:1000h 08:0FFFh	256 kByte I/O
08:1000h 08:0FFFh	P6.3 (/CS3)	08:1000h 08:0FFFh	P6.3 (/CS3)
08:0000h 07:FFFFh	4 kByte ext.UART P6.2 (/CS2)	08:0000h 07:FFFFh	4 kByte ext. UART P6.2 (/CS2)
04:0000h 03:FFFFh	256 kByte RAM Bank U2/U3 P6.1 (/CS1)	04:0000h 03:FFFFh	256 kByte FLASH Bank U1 P6.0 (/CS0)
00:0000h	256 kByte FLASH Bank U1 P6.0 (/CS0)	00:0000h	256 kByte RAM Bank U2/U3 P6.1 (/CS1)

Bild 6: Beispiele für Speichermodelle

## 5.1 Bus-Timing

Für den Anschluß von externen Speicherbausteinen sollte die Einstellung des entsprechenden BUSCON-Registers wie folgt lauten:

BUSCONx: 04AEh = 1 Waitstate, R/W-Delay, kein Tri-state, kurzes ALE, 16-bit Demultiplexed, Adress-CSx aktiv (gilt für alle Speicherbausteine bis 70 ns Zugriffszeit auf dem phyCORE-ST10F168/ST10F269)

Mit diesen Einstellungen wird ein Waitstate und das R/W-Delay aktiviert. Mit der Konfiguration eines Waitstates (1 Waitstate :  $T_C = 50$  ns) und einem R/W-Delay lassen sich Speicher mit bis zu 70 ns Zugriffszeit bei einem Buszyklus von 150 ns einsetzen.



## **6 Serielle Schnittstellen**

### **6.1 RS-232 Schnittstelle**

Auf dem phyCORE-ST10F168/ST10F269 befindet sich ein RS-232-Treiber (U6) für die Pegelanpassung der Leitungen P3.11/RxD0 und P3.10/TxD0 sowie der zweiten seriellen Schnittstelle P3.1/RxD1 und P3.0/TxD1. Die Ports P3.11, P3.10, P3.0 und P3.1 lassen sich alternativ auch als Standard I/O am phyCORE-Connector X1 nutzen. Hierzu müssen die Jumper J21, J22 bzw. J13 und J14 geöffnet werden.

Ist der optionale UART-Baustein auf dem phyCORE-ST10F168/ST10F269 bestückt, kann durch Schließen der Jumper J13 und J14 auf Position 2+3 eine Pegelanpassung für die zweite serielle Schnittstelle erfolgen. Beide herausgeführten Schnittstellen können für die Verbindung zu einer COM-Schnittstelle (z.B. PC) verwendet werden. Hierzu ist die RxD-Leitung mit der TxD-Leitung der COM-Schnittstelle und die TxD-Leitung mit der RxD-Leitung der COM-Schnittstelle zu verbinden. Die Schaltungsmasse des phyCORE-ST10F168/ST10F269 ist zusätzlich an die Schaltungsmasse der COM-Schnittstelle anzuschließen.

Durch die on-chip UART des Microcontrollers werden keine Handshake-Leitungen unterstützt. Diese können bei Bedarf durch Port Pins des Microcontrollers nachgebildet werden. Für die Unterstützung eines RS-232-Pegels für diese Handshake-Leitungen ist außerhalb des Moduls ein RS-232-Treiber vorzusehen.

Bei Bestückung des optionalen UART-Bausteins U7 kann zusätzlich ein RS-232-Treiber (U5) bestückt werden. Dieser Treiber unterstützt die RS-232-Pegelanpassung der von der externen UART hardwareseitig erzeugten Handshake-Signale.

#### **Achtung!**

Sollten die Jumper J21 und J22 auf dem phyCORE-Modul unbestückt sein, so ist es nicht möglich mit den FlashTools oder mit einem BOOT-Monitor auf das Modul zuzugreifen.

## 6.2 CAN-Schnittstelle

Auf dem phyCORE-ST10F168/ST10F269 sind für die Pegelanpassung der CAN-Sende-/Empfangsleitungen (CANTx / CANRx) zwei CAN-Bustreiber (U11, U12; PCA82C251 oder Si9200EY) vorgesehen. Die CAN-Bustreiber unterstützen bis zu 110 Knoten an einem CAN-Bus. Die Datenübertragung erfolgt mit einem differentiellen Pegel auf CANH und CANL. Eine Masseverbindung zwischen den Knoten am CAN-Bus ist nicht zwingend erforderlich, wird jedoch für eine Verbesserung der EMV-Eigenschaften empfohlen. Für die korrekte Funktion der Datenübertragung auf dem CAN-Bus sind 2 Abschlußwiderstände, je ein 120 Ohm-Widerstand am Ende des CAN-Busses, vorzusehen.

Bei größeren Busausdehnungen wird weiterhin eine externe galvanische Entkopplung zwischen CAN-Bustreiber und dem phyCORE-ST10F168/ST10F269 empfohlen. In diesem Fall sind die Leitungen CANTx und CANRx mit Hilfe von Jumper J16, J17, J18 und J19 vom on-board Bustreiber zu trennen. Für die galvanisch getrennte Verbindung zum externen Bustreibers sollten schnelle Optokoppler verwendet werden. Hierfür eignen sich die Typen TLP113 der Fa. Toshiba oder HCPL06xx der Fa. Hewlett Packard. Für die korrekte Busanschaltung sind die Empfehlungen DS102 vom CiA<sup>1</sup> zu berücksichtigen.

### **Achtung!**

Bei der Verwendung des ST10F168/ST10F269 reduziert die Nutzung der CAN-Schnittstelle den verfügbaren Adreßraum auf 1 MByte pro /CS-Signal.

---

<sup>1</sup>: Die CiA wurde im März 1992 mit dem Ziel gegründet, die Entwicklung des Controller Area Netzwerk (CAN) zu fördern und Richtlinien für zukünftige Erweiterungen des CAN-Protokolls festzulegen. Dazu stellt sie sowohl technische als auch Produkt- und Marketinginformationen zur Verfügung.

## 7 Echtzeituhr RTC-8563 (U10)

Für Echtzeitanwendungen ist das phyCORE-ST10F168/ST10F269 mit einer Echtzeituhr des Typs RTC-8563 ausgestattet. Diese Echtzeituhr bietet die folgenden Funktionen:

- Serielle Kommunikation über I<sup>2</sup>C Bus
- Stromaufnahme
  - Bus aktiv: max. 50 mA
  - Bus inaktiv, CLKOUT = 32 kHz: max. 1,7 A
  - Bus inaktiv, CLKOUT = 0 kHz: max. 0,75 µA
- Uhrfunktion mit 4-Jahres Kalender
- Jahrhundertbit
- Universeller Timer mit Alarm- und Überlaufanzeige
- 24-Stunden Format
- Automatische Word-Adressen Inkrementierung
- Programmierbare Alarm-, Timer- und Interruptfunktion

Wird das phyCORE-ST10F168/ST10F269 über eine Batterie gepuffert, läuft die Echtzeituhr auch nach Ausfall der Spannungsversorgung des Boards weiter.

Die Programmierung der Echtzeituhr erfolgt über I<sup>2</sup>C-Bus (Adresse 0xA2 = 10100010), der mit Hilfe der Ports P3.4 (SCL) und P3.3 (SDA) realisiert ist. Zusätzlich verfügt die Echtzeituhr über einen Interruptausgang, der über Jumper J11 auf den Port P2.9 führt. Ereignisse für einen Interrupt können ein Uhrenalarm, ein Timeralarm, ein Timerüberlauf bzw. ein Alarm des Ereigniszählers sein. Alle Interrupts müssen durch Software gelöscht werden. Durch die Interruptfunktion ist die Echtzeituhr für die unterschiedlichsten Applikationen verwendbar. (*Weitere Informationen über die Register der Echtzeituhr finden Sie im beigelegten Datenblatt.*)

### **Achtung!**

Nach Anlegen der Versorgungsspannung bzw. nach einem Reset generiert die Echtzeituhr **keinen** Interrupt, da sie hierfür erst initialisiert werden muß.



## 8 Serielles EEPROM/FRAM (U9)

Das phyCORE-ST10F168/ST10F269 verfügt über einen nichtflüchtigen Speicher mit einem seriellen Interface (I<sup>2</sup>C-Interface) zur Ablage von Konfigurationsdaten. Je nach Bestückungsvariante kann dieser Speicher mit einem EEPROM in der Größe von 4 - 32 kByte oder mit einem FRAM bestückt werden.

Eine Beschreibung des I<sup>2</sup>C-Protokolls ist dem Datenblatt des Bauteins zu entnehmen.

*Tabelle 20* gibt einen Überblick zu bestückbaren Bausteinen zum Zeitpunkt der Drucklegung des Handbuchs.

Speichertyp	Größe	Baustein	Hersteller
EEPROM	4 kByte	AT25320	ATMEL
	8 kByte	AT25640	ATMEL
	32 kByte	24WC256	CATALYST
FRAM	512 Byte	FM25040	RAMTRON
	8 kByte	FM25160	RAMTRON

*Tabelle 20: Bestückungsoptionen für U9*

Verschiedene E<sup>2</sup>PROM/FRAM verfügen über eine Schreibschutzfunktion<sup>1</sup>. Diese kann über Jumper J12 aktiviert werden. Dazu wird durch Schließen des Jumpers Pin 7 des seriellen E<sup>2</sup>PROM/FRAM mit VCC verbunden.

Schreibschutz des E <sup>2</sup> PROM/FRAM	J12
Schreibschutz ist deaktiviert	offen*
Schreibschutz ist aktiviert	geschlossen

\* = Default-Einstellung

*Tabelle 21: E<sup>2</sup>PROM/FRAM Schreibschutz*

---

<sup>1</sup>: Bitte informieren Sie sich über das Vorhandensein der Schreibschutzfunktion im Datenblatt des installierten E<sup>2</sup>PROM/FRAM.

---

Mit Jumper J15 wird die Adresse des seriellen E<sup>2</sup>PROM/FRAM konfiguriert. Standardmäßig ist die Adresse des E<sup>2</sup>PROM/FRAM auf 0xA8 eingestellt.

Folgende Alternativen sind möglich:

Adresse des E <sup>2</sup> PROM/FRAM	J15
0xA8	2 + 3*
0xAC	1 + 2

\* = Default-Einstellung

*Tabelle 22: E<sup>2</sup>PROM/FRAM Adresse*

## 9 Remote Supervisor Chip (U8)

U8 ist vorgesehen für einen Remote Supervisor Chip. Mit diesem IC kann über ein serielles Interface (RS-232 oder CAN) ein Bootvorgang ausgelöst werden. Damit wird die Bedingung für den Start der PHYTEC FlashTools geschaffen. Dies ermöglicht, ohne die Betätigung eines BOOT-Jumpers oder -Tasters, ferngesteuert ein Update der Software im Flash einzuleiten.

Da der Remote Supervisor Chip noch in Entwicklung ist, kann das Feature erst bei zukünftigen Modulen angeboten werden. Jumper J20 bleibt daher offen.

Es ergeben sich folgende Konfigurationen:

Download Quelle	J20
keine	offen*
Port P3.11 / RxD0	2 + 6
Port P3.1 / RxD1	1 + 2
Port CAN1Rx	3 + 5
Port CAN2Rx	3 + 4

\* = Default-Einstellung

*Tabelle 23: Remote Download Quelle*

## **10 On-Board Flash Speicher (U1)**

Durch den Einsatz von Flash Speichern als nichtflüchtiger Codepeicher können Sie die Vorteile der modernen Flash-Technik nutzen. Neben den 256 kByte on-chip Flash des Microcontrollers steht ein zusätzlicher on-board Flash-Baustein (U1) auf dem phyCORE-ST10F168/ST10F269 zur Verfügung. Es können folgende Flashtypen bestückt werden:

- 29F200 mit 1\* 16 kByte, 2\* 8 kByte, 1\* 32 kByte, 3\* 64 kByte
- 29F400 mit 1\* 16 kByte, 2\* 8 kByte, 1\* 32 kByte, 7\* 64 kByte
- 29F800 mit 1\* 16 kByte, 2\* 8 kByte, 1\* 32 kByte, 15\* 64 kByte
- 29F160 mit 1\* 16 kByte, 2\* 8 kByte, 1\* 32 kByte, 31\* 64 kByte

Die Verwendung von Flash Speicher erlaubt die Realisierung einer on-board Programmierung des Moduls. Die Flash Speicher sind mit 5 VDC programmierbar, wodurch keine besondere Programmierspannung benötigt wird.

Der Einsatz der Flash-Bausteine als einziger Code-Speicher des Moduls bewirkt, daß das Flash nicht oder nur sehr bedingt zur nichtflüchtigen Ablage von Daten geeignet ist. Dies ist durch die interne Architektur der Flash-Bausteine verursacht, da während des Flash-internen Programmierprozesses ein Lesen von Daten aus dem Baustein unmöglich ist. Demzufolge muß für eine Flash-Programmierung die Programmausführung aus dem Flash heraus verlagert werden (z.B. in von Neumann-RAM), was in der Regel einem einschneidenden Eingriff in den "normalen" Programmablauf gleichkommt.

Nach Stand der Technik zur Drucklegung dieses Manuals weisen die Flash-Bausteine eine Lebenserwartung von min. 100.000 Löschen-/Programmierzyklen auf.



## **11 Batteriepufferung**

Die zur Batteriepufferung nötige Batterie ist für die Grundfunktion des phyCORE-ST10F168/ST10F269 nicht zwingend erforderlich. Allerdings bietet sich die Batteriepufferung als eine günstige und einfache Möglichkeit des nichtflüchtigen Abspeicherns von Daten an und sie ist notwendig für den Datenerhalt der Real-Time Clock.

Der VBAT-Eingang (X1C6) am phyCORE-Connector ist für den Anschluß einer externen Batterie vorgesehen. Der Minuspol der Batterie ist mit der Schaltungsmasse GND des phyCORE-ST10F168/ST10F269 zu verbinden. Wir empfehlen nach dem Stand der Technik zur Drucklegung dieses Manuals Lithium-Batterien, da diese hohe Kapazitäten bei sehr geringer Selbstentladung aufweisen. Der RAM-Baustein und die Echtzeituhr werden bei fehlender Versorgungsspannung VCC von einer eventuell vorhandenen Batterie über VBAT gespeist.

Dazu ist der SRAM und die Echtzeituhr standardmäßig mit VPD verbunden. Bei anliegender Versorgungsspannung VCC wird dann VPD von VCC abgeleitet. Diese Aufgabe übernimmt der Voltage Supervisor Chip (U13). Sinkt die Spannung VCC ab, so schaltet der Voltage Supervisor Chip die Spannung der Stützbatterie nach VPD um, falls eine Stützbatterie vorhanden ist. Damit wird eine Pufferung der SRAMs und der Echtzeituhr bei plötzlichem Spannungsausfall erreicht. Hierfür sind allerdings nur die standardmäßig installierten SRAM-Bausteine geeignet, die einen Stromverbrauch im Power-Down Mode von ca. 1µA haben. Bei Einsatz anderer Speicherbausteine muß sichergestellt sein, daß der Stromverbrauch im Power-Down Mode nicht wesentlich höher ist (insbesondere sehr schnelle SRAMs weisen einen sehr hohen Stromverbrauch auf), da die Stützbatterie sonst sehr schnell entladen wird.

Die Stromaufnahme hängt von den verwendeten Bausteinen bzw. dem Speicherausbau ab (*siehe Kapitel 12, "Technische Daten"*).

**Hinweis:**

Aus Gründen der Betriebssicherheit möchten wir jedoch darauf hinweisen, daß trotz Batteriepufferung eine Veränderung der Dateninhalte im RAM infolge äußerer Störeinflüsse nicht absolut ausgeschlossen werden kann.

## 12 Technische Daten

Das phyCORE-ST10F168/ST10F269 ist in seinen mechanischen Abmessungen in *Bild 7* dargestellt. Die Höhe des Moduls beträgt ohne phyCORE-Connector ca. 6 mm. Hierbei tragen die Bauteile jeweils ca. 2,0 mm auf der Platinenunterseite sowie ca. 2,5 mm auf der Oberseite auf. Die Platine selbst ist ca. 1,5 mm stark.

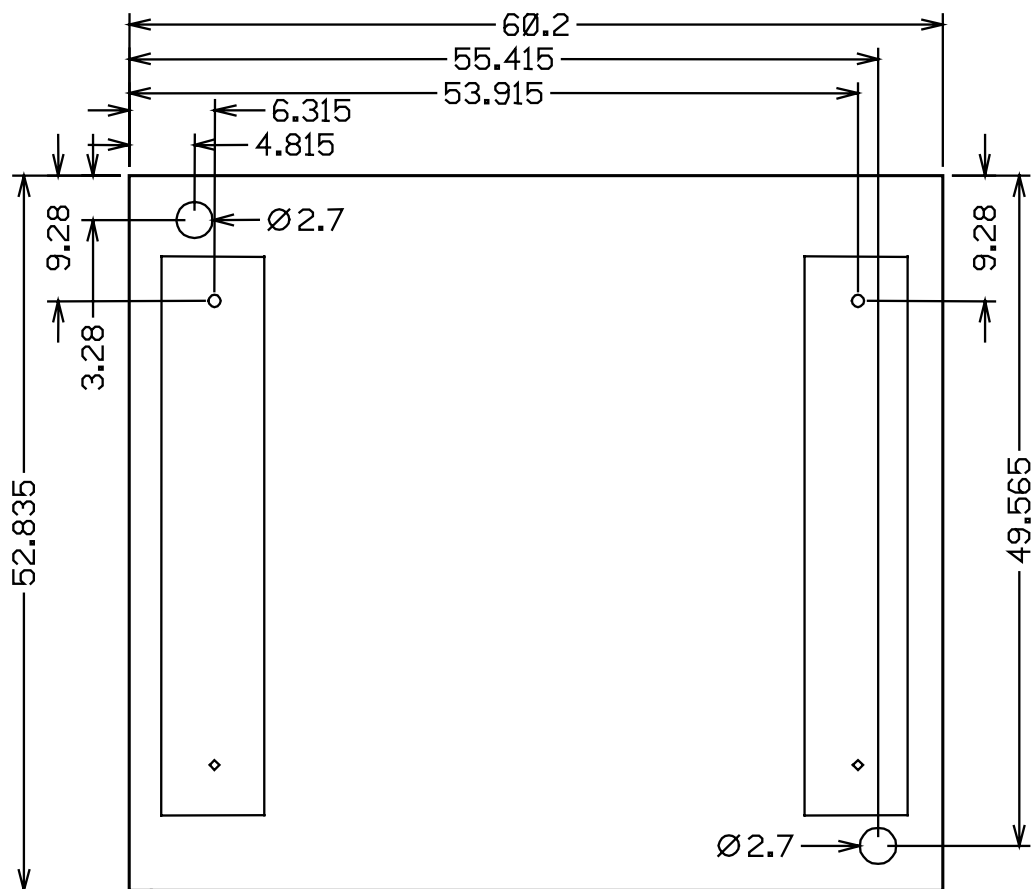


Bild 7: Mechanische Abmaße

**Technische Daten:**

- Modulgröße: 60 mm x 53 mm
- Gewicht: ca. 25 g bei Maximalausbau
- Lagertemperaturbereich: -40°C bis +90°C
- Betriebstemperaturbereich : Standard: 0°C bis +70°C  
Erweitert: -40°C bis +85°C
- Luftfeuchtebereich: max. 95 % r.F. nicht kondensierend
- Betriebsspannungen: 5 V 5 %, VBAT 3 V 20 %
- Stromaufnahme: Bedingung:  
max. 220 mA VCC = 5 V, VBAT = 0 V,  
typ. 110 mA 256 kByte RAM, 5 MHz Quarz,  
20°C  
max. 100 µA VCC = 0 V, VBAT = 3 V, 20°C  
typ. 2 µA (RAM)  
typ. 1 µA Real-Time Clock

Diese Daten beziehen sich auf die Standardkonfiguration des phyCORE-ST10F168/ST10F269 bei Drucklegung.

Beachten Sie bitte, daß die Lagertemperatur bei der Verwendung der Batteriepufferung für die RAMs nur 0°C bis + 70°C beträgt.

### **13 Hinweise zum Umgang mit dem phyCORE-ST10F168/ST10F269**

Als Controller können alle ST10F168 kompatiblen Controller eingesetzt werden (ST10F168, ST10F269).

Bitte beachten Sie, daß beim Einsatz der controllereigenen CAN-Schnittstelle(n) auf Port 4 beim ST10F168/ST10F269 nur 20 externe Adressleitungen (A0...A19) und damit nur 1 MByte Adressraum zur Verfügung stehen.

Um die Adressleitungen A18...A23 (für >256 kByte Flash) zu aktivieren, müssen Konfigurationswiderstände an den Datenleitungen D12 bzw. D11 des Moduls gegen GND geschaltet werden (*siehe Kapitel 4, "System-Konfiguration"*).

Der Adress- und Datenbus sind am Modul ungepuffert herausgeführt. Beim Anschluß externer Bausteine an den Daten-/Adreßbus sowie die Steuerleitungen (/RD, /WR) sollten Sie diese Signale zwischen Modul und Peripheriebausteinen mit externen Puffern versehen (z.B. 74AHCT245).

Der Datenbus D0..15 (Port 0) sollte mit 100 k $\Omega$  Pullup-Widerständen gegen VCC abgeschlossen werden. Außerdem sollten Sie die Möglichkeit vorsehen, direkt an dem Port 0 (Pin 0...15) Konfigurationswiderstände gegen GND zu schalten. Das ermöglicht es Ihnen, den ST10F168/ST10F269 in verschiedene Konfigurationen zu starten, da diese Pins während RESET abgefragt werden (*siehe Controller Handbuch und Kapitel 4, "System-Konfiguration"*).

Der /NMI-Eingang ist mit einem Pullup-Widerstand (10 k $\Omega$ ) gegen VCC gelegt. So kann mit einer high-low Flanke z.B. durch einen Taster gegen GND der NMI ausgelöst werden, was auch bei der Softwareentwicklung mit einem Monitor sinnvoll ist (*siehe Monitor Handbuch*).

Von einem Wechsel einzelner Komponenten wie dem Microcontroller, dem Quarz oder Anderer ist aufgrund der hohen Packungsdichte des Moduls generell abzuraten. Sollte dies wider Erwarten vonnöten sein, so ist zu beachten, daß beim Auslöten die Leiterplatte, sowie umliegende Bauteile oder Sockel, nicht beschädigt werden. Die Löt pads können sich bei Überhitzung von der Platine ablösen, wodurch das Modul unbrauchbar wird. Erhitzen Sie vorsichtig paarweise die benachbarten Anschlüsse; nach einigen Wechseln können Sie das Bauteil mit der Lötspitze abheben. Alternativ kann ein entsprechendes Heißluft-Werkzeug zum Erhitzen der Lötstellen verwendet werden.

## **14 Das phyCORE-ST10F168/ST10F269 auf dem Development Board phyCORE-HD200**

Um die schnelle und problemlose Inbetriebnahme unserer Microcontroller-Module unter üblichen Laborbedingungen zu vereinfachen, bieten wir als Ergänzung ein passendes Development Board an. Dieses stellt standardisierte Funktionen und Anschlüsse für die Spannungsversorgung und die einfache Kommunikation mit dem Modul zur Verfügung.

### **14.1 Das Konzept des Development Board phyCORE-HD200**

Das Development Board phyCORE-HD200 ist Kernstück einer flexiblen Inbetriebnahme- und Entwicklungsplattform, die einfach und rasch an die Erfordernisse spezieller Applikationen oder neuer Microcontroller-Module angepaßt werden kann. Als Ergänzung bieten wir zum Aufbau dieser Plattformen Erweiterungsplatinen mit unterschiedlichen Funktionen an.

Dieser "Mehr-Platinen-Ansatz" ist in *Bild 8* dargestellt:

- Das eigentliche **Development Board** (1) stellt, als das Kernstück, die absolut rudimentären Funktionen und Anschlüsse zur Verfügung. Dazu zählen die Anschlüsse für ein **externes Netzteil** (2) sowie für die **seriellen Schnittstellen** (3) des Microcontroller-Moduls (je nach Modul bis zu zwei RS-232 und bis zu zwei RS-485 oder CAN-Schnittstellen).
- Auf dem Development Board werden alle Signale des aufgesteckten Moduls über eine starre 1:1 Zuordnung auf zwei Platinen-Steckverbinder geführt. Durch diese starre Zuordnung hängt die konkrete Belegung des **Expansion-Bus** (4) ausschließlich von der Belegung des eingesetzten Microcontroller-Moduls ab.

- Da die mechanische Ausführung des Erweiterungsbusses bei allen entsprechenden Development Boards unseres Hauses standardisiert ist, können wir eine Auswahl von allgemein passenden **Erweiterungsplatinen** (5) anbieten. Diese unterstützen, in unterschiedlichem Umfang, **zusätzliche I/O-Funktionen** (6) zur gezielten Demonstration von bestimmten Controller-Features des eingesetzten **Microcontroller-Moduls** (9).
- Ein **Patchfeld** (7) auf jeder Erweiterungsplatine erlaubt die flexible Verbindung zwischen den jeweils angebotenen Funktionsgruppen und den entsprechenden Signalen des verwendeten Microcontroller-Moduls auf dem Development Board. Hierzu sind wiederum alle Signale des Erweiterungsbusses in einer starren 1:1 Zuordnung auf Stiftleisten geführt. Durch Aufstecken der beigefügten **Patchkabel** (8) auf die korrespondierenden Pins der Stiftleisten werden die erforderlichen Verbindungen hergestellt.

Folgende Skizze verdeutlicht die Zusammenhänge:

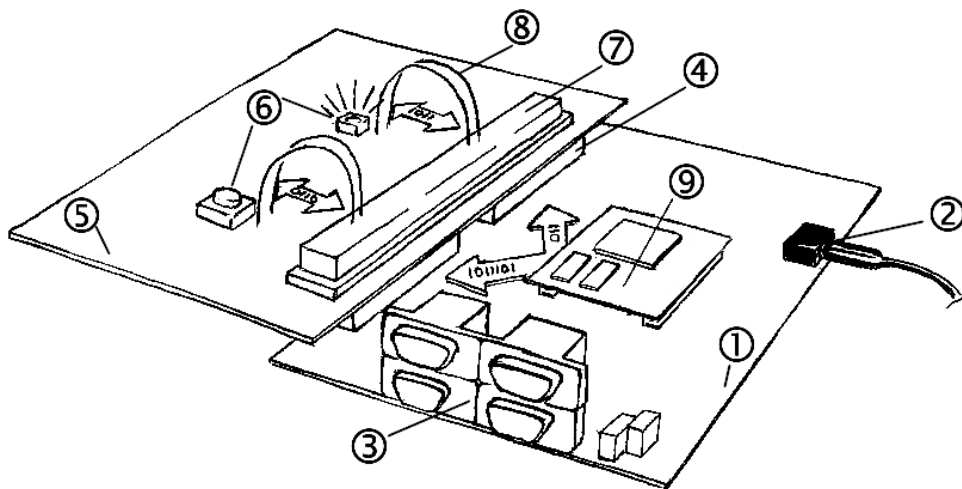


Bild 8: Das Mehrplatinenkonzept mit phyCORE-ST10F168/ST10F269, Development Board und Erweiterungsplatine

Die folgenden Kapitel enthalten spezifische Informationen, die für den Betrieb des phyCORE-ST10F168/ST10F269 auf dem Development Board phyCORE-HD200 relevant sind.

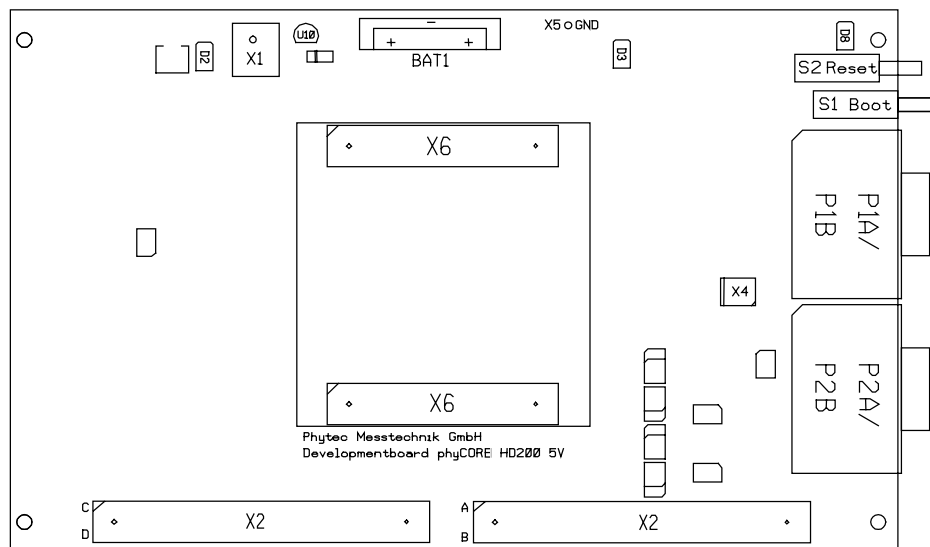


## 14.2 Anschlüsse und Jumper des Development Board phyCORE-HD200

### 14.2.1 Anschlüsse

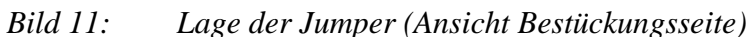
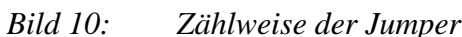
Wie in *Bild 9* dargestellt, stehen folgende Anschlüsse zur Verfügung:

- X1 - Kleinspannungsbuchse zum Anschluß der Versorgungsspannung
- X2 - Expansion-Connector zum Anschluß einer Erweiterungsplatine
- P1 – DB-9 Buchsen zum Anschluß serieller Schnittstellen nach RS-232 Standard
- P2 – DB-9 Stecker zum Anschluß von CAN Feldbussen
- X4 - Spannungsabgriff für die Versorgung externer Baugruppen
- X5 - GND-Anschluß für Meßzwecke
- X6 – phyCORE-Connector zur Aufnahme des phyCORE-Moduls
- U9 od. U10 - Anschluß für optionalen Nummernchip
- BAT1- Anschluß für eine optionale Pufferbatterie



*Bild 9: Lage der Anschlüsse auf dem Development Board phyCORE-HD200*

Es sei ausdrücklich darauf hingewiesen, daß bei allen Modulan-  
schlüssen unbedingt die Maximalspannungen und -ströme nicht  
überschritten werden dürfen. Die Grenzwerte hierfür können Sie dem  
jeweiligen Microcontroller-Handbuch und den entsprechenden  
Datenblättern der eingesetzten Schaltkreise auf dem Development  
Board entnehmen. Da eventuell auftretende Störungen stark vom  
Einsatzgebiet bzw. Anwendungsfall abhängen, obliegt es der  
Verantwortung des Anwenders, in entsprechend kritischer Umgebung  
geeignete Schutzmaßnahmen zu treffen.



Das folgende Bild zeigt die Default-Einstellungen für die Jumper auf dem Development Board phyCORE-HD200.

Die Default-Einstellungen konfigurieren das Development Board phyCORE-HD200 nur für Funktionen eines Standard phyCORE-ST10F168/ST10F269 (Standard = ST10F168 Controller; UART nicht bestückt), d.h. z.B. Nutzung der 1. seriellen Schnittstelle, der LED D3, des Boot-Tasters etc.). Weiterführende Funktionen (Nutzung eines optionalen UART für die 2. serielle Schnittstelle, Nutzung der 2. CAN-Schnittstelle etc.) müssen gemäß der Tabellen ab *Kapitel 14.3* konfiguriert werden.

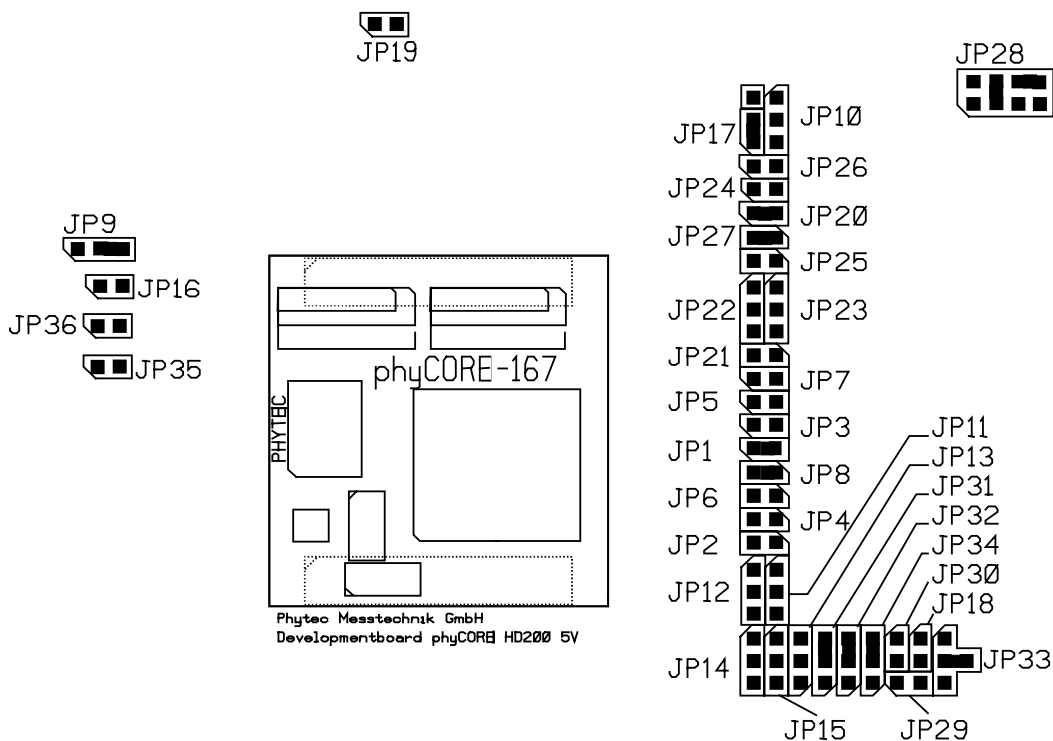


Bild 12: Default Jumper-Stellungen des Development Board HD200 für Standard phyCORE-ST10F168/ST10F269 (nicht maßstabsgetreu)

### 14.2.3 Nicht unterstützte Features und unzulässige Jumper-Stellungen

Die folgende Tabelle weist Jumper-Stellungen aus, die aufgrund von speziellen Produkteigenschaften des phyCORE-ST10F168/ST10F269 auf dem Development Board phyCORE-HD200 unzulässig sind.

#### Spannungsversorgung:

Das Development Board phyCORE-HD200 unterstützt zwei Hauptspannungsversorgungen für den Betrieb verschiedener phyCORE-Module. Beim Einsatz des phyCORE-ST10F168/ST10F269 wird nur eine Hauptspannungsversorgung VCC mit 5 V benötigt. Die Anschlüsse für eine zweite Spannungsversorgung VCC2 sind am phyCORE-ST10F168/ST10F269 nicht definiert bzw. kontaktiert und dürfen deshalb nicht benutzt werden.

Jumper	Stellung	Wirkung
JP16	geschlossen	VCC2 an offene Pins X1C4 und X1C5 des phyCORE-ST10F168/ST10F269

Tabelle 24: Unzulässige Jumper-Stellung JP16 auf dem Development Board

#### Keine RS-485 Schnittstelle:

Der DB-9 Stecker P2B des Development Boardes kann alternativ zu CAN auch als RS-485 Schnittstelle verwendet werden. Da das phyCORE-ST10F168/ST10F269 keine RS-485 Schnittstelle besitzt, sind die entsprechenden Jumperkonfigurationen auf dem Development Board unzulässig.

Jumper	Stellung	Wirkung
JP30	geschlossen	TxD-Signal der 2. seriellen Schnittstelle an Pin 8 des DB-9 Steckers P2B
JP33	1 + 2	RxD-Signal der 2. seriellen Schnittstelle an Pin 2 des DB-9 Steckers P2B

Tabelle 25: Unzulässige Jumper-Stellung JP30/JP33 auf dem Development Board

### Referenzspannungsquelle für A/D-Wandler

Die Pins X1C42, X1C47, X1D39, X1D44 und X1D49 (VAGND) des phyCORE-ST10F168/ST10F269 sind auf dem Development Board phyCORE-HD200 direkt mit GND verbunden. Deshalb ist eine freie Definition von VAGND nicht möglich. Der Jumper J5 auf dem phyCORE-ST10F168/ST10F269 ist ohne Funktion.

## 14.3 Funktionsgruppen des Development Board

Dieser Abschnitt beschreibt im Detail die vom phyCORE-ST10F168/ST10F269 unterstützen Funktionsgruppen des Development Board phyCORE-HD200 sowie entsprechende Jumper-Stellungen. In Abhängigkeit von bestimmten Bestückungsoptionen des eingesetzten phyCORE-ST10F168/ST10F269 können Konfigurationen eingestellt werden, die unter Umständen verschieden von den in *Bild 11* aufgezeigten Default Jumper-Stellungen sind. Durch die Änderung der Default-Einstellungen auf eine abweichende Konfiguration können alternative oder zusätzliche Funktionen des Development Board phyCORE-HD200 aktiviert werden.

### 14.3.1 Spannungsversorgung an X1

#### **Achtung!**

Verwenden Sie keine Labornetzteile! Die Einschaltspitzen könnten das eingesetzte Modul zerstören!

Vermeiden Sie außerdem bei anliegender Spannung, das Modul bzw. die Jumperbelegung zu wechseln!

Zulässiger Spannungsbereich : +5 VDC geregelt.

Die erforderliche Strombelastbarkeit ist von den benutzten Optionen des phyCORE-ST10F168/ST10F269 sowie von den verwendeten Zusatzplatinen abhängig. Wir empfehlen die Verwendung von Netzteilen mit mindestens 500 mA Strombelastbarkeit.

Jumper	Stellung	Wirkung
JP9	2 + 3	5 V als Hauptversorgungsspannung des phyCORE-ST10F168/ST10F269

Tabelle 26: JP9 Konfiguration der Hauptversorgungsspannung VCC

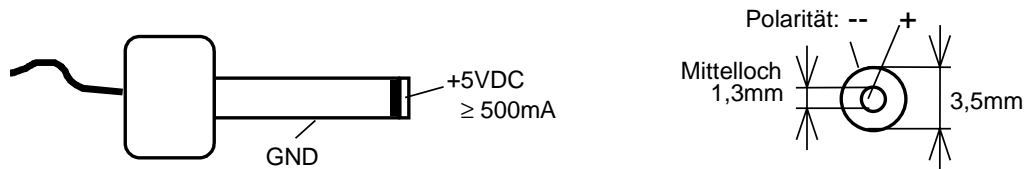


Bild 13: Anschluß der Versorgungsspannung an X1

### Achtung!

Für diese Funktion sind die folgenden Jumper-Stellungen unzulässig:

Jumper	Stellung	Wirkung
JP9	1 + 2	3,3 V als Hauptversorgungsspannung des phyCORE-ST10F168/ST10F269
	offen	phyCORE-ST10F168/ST10F269 wird nicht mit einer Hauptversorgungsspannung versorgt

Tabelle 27: JP9 Unzulässige Jumper-Stellungen der Hauptspannungsversorgung

Bei Konfiguration von Jumper JP9 auf Position 1+2 wird eine Hauptversorgungsspannung von 3,3 V eingestellt, was zur Zerstörung des phyCORE-ST10F168/ST10F269 führen kann. Bei geöffnetem Jumper JP9 liegt keine Versorgungsspannung am phyCORE-ST10F168/ST10F269 an. Diese Jumper-Stellung ist deshalb ebenfalls unzulässig.

### 14.3.2 Aktivierung des Bootstrap Loaders

Der ST10F168 bzw. der ST10F269 Microcontroller verfügt über einen on-chip Bootstrap Loader der von verschiedenen Utility-Programmen (z.B. FlashTools, Monitor166) verwendet wird. Mit Hilfe dieses Bootstrap Loaders und der auf einem PC installierten korrespondierenden Software ist z.B. die on-board Programmierung des Flash Speichers oder das Debuggen des Anwenderprogrammes über eine RS-232 Verbindung möglich.

Zum Starten des on-chip Bootstrap Loaders auf dem phyCORE-ST10F168/ST10F269 muß an der Datenleitung D4 des Microcontrollers zum Zeitpunkt des Wechsels des RESET Signals vom aktiven in den inaktiven Zustand ein Low-Pegel anliegen. Dies erfordert das Anliegen eines High-Pegels an Pin X1C9 des phyCORE-ST10F168/ST10F269 (Boot-Eingang ist high-aktiv).

Unter Verwendung der auf dem Development Board phyCORE-HD200 befindlichen Funktionseinheiten kann dies auf drei verschiedenen Wegen erreicht werden:

1. Mit Jumper JP28 kann der BOOT-Taster (S1) mit VCC verbunden werden. Damit wird bei Drücken des BOOT-Tasters während des Reset-Vorgangs oder dem Einschalten der Versorgungsspannung der Bootstrap Loader gestartet.

Jumper	Stellung	Wirkung
JP28	6 + 8 und 3 + 4	Boot-Taster (in Verbindung mit RESET oder Einschalten der Versorgungsspannung) aktiviert Bootmodus des ST10F168/ST10F269

Tabelle 28: JP28 Konfiguration des BOOT-Tasters



- Das Boot-Pin des phyCORE-ST10F168/ST10F269 wird über einem Pull-up Widerstand mit VCC verbunden. Dadurch wird über eine modulinterne Schaltung die Datenleitung D4 auf Masse gezogen, welches dann den Bootloader startet. Dies erspart das zusätzliche Drücken des BOOT-Tasters während des Auslösens von RESET oder dem Einschalten der Versorgungsspannung.

**Achtung!**

Bei dieser Konfiguration ist keine Ausführung eines normalen Reset und damit der Start Ihrer Applikation möglich. Es wird immer der Bootstrap Loader gestartet

Jumper	Stellung	Wirkung
JP28	4 + 6	Boot-Pin über Pull-up Widerstand an VCC angeschlossen. Bootstrap Loader wird bei Reset oder Einschalten der Versorgungsspannung immer gestartet

Tabelle 29: JP28 Konfiguration für permanenten Start des Bootstrap Loaders

- Der Bootstrap Loader kann auch über die DB-9 Buchse P1A gestartet werden. Dazu muß der Zustandswechsel des RESET-Signals über Pin 7 gesteuert werden, während an Pin 4 ein statischer High-Pegel für das BOOT-Signal anliegt.

Jumper	Stellung	Wirkung
JP22	2 + 3	Pin 7 (CTS) der DB-9 Buchse P1A als RESET-Signal für das phyCORE-ST10F168/ST10F269
JP23	2 + 3	Pin 4 (DSR) der DB-9 Buchse P1A als BOOT-Signal für das phyCORE-ST10F168/ST10F269
JP10	2 + 3	High-Pegel an Pin 4 der DB-9 Buchse P1A (DSR von Host-PC) startet Bootstrap Loader

Tabelle 30: JP22, JP23, JP10 Konfiguration von BOOT über RS-232

**Achtung!**

Für diese Funktion ist die folgende Jumper-Stellung unzulässig:

Jumper	Stellung	Wirkung
JP10	1 + 2	Jumper-Einstellung erzeugt Low-Pegel am Boot-Eingang des phyCORE-ST10F168/ST10F269

Tabelle 31: Unzulässige Jumper-Stellung beim BOOT über RS-232

### 14.3.3 Erste serielle Schnittstelle an Buchse P1A

Der Anschluß P1A ist die untere Buchse der Doppelbuchse P1. P1A ist über Jumper mit der ersten seriellen Schnittstelle des phyCORE-ST10F168/ST10F269 verbunden. In Verbindung mit einem Host-System kann das phyCORE-ST10F168/ST10F269 über die Buchse P1A in den Bootstrap-Modus gebracht werden (siehe Kapitel 14.3.2).

Jumper	Stellung	Wirkung
JP20	geschlossen <sup>1</sup>	Pin 2 der DB9-Buchse P1A mit TxD0 der ersten RS-232 Schnittstelle verbunden
JP21	offen	Pin 9 der DB9-Buchse P1A nicht belegt
JP22	offen	Pin 7 der DB9-Buchse P1A nicht belegt
	2 + 3 <sup>2</sup>	RESET-Eingang des Moduls kann durch Signalpegel auf der RTS Leitung vom Host gesteuert werden
JP23	offen	Pin 4 der DB9-Buchse P1A nicht belegt
	2 + 3 <sup>2</sup>	BOOT-Eingang des Moduls kann durch Signalpegel auf der DTR Leitung vom Host gesteuert werden ( <b>Achtung!</b> JP 10 auf 2 + 3 setzen).
JP24	offen	Pin 6 der DB9-Buchse P1A nicht belegt
JP25	offen	Pin 8 der DB9-Buchse P1A nicht belegt
JP26	offen	Pin 1 der DB9-Buchse P1A nicht belegt
JP27	geschlossen <sup>1</sup>	Pin 3 der DB9-Buchse P1A mit RxD0 der ersten RS-232 Schnittstelle verbunden

Tabelle 32: Jumper-Konfiguration der DB-9 Buchse P1A (1. RS-232)

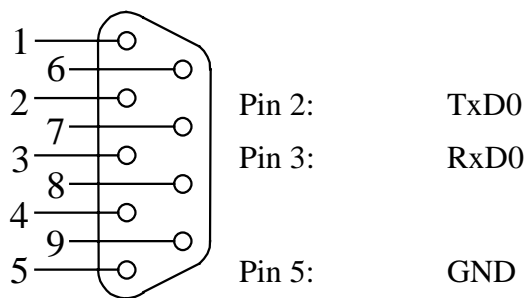


Bild 14: Belegung der DB-9 Buchse P1A als erste RS-232 (Ansicht Vorderseite)

- <sup>1</sup>: Dieser Jumper sollte immer gesetzt sein, da standardmäßig die FlashTools über die erste serielle Schnittstelle mit dem phyCORE-Modul kommunizieren.
- <sup>2</sup>: Alternative Jumperkonfiguration für zusätzliche Funktion (siehe Kapitel 14.3.2). Nicht erforderlich für einfache Kommunikation.

**Achtung!**

In diesem Anwendungsfall sind die folgenden Jumper-Stellungen unzulässig:

<b>Jumper</b>	<b>Stellung</b>	<b>Wirkung</b>
JP20	offen	Pin 2 der DB9-Buchse unbelegt, keine Verbindung zu TxD0 der ersten seriellen Schnittstelle
JP21	geschlossen	Pin 9 der DB-9 Buchse P1A ist mit P8.2 des phyCORE-ST10F168/ST10F269 verbunden
JP22	1 + 2	Pin 7 der DB-9 Buchse P1A ist mit P2.15 des phyCORE-ST10F168/ST10F269 verbunden
JP23	1 + 2	Pin 4 der DB-9 Buchse P1A ist mit P8.0 des phyCORE-ST10F168/ST10F269 verbunden
JP24	geschlossen	Pin 6 der DB-9 Buchse P1A ist mit P8.1 des phyCORE-ST10F168/ST10F269 verbunden
JP25	geschlossen	Pin 8 der DB-9 Buchse P1A ist mit P2.14 des phyCORE-ST10F168/ST10F269 verbunden
JP26	geschlossen	Pin 1 der DB-9 Buchse P1A ist mit P8.3 des phyCORE-ST10F168/ST10F269 verbunden
JP27	offen	Pin 3 der DB9-Buchse unbelegt, keine Verbindung zu RxD0 der ersten seriellen Schnittstelle

*Tabelle 33: Unzulässige Jumper-Stellungen bei der Konfiguration von P1A als erste RS-232 Schnittstelle*

Der Spannungspegel auf den RS-232 Leitungen kann zur Zerstörung des phyCORE-ST10F168/ST10F269 führen.

#### 14.3.4 Zweite serielle Schnittstelle an Buchse P1B

Der Anschluß P1B ist die obere Buchse der Doppelbuchse P1. P1B ist über Jumper mit der zweiten seriellen Schnittstelle des phyCORE-ST10F168/ST10F269 verbunden. Abhängig von der Konfiguration des Moduls (*siehe Kapitel 3.10*) und der Bestückungsoption eines UARTs ergeben sich folgende Möglichkeiten.

1. **UART** (U7) auf dem phyCORE-ST10F168/ST10F269 ist **NICHT bestückt** (Standard Auslieferungszustand) und keine Emulation der seriellen Schnittstelle mit Ports P3.0 und P3.1.

Jumper	Stellung	Wirkung
JP1	offen	Pin 2 der DB-9 Buchse P1B nicht belegt
JP2	offen	Pin 9 der DB-9 Buchse P1B nicht belegt
JP3	offen	Pin 7 der DB-9 Buchse P1B nicht belegt
JP4	offen	Pin 4 der DB-9 Buchse P1B nicht belegt
JP5	offen	Pin 6 der DB-9 Buchse P1B nicht belegt
JP6	offen	Pin 8 der DB-9 Buchse P1B nicht belegt
JP7	offen	Pin 1 der DB-9 Buchse P1B nicht belegt
JP8	offen	Pin 3 der DB-9 Buchse P1B nicht belegt

Tabelle 34: Jumper-Konfiguration der DB-9 Buchse P1B (ohne 2. RS-232)

Bei dieser Konfiguration steht keine zweite Schnittstelle zur Verfügung.

**Achtung!**

In diesem Anwendungsfall sind die folgenden Jumper-Stellungen unzulässig:

<b>Jumper</b>	<b>Stellung</b>	<b>Erklärung</b>
JP1	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein TxD1_RS232 Signal zur Verfügung (P1B Pin 2)
JP2	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein RI1_TTL Signal zur Verfügung (P1B Pin 9)
JP3	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein CTS1_RS232 Signal zur Verfügung (P1B Pin 7)
JP4	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein DSR1_RS232 Signal zur Verfügung (P1B Pin 4)
JP5	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein DTR1_RS232 Signal zur Verfügung (P1B Pin 6)
JP6	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein RTS1_RS232 Signal zur Verfügung (P1B Pin 8)
JP7	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein CD1_RS232 Signal zur Verfügung (P1B Pin 1)
JP8	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein RxD1_RS232 Signal zur Verfügung (P1B Pin 3)

*Tabelle 35: Unzulässige Jumper-Stellungen bei der Konfiguration von P1B (ohne 2. RS-232)*

Wenn versehentlich ein RS-232 Kabel an P1B angeschlossen ist, dann kann der Spannungspegel auf den RS-232 Leitungen zur Zerstörung des phyCORE-ST10F168/ST10F269 führen.

## 2. UART auf dem phyCORE-ST10F168/ST10F269 ist **bestückt**:

Ist ein UART-Baustein auf dem phyCORE-ST10F168/ST10F269 bestückt, steht an der DB-9 Buchse P1B eine vollständige zweite RS-232 Schnittstelle zur Verfügung.

Jumper	Stellung	Wirkung
JP1	geschlossen	Pin 2 mit TxD1_RS232 des UART U7 verbunden
	offen	Pin 2 der DB9-Buchse P1B nicht belegt
JP2	geschlossen	Pin 9 mit RI1_TTL des UART U7 verbunden
	offen	Pin 9 der DB9-Buchse P1B nicht belegt
JP3	geschlossen	Pin 7 mit CTS1_RS232 des UART U7 verbunden
	offen	Pin 7 der DB9-Buchse P1B nicht belegt
JP4	geschlossen	Pin 4 mit DSR1_RS232 des UART U7 verbunden
	offen	Pin 4 der DB9-Buchse P1B nicht belegt
JP5	geschlossen	Pin 6 mit DTR1_RS232 des UART U7 verbunden
	offen	Pin 6 der DB9-Buchse P1B nicht belegt
JP6	geschlossen	Pin 8 mit RTS1_RS232 des UART U7 verbunden
	offen	Pin 8 der DB9-Buchse P1B nicht belegt
JP7	geschlossen	Pin 1 mit DCD1_RS232 des UART U7 verbunden
	offen	Pin 1 der DB9-Buchse P1B nicht belegt
JP8	geschlossen	Pin 3 mit RxD1_RS232 des UART U7 verbunden
	offen	Pin 3 der DB9-Buchse P1B nicht belegt

Tabelle 36: Jumper-Konfiguration der DB-9 Buchse P1B (UART, 2. RS-232)

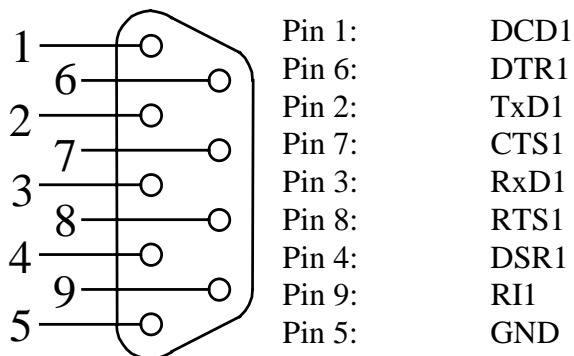


Bild 15: Belegung der DB-9 Buchse P1B als 2. RS-232 (UART bestückt, Ansicht Vorderseite)

3. **UART** (U7) auf dem phyCORE-ST10F168/ST10F269 ist **NICHT bestückt** (Standard Auslieferungszustand), jedoch Emulation<sup>1</sup> der seriellen Schnittstelle mit Ports P3.0 und P3.1.

Jumper	Stellung	Wirkung
JP1	geschlossen	Port 3.0 des ST10Fx als TxD1 emuliert und über RS-232 Treiber U6 auf dem phyCORE-Modul mit Pin 2 der DB9-Buchse P1B verbunden
JP2	offen	Pin 9 der DB-9 Buchse P1B nicht belegt
JP3	offen	Pin 7 der DB-9 Buchse P1B nicht belegt
JP4	offen	Pin 4 der DB-9 Buchse P1B nicht belegt
JP5	offen	Pin 6 der DB-9 Buchse P1B nicht belegt
JP6	offen	Pin 8 der DB-9 Buchse P1B nicht belegt
JP7	offen	Pin 1 der DB-9 Buchse P1B nicht belegt
JP8	geschlossen	Port 3.1 des ST10Fx als RxD1 emuliert und über RS-232 Treiber U6 auf dem phyCORE-Modul mit Pin 3 der DB9-Buchse P1B verbunden

Tabelle 37: Jumper-Konfiguration der DB-9 Buchse P1B (Software-Emulation der 2. RS-232)

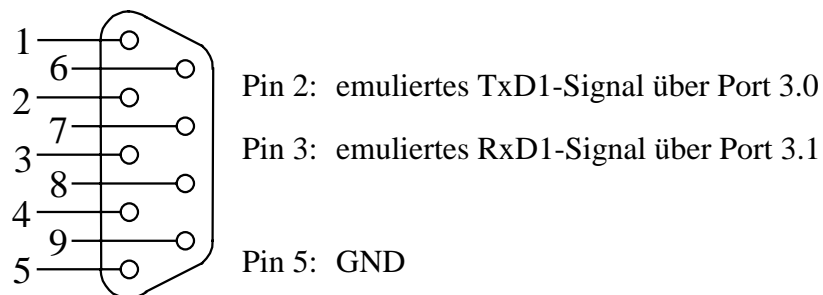


Bild 16: Pinbelegung der DB9-Buchsen P1B bei emulierter RS-232 Schnittstelle (Ansicht Vorderseite)

<sup>1</sup>: Die Emulation der seriellen Schnittstelle erfordert spezielle Software-Treiber, die in der Regel Teil von Entwicklungswerkzeugen (z.B. Debugger, Monitor-Programme etc.) sind.

**Achtung!**

Ist der **UART** (U7) auf dem phyCORE-ST10F168/ST10F269 **NICHT bestückt** (Standard Auslieferungszustand), sind folgende Konfigurationen unzulässig:

Jumper	Stellung	Erklärung
JP2	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein RI1 Signal zur Verfügung (P1B Pin 9)
JP3	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein CTS1 Signal zur Verfügung (P1B Pin 7)
JP4	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein DSR1 Signal zur Verfügung (P1B Pin 4)
JP5	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein DTR1 Signal zur Verfügung (P1B Pin 6)
JP6	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein RTS1 Signal zur Verfügung (P1B Pin 8)
JP7	geschlossen	phyCORE-ST10F168/ST10F269 stellt kein DCD1 Signal zur Verfügung (P1B Pin 1)

*Tabelle 38: Unzulässige Jumper-Stellungen bei der Konfiguration von P1B (Software-Emulation 2. RS-232)*



### 14.3.5 Erste CAN-Schnittstelle an Stecker P2A

Der Anschluß P2A ist der untere Stecker des Doppelsteckers P2. Zu diesem Stecker werden die Signale der Schnittstelle CAN0 des phyCORE-ST10F168/ST10F269 geführt. Aufgrund von verschiedenen Konfigurationen bezüglich der CAN-Treiber und deren Spannungsversorgung ergeben sich die drei nachfolgend dargestellten Möglichkeiten:

1. Die CAN-Treiber auf dem phyCORE-ST10F168/ST10F269 werden verwendet und deren Signale direkt an den Stecker P2A geführt:

Jumper	Stellung	Wirkung
JP31	2 + 3	P2A Pin 2 ist mit CAN_L0 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP32	2 + 3	P2A Pin 7 ist mit CAN_H0 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP11	offen	Eingang am Optokoppler U4 auf dem Development Board phyCORE-HD200 offen
JP12	offen	Ausgang am Optokoppler U5 auf dem Development Board phyCORE-HD200 offen
JP13	offen	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 spannungsfrei
JP18	offen	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 potentialfrei
JP29	offen	kein Spannungsabgriff über CAN-Bus

Tabelle 39: Jumper-Konfiguration des CAN-Steckers P2A mit CAN-Treiber auf phyCORE-ST10F168/ST10F269

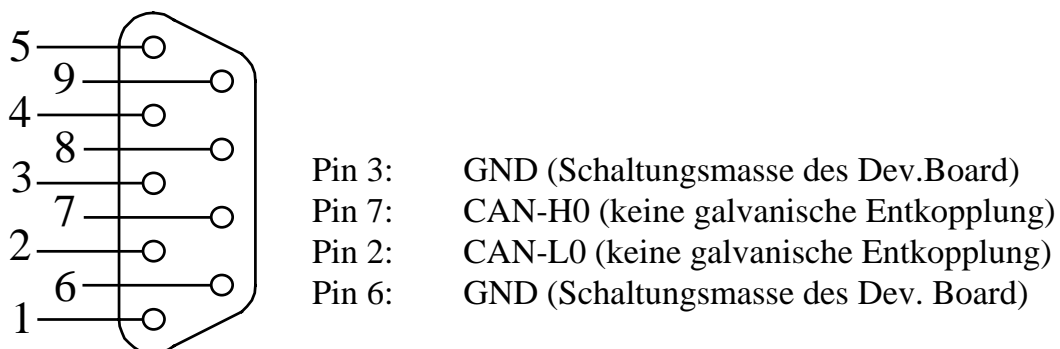


Bild 17: Belegung des DB9-Steckers P2A (CAN-Treiber auf phyCORE-ST10F168/ST10F269)

2. Die CAN-Treiber des phyCORE-ST10F168/ST10F269 sind deaktiviert und der CAN-Treiber U2 des Development Board wird **ohne galvanische Trennung** verwendet.

Jumper	Stellung	Wirkung
JP31	1 + 2	P2A Pin 2 ist mit CAN-L0 des Treibers U2 auf dem Development Board verbunden
JP32	1 + 2	P2A Pin 7 ist mit CAN-H0 des Treibers U2 auf dem Development Board verbunden
JP11	2 + 3	Optokoppler U4 auf dem Development Board mit CAN1_Tx (P4.6 <sup>1</sup> ) des ST10F168/ST10F269 verbunden
	1 + 2	Optokoppler U4 auf dem Development Board mit CAN1_Tx (P8.1 <sup>2</sup> ) des ST10F168/ST10F269 verbunden
JP12	2 + 3	Optokoppler U5 auf dem Development Board mit CAN1_Rx (P4.5 <sup>3</sup> ) des ST10F168/ST10F269 verbunden
	1 + 2	Optokoppler U5 auf dem Development Board mit CAN1_Rx (P8.0 <sup>4</sup> ) des ST10F168/ST10F269 verbunden
JP13	2 + 3	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden lokal versorgt
JP18	geschlossen	CAN-Treiber und Optokoppler auf dem Development Board mit lokalen GND Potential verbunden
JP29	offen	kein Spannungsabgriff über CAN-Bus

Tabelle 40: Jumper-Konfiguration des CAN-Steckers P2A mit CAN-Treiber auf Development Board

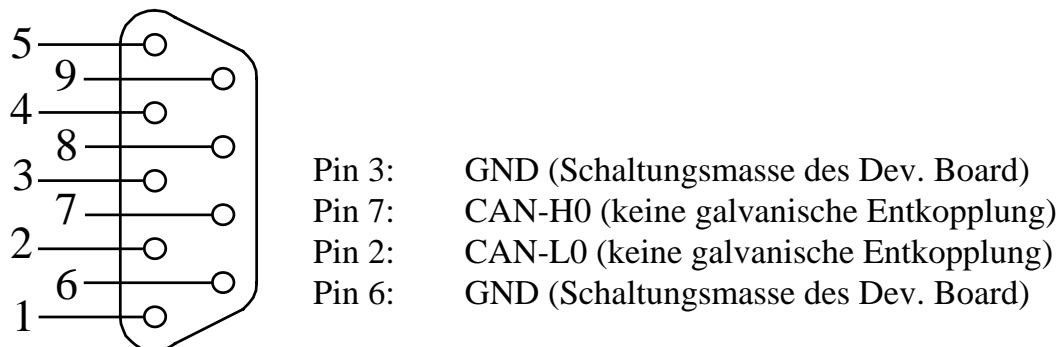


Bild 18: Belegung des DB-9 Steckers P2A (CAN Treiber auf Development Board)

- 1: Port P4.6 ist der default Port für CAN1\_Tx (standard).
- 2: Port P8.1 ist der alternative Port für CAN1\_Tx (siehe Controller-Handbuch).
- 3: Port P4.5 ist der default Port für CAN1\_Rx (standard).
- 4: Port P8.0 ist der alternative Port für CAN1\_Rx (siehe Controller-Handbuch).

**Achtung!**

In diesem Anwendungsfall sind die folgenden Jumper-Stellungen unzulässig:

<b>Jumper</b>	<b>Stellung</b>	<b>Wirkung</b>
JP31	2 + 3	P2A Pin 2 ist mit CAN-L0 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP32	2 + 3	P2A Pin 7 ist mit CAN-H0 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP11	offen	Eingang am Optokoppler U4 auf dem Development Board offen
JP12	offen	Ausgang am Optokoppler U5 auf dem Development Board offen
JP13	1 + 2	CAN Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden von einer externen Spannung über einen separaten Regler versorgt
JP29	geschlossen	Spannungszuführung für separaten Regler über Pin 9 des DB-9 Steckers P2A

*Tabelle 41: Unzulässige Jumper-Stellungen CAN-Stecker P2A (CAN-Treiber auf Development Board)*

3. Die CAN-Treiber des phyCORE-ST10F168/ST10F269 sind deaktiviert und der CAN-Treiber U2 des Development Boardes wird **mit galvanischer Trennung** verwendet. In diesem Fall ist die externe Zuführung einer CAN-Versorgungsspannung im Bereich von 7 – 13 V notwendig. Bitte beachten Sie, dass Sie die externe Spannung nur über einen der beiden Stecker P2A **oder** P2B zuführen.

Jumper	Stellung	Wirkung
JP31	1 + 2	P2A Pin 2 ist mit CAN-L0 des Treibers U2 auf dem Development Board verbunden
JP32	1 + 2	P2A Pin 7 ist mit CAN-H0 des Treibers U2 auf dem Development Board verbunden
JP11	2 + 3	Optokoppler U4 auf dem Development Board mit CAN1_Tx (P4.6 <sup>1</sup> ) des ST10F168/ST10F269 verbunden
	1 + 2	Optokoppler U4 auf dem Development Board mit CAN1_Tx (P8.1 <sup>2</sup> ) des ST10F168/ST10F269 verbunden
JP12	2 + 3	Optokoppler U5 auf dem Development Board mit CAN1_Rx (P4.5 <sup>3</sup> ) des ST10F168/ST10F269 verbunden
	1 + 2	Optokoppler U5 auf dem Development Board mit CAN1_Rx (P8.0 <sup>4</sup> ) des ST10F168/ST10F269 verbunden
JP13	1 + 2	CAN Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden von einer externen Spannung über einen separaten Regler versorgt
JP18	offen	CAN-Treiber und Optokoppler sind vom GND Potential des Development Board getrennt
JP29	geschlossen	Spannungszuführung für separaten Regler über Pin 9 des DB-9 Steckers P2A

Tabelle 42: Jumper-Konfiguration des CAN-Steckers P2A mit CAN-Treiber auf Development Board und galvanischer Trennung

- 
- 1: Port P4.6 ist der default Port für CAN1\_Tx (standard).  
 2: Port P8.1 ist der alternative Port für CAN1\_Tx (siehe Controller-Handbuch).  
 3: Port P4.5 ist der default Port für CAN1\_Rx (standard).  
 4: Port P8.0 ist der alternative Port für CAN1\_Rx (siehe Controller-Handbuch).
-

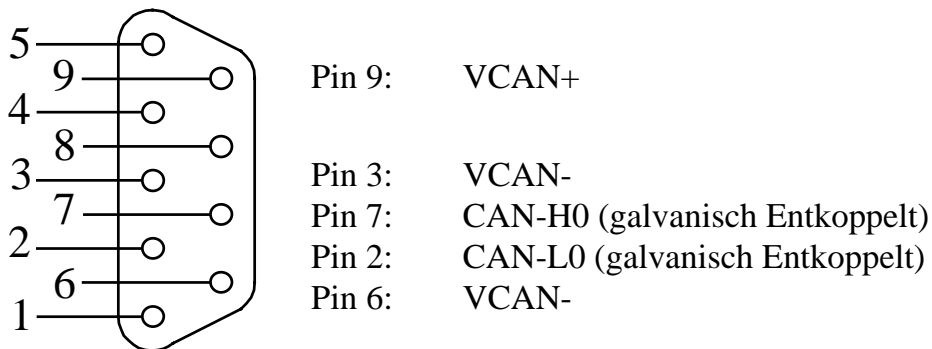


Bild 19: Belegung des DB-9 Steckers P2A (CAN-Treiber auf Development Board und galvanische Trennung)

### Achtung!

In diesem Anwendungsfall sind die folgenden Jumper-Stellungen unzulässig:

Jumper	Stellung	Wirkung
JP31	2 + 3	P2A Pin 2 ist mit CAN-L0 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP32	2 + 3	P2A Pin 7 ist mit CAN-H0 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP11	offen	Eingang am Optokoppler U4 auf dem Development Board offen
JP12	offen	Ausgang am Optokoppler U5 auf dem Development Board offen
JP13	2 + 3	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden lokal versorgt
JP29	offen	kein Spannungsabgriff über CAN-Bus

Tabelle 43: Unzulässige Jumper-Stellungen bei galvanisch getrenntem CAN-Bus (CAN-Treiber auf Development Board)

### 14.3.6 Zweite CAN-Schnittstelle an Stecker P2B

Der Anschluß P2B ist der obere Stecker des Doppelsteckers P2. Zu diesem Stecker werden die Signale der Schnittstelle CAN1 des phyCORE-ST10F269 geführt. Dies setzt eine Bestückung mit dem ST10F269 Controller voraus. Aufgrund von verschiedenen Konfigurationen bezüglich der CAN-Treiber und deren Spannungsversorgung ergeben sich die drei nachfolgend dargestellten Möglichkeiten:

1. Der CAN-Treiber auf dem phyCORE-ST10F269 wird verwendet und dessen Signale direkt an den Stecker P2B geführt:

Jumper	Stellung	Wirkung
JP33	2 + 4	P2B Pin 2 ist mit CAN_L1 vom on-board Treiber des phyCORE-ST10F269 verbunden
JP34	2 + 3	P2B Pin 7 ist mit CAN_H1 vom on-board Treiber des phyCORE-ST10F269 verbunden
JP14	offen	Eingang am Optokoppler U6 auf dem Development Board phyCORE-HD200 offen
JP15	offen	Ausgang am Optokoppler U7 auf dem Development Board phyCORE-HD200 offen
JP13	offen	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 spannungsfrei
JP18	offen	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 potentialfrei
JP29	offen	kein Spannungsabgriff über CAN-Bus

Tabelle 44: Jumper-Konfiguration des CAN-Steckers P2B mit CAN-Treiber auf phyCORE-ST10F269 (nur mit ST10F269)

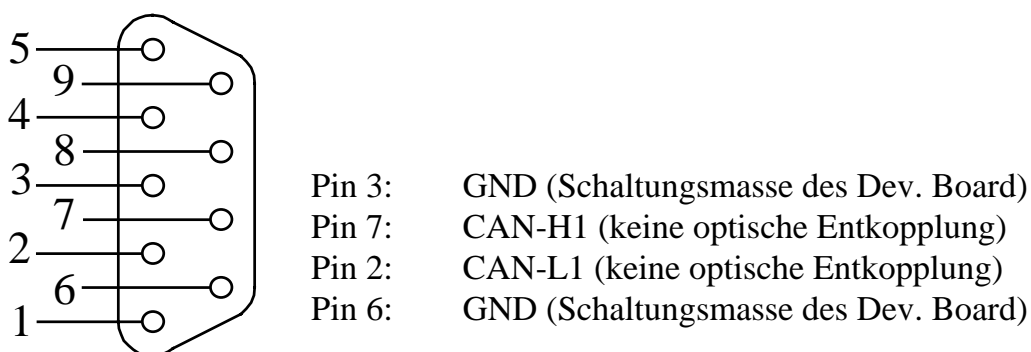


Bild 20: Belegung des DB-9 Steckers P2B (CAN-Treiber auf phyCORE-ST10F269, nur mit ST10F269)

2. Die CAN-Treiber des phyCORE-ST10F269 sind deaktiviert und der CAN-Treiber U3 des Development Board wird **ohne galvanische Trennung** verwendet.

Jumper	Stellung	Wirkung
JP33	2 + 3	P2B Pin 2 ist mit CAN-L1 des Treibers U3 auf dem Development Board verbunden
JP34	1 + 2	P2A Pin 7 ist mit CAN-H1 des Treibers U3 auf dem Development Board verbunden
JP14	2 + 3	Optokoppler U6 auf dem Development Board mit CAN2_Tx (P4.7) des ST10F269 verbunden
JP15	2 + 3	Optokoppler U7 auf dem Development Board mit CAN2_Rx (P4.4) des ST10F269 verbunden
JP13	2 + 3	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden lokal versorgt
JP18	geschlossen	CAN-Treiber und Optokoppler auf dem Development Board mit lokalen GND Potential verbunden
JP29	offen	kein Spannungsabgriff über CAN-Bus

Tabelle 45: Jumper-Konfiguration des CAN-Steckers P2B mit CAN-Treiber auf Development Board (nur mit ST10F269)

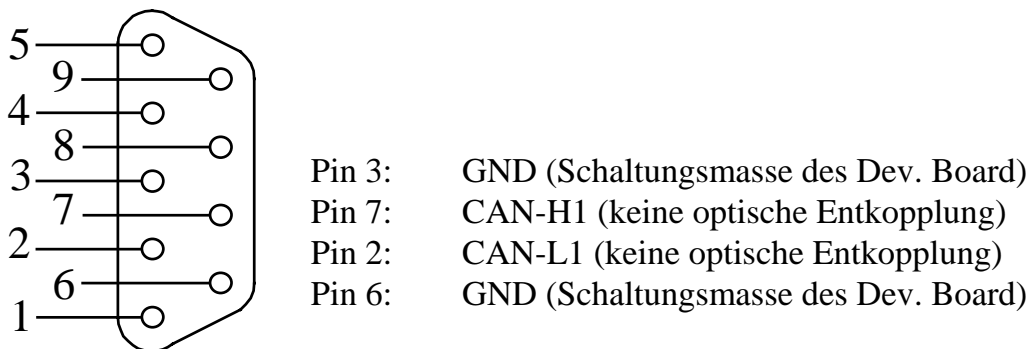


Bild 21: Belegung des DB-9 Steckers P2B (CAN Treiber auf Development Board, nur mit ST10F269)

**Achtung!**

In diesem Anwendungsfall sind die folgenden Jumper-Stellungen unzulässig:

Jumper	Stellung	Wirkung
JP30	geschlossen	P2B Pin 8 ist mit TxD1_RS232 des phyCORE-ST10F269 verbunden
JP33	1 + 2	P2B Pin 2 ist mit P2.5 des phyCORE-ST10F269 verbunden
	2 + 4	P2B Pin 2 ist mit CAN_L1 vom on-board Treiber des phyCORE-ST10F269 verbunden
JP34	2 + 3	P2B Pin 7 ist mit CAN-H1 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP14	offen	Eingang am Optokoppler U6 auf dem Development Board offen
	1 + 2	Optokoppler U6 auf dem Development Board mit P8.3 des ST10F269 verbunden
JP15	offen	Ausgang am Optokoppler U7 auf dem Development Board offen
	1 + 2	Optokoppler U7 auf dem Development Board mit P8.2 des ST10F269 verbunden
JP13	1 + 2	CAN Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden von einer externen Spannung über einen separaten Regler versorgt
JP29	geschlossen	Spannungszuführung für separaten Regler über Pin 9 des DB-9 Steckers P2B

*Tabelle 46: Unzulässige Jumper-Stellungen CAN-Stecker P2B (CAN-Treiber auf Development Board, nur mit ST10F269)*



3. Die CAN-Treiber des phyCORE-ST10F269 sind deaktiviert und der CAN-Treiber U3 des Development Boardes wird **mit galvanischer Trennung** verwendet. In diesem Fall ist die externe Zuführung einer CAN-Versorgungsspannung im Bereich von 7 – 13 V notwendig. Bitte beachten Sie, dass Sie die externe Spannung nur über einen der beiden Stecker P2A **oder** P2B zuführen.

Jumper	Stellung	Wirkung
JP33	2 + 3	P2B Pin 2 ist mit CAN-L1 des Treibers U3 auf dem Development Board verbunden
JP34	1 + 2	P2A Pin 7 ist mit CAN-H1 des Treibers U3 auf dem Development Board verbunden
JP14	2 + 3	Optokoppler U6 auf dem Development Board mit CAN2_Tx (P4.7) des ST10F269 verbunden
JP15	2 + 3	Optokoppler U7 auf dem Development Board mit CAN2_Rx (P4.4) des ST10F269 verbunden
JP13	1 + 2	CAN Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden von einer externen Spannung über einen separaten Regler versorgt
JP18	offen	CAN-Treiber und Optokoppler sind vom GND Potential des Development Board getrennt
JP29	geschlossen	Spannungszuführung für separaten Regler über Pin 9 des DB-9 Steckers P2B

*Tabelle 47: Jumper-Konfiguration des CAN-Steckers P2B mit CAN-Treiber auf Development Board und galvanischer Trennung (nur mit ST10F269)*

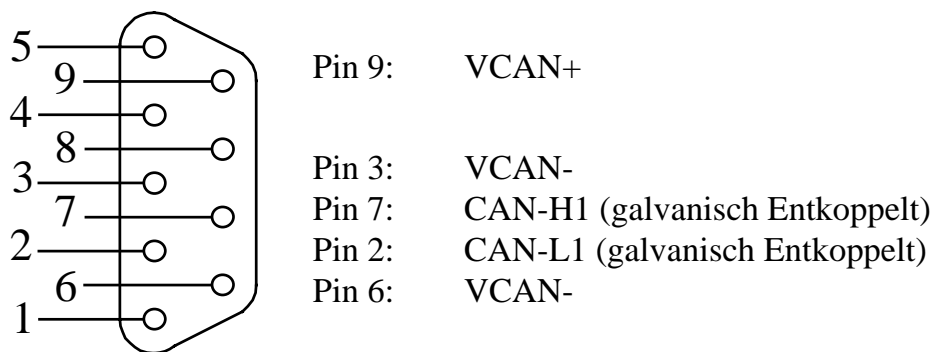


Bild 22: Belegung des DB-9 Steckers P2B (CAN-Treiber auf Development Board und galvanische Trennung, nur mit ST10F269)

### Achtung!

In diesem Anwendungsfall sind die folgenden Jumper-Stellungen unzulässig:

Jumper	Stellung	Wirkung
JP30	geschlossen	P2B Pin 8 ist mit TxD1_RS232 des phyCORE-ST10F269 verbunden
JP33	1 + 2	P2B Pin 2 ist mit P2.5 des phyCORE-ST10F269 verbunden
	2 + 4	P2B Pin 2 ist mit CAN_L1 vom on-board Treiber des phyCORE-ST10F269 verbunden
JP34	2 + 3	P2B Pin 7 ist mit CAN-H1 vom on-board Treiber des phyCORE-ST10F168/ST10F269 verbunden
JP14	offen	Eingang am Optokoppler U6 auf dem Development Board offen
	1 + 2	Optokoppler U6 auf dem Development Board mit P8.3 des ST10F269 verbunden
JP15	offen	Ausgang am Optokoppler U7 auf dem Development Board offen
	1 + 2	Optokoppler U7 auf dem Development Board mit P8.2 des ST10F269 verbunden
JP13	2 + 3	CAN-Treiber und Optokoppler auf dem Development Board phyCORE-HD200 werden lokal versorgt
JP29	offen	kein Spannungsabgriff über CAN-Bus

Tabelle 48: Unzulässige Jumper-Stellungen bei galvanisch getrenntem CAN-Bus (CAN-Treiber auf Development Board)

### 14.3.7 Visualisierungs LED D3

Das Development Board phyCORE-HD200 ist mit einer LED D3 ausgestattet, die als einfaches Anzeige-Element verwendet werden kann. Diese LED kann mit Hilfe des Portpins P2.0 des phyCORE-ST10F168/ST10F269 auf GPIO0 (JP17 = geschlossen) angesteuert werden. Beim Anliegen eines Low-Pegels am Port P2.0 leuchtet die LED D3, bei einem High-Pegel an GPIO0 bleibt die LED D3 erloschen.

Jumper	Stellung	Wirkung
JP17	geschlossen	Low-Pegel an Port 2.0 des ST10F168 bzw. ST10F269 bringt LED D3 zum Leuchten

Tabelle 49: JP17 Konfiguration der Visualisierungs-LED D3

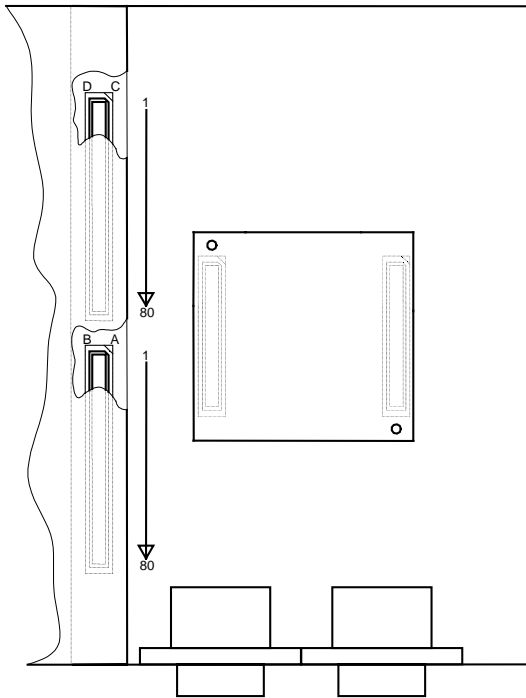
### 14.3.8 Die Belegungen des phyCORE, des Expansion-Bus und des Patchfeldes im Überblick

Wie bereits in *Kapitel 14.1* erläutert, werden alle Signale des phyCORE-ST10F168/ST10F269 mittels einer starren 1:1-Zuordnung auf den Expansion-Connector X2 geführt. Dieser wird wiederum anhand einer weiteren, ebenfalls starren 1:1-Zuordnung mit dem Patchfeld einer optional angeschlossenen Erweiterungsplatine verbunden.

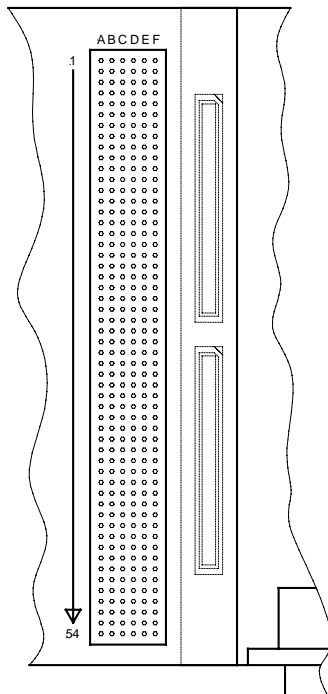
Bitte beachten Sie hierbei, daß je nach Ausführung und Größe der Erweiterungsplatine unter Umständen nur ein Teil des kompletten Patchfeldes realisiert wird. Dieser stellt eine Untermenge der hier aufgeführten Signale dar. Die Numerierungen behalten dabei trotzdem ihre Gültigkeit.

Analog zur Zählweise des phyCORE-Connectors wird auch beim Expansion-Connector und dem Patchfeld eine zweidimensionale Zählmatrix verwendet. Deren Ausrichtung weicht allerdings aus Gründen des Layouts von der des phyCORE-Connectors ab.

Bitte entnehmen sie den beiden folgenden Bildern die Ausrichtung der Zählmatrix für den Expansion-Connector sowie des Patchfeldes:



**Bild 23:** Zählweise für den Expansion-Bus



**Bild 24:** Zählweise des Patchfeldes

Für das phyCORE-ST10F168/ST10F269 auf einem Development Board phyCORE-HD200 und einer angeschlossenen Erweiterungsplatine ergeben sich folgende Pinbelegungen:

<b>Signal</b>	<b>phyCORE-ST10Fx</b>	<b>Expansion-Bus</b>	<b>Patchfeld</b>
P0L.0/D0	18B	18B	33F
P0L.1/D1	19A	19A	34A
P0L.2/D2	20A	20A	34E
P0L.3/D3	20B	20B	34B
P0L.4/D4	21A	21A	34D
P0L.5/D5	21B	21B	34F
P0L.6/D6	22B	22B	35A
P0L.7/D7	23A	23A	35E
P0H.0/D8	28B	28B	37C
P0H.1/D9	29A	29A	37E
P0H.2/D10	30A	30A	37B
P0H.3/D11	30B	30B	37F
P0H.4/D12	31A	31A	38A
P0H.5/D13	31B	31B	38C
P0H.6/D14	32B	32B	38E
P0H.7/D15	33A	33A	38B
P1L.0/A0	8B	8B	30B
P1L.1/A1	9A	9A	30D
P1L.2/A2	10A	10A	20F
P1L.3/A3	10B	10B	31A
P1L.4/A4	11A	11A	31E
P1L.5/A5	11B	11B	31B
P1L.6/A6	12B	12B	31F
P1L.7/A7	13A	13A	31A
P1H.0/A8	13B	13B	32C
P1H.1/A9	14A	14A	32E
P1H.2/A10	15A	15A	32B
P1H.3/A11	15B	15B	32F
P1H.4/A12/CC24IO	16A	16A	33A
P1H.5/A13/CC25IO	16B	16B	33C
P1H.6/A14/CC26IO	17B	17B	33E
P1H.7/A15/CC27IO	18A	18A	33B

*Tabelle 50: Daten/Adressbus-Pinzuordnung phyCORE-ST10F168/ST10F269 /  
Development Board / Erweiterungsplatine*

Signal	phyCORE-ST10Fx	Expansion-Bus	Patchfeld
P4.0/A16	23B	23B	35B
P4.1/A17	24A	24A	35D
P4.2/A18	25A	25A	35F
P4.3/A19	25B	25B	36A
P4.4/A20/CAN2_RxD	26A	26A	36E
P4.5/A21/CAN1_RxD	26B	26B	36B
P4.6/A22/CAN1_TxD	27B	27B	36F
P4.7/A23/CAN2_TxD	28A	28A	37A
P2.0/CC0IO	11D	11D	4A
P2.1/CC1IO	12D	12D	4B
P2.2/CC2IO	13C	13C	4F
P2.3/CC3IO	13D	13D	5A
P2.4/CC4IO	14C	14C	5C
P2.5/CC5IO	15C	15C	5E
P2.6/CC6IO	15D	15D	5B
P2.7/CC7IO	16C	16C	5F
P2.8/CC8IO/EX0IN	2B	2B	28E
P2.9/CC9IO/EX1IN	3A	3A	28B
P2.10/CC10IO/EX2IN	3B	3B	28F
P2.11/CC11IO/EX3IN	19C	19C	6F
P2.12 /CC12IO/EX4IN	20C	20C	7A
P2.13/CC13IO/EX5IN	37D	37D	12F
P2.14/CC14IO/EX6IN	25D	25D	8F
P2.15/CC15IO/EX7IN/ T7IN	26D	26D	9E
P3.0/T0IN	44A	44A	42E
P3.1/T6OUT	45A	45A	42B
P3.2/CAPIN	45B	45B	42F
P3.3/T3OUT	46A	46A	43A
P3.4/T3EUD	46B	46B	43C
P3.5/T4IN	47B	47B	43E
P3.6/T3IN	48A	48A	43B
P3.7/T2IN	48B	48B	43F
P3.8/MRST	42B	42B	41F
P3.9/MTSR	43A	43A	42A
P3.10/TxD0_TTL	17D	17D	6C
P3.11/RxD0_TTL	16D	16D	6A
P3.12/ /WRH /BHE	33B	33B	38F
P3.13/SCLK	43B	43B	42C
P3.15/CLKOUT	1B	1B	28C

Tabelle 51: Pinzuordnung Port P2, P3, P4 phyCORE-ST10F168/ST10F269 /  
Development Board / Erweiterungsplatine

<b>Signal</b>	<b>phyCORE-ST10Fx</b>	<b>Expansion-Bus</b>	<b>Patchfeld</b>
P5.0/AN0	50C	50C	17A
P5.1/AN1	49C	49C	16F
P5.2/AN2	48D	48D	16B
P5.3/AN3	48C	48C	16E
P5.4/AN4	47D	47D	16C
P5.5/AN5	46D	46D	16A
P5.6/AN6	46C	46C	15F
P5.7/AN7	45D	45D	15B
P5.8/AN8	45C	45C	15E
P5.9/AN9	44C	44C	15C
P5.10/AN10/T6EUD	43D	43D	15A
P5.11/AN11/T5EUD	43C	43C	14F
P5.12/AN12/T6IN	42D	42D	14B
P5.13/AN13/T5IN	41D	41D	14E
P5.14/AN14/T4EUD	41C	41C	14A
P5.15/AN15/T2EUD	40D	40D	13F
P6.0/ /CS0	49A	49A	44A
P6.1/ /CS1	50A	50A	44E
P6.2/ /CS2	6B	6B	29F
P6.3/ /CS3	5B	5B	29B
P6.4/ /CS4	5A	5A	29E
P6.5/ /HOLD	35B	35B	39B
P6.6/ /HLDA	36A	36A	39D
P6.7/ /BREQ	36B	36B	39F
P7.0/POUT0	37B	37B	40A
P7.1/POUT1	38A	38A	40E
P7.2/POUT2	38B	38B	40B
P7.3/POUT3	39A	39A	40D
P7.4/CC28IO	40A	40A	40F
P7.5/CC29IO	40B	40B	41A
P7.6/CC30IO	41A	41A	41E
P7.7/CC31IO	41B	41B	41B
P8.0/CC16IO	27D	27D	9B
P8.1/CC17IO	28D	28D	10A
P8.2/CC18IO	30D	30D	10B
P8.3/CC19IO	31D	31D	11A
P8.4/CC20IO	35C	35C	12A
P8.5/CC21IO	35D	35D	12E
P8.6/CC22IO	36C	36C	12B
P8.7/CC23IO	36D	36D	12D

*Tabelle 52: Pinzuordnung Port P5, P6, P7, P8 phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine*

Signal	phyCORE-ST10Fx	Expansion-Bus	Patchfeld
CAN1_RxD/P4.5/A21	26B	26B	36B
CAN1_TxD/P4.6/A22	27B	27B	36F
CAN2_RxD/P4.4/A20	26A	26A	36E
CAN2_TxD/P4.7/A23	28A	28A	37A
CAN-H0	21D	21D	7D
CAN-L0	20D	20D	7E
CAN-H1	18C	18C	6E
CAN-L1	18D	18D	6B
RxD0_RS232	22D	22D	7F
TxD0_RS232	23D	23D	8E
RxD1_RS232	21C	21C	7B
TxD1_RS-232	23C	23C	8A
/RTS1_RS232	24C	24C	8B
/CTS1_RS232	25C	25C	8D
/DSR1_RS232	26C	26C	9A
/DTR1_RS232	28C	28C	9F
/RII_TTL	29C	29C	10C
/CD1_TTL	30C	30C	10E
SCL	31C	31C	10F
SDA	32D	32D	11C

Tabelle 53: Schnittstellen-Pinzuordnung phyCORE-ST10F168/ST10F269 /  
Development Board / Erweiterungsplatine

Signal	phyCORE-ST10Fx	Expansion-Bus	Patchfeld
/RD	7B	7B	30A
/WR/ /WRL	8A	8A	30E
/READY	34A	34A	39A
/ALE	6A	6A	29D
VPP	4D	4D	2C
(NUR bei ST10F168)	5D	5D	1D
/RSTIN	10C, 10D	10C, 10D	3D, 3F
/RSTOUT	11C	11C	4E
BOOT	9C	9C	3B
/NMI	4A	4A	29A
/PFO	8C	8C	3E
/CS_UART	34C	34C	11F
IRQ_UART	33C	33C	11E
/IRQ_RTC	33D	33D	11B
PFI	7D	7D	2F
WDI	8D	8D	3A

Tabelle 54: Schnittstellen-Pinzuordnung phyCORE-ST10F168/ST10F269 /  
Development Board / Erweiterungsplatine



<b>Signal</b>	<b>phyCORE-ST10Fx</b>	<b>Expansion-Bus</b>	<b>Patchfeld</b>
VCC	1C, 2C, 1D, 2D	1C, 2C, 1D, 2D	1A, 1C
VCC2	Nicht definiert		2A, 1B
XTAL1	1A	1A	28A
VPD	6D	6D	2D
VBAT	6C	6C	2B
VAREF	50D	50D	17E
VAGND	42C, 47C, 39D, 44D, 9D	42C, 47C, 39D 44D, 49D	Mit GND verbunden
GND	2A, 7A, 12A, 17A, 22A, 27A, 32A, 37A, 42A, 47A, 4B, 9B, 14B, 19B, 24B, 29B, 34B, 39B, 44B, 49B, 3C, 7C, 12C, 17C, 22C, 27C, 32C, 37C, 3D, 9D, 14D, 19D, 24D, 29D, 34D	2A, 7A, 12A, 17A, 22A, 27A, 32A, 37A, 42A, 47A, 52A, 57A, 62A, 67A, 72A, 77A, 4B, 9B, 14B, 19B, 24B, 29B, 34B, 39B, 44B, 49B, 54B, 59B, 64B, 69B, 74B, 79B, 3C, 7C, 12C, 17C, 22C, 27C, 32C, 37C, 42C, 47C, 52C, 57C, 62C, 67C, 72C, 77C, 3D, 9D, 14D, 19D, 24D, 29D, 34D, 42D, 47D, 52D, 57D, 62D, 67D, 72D, 77D	3C, 4C, 7C, 8C, 9C, 12C, 13C, 14C, 17C, 18C, 19C, 22C, 23C, 24C, 27C, 29C, 30C, 31C, 34C, 35C, 36C, 39C, 40C, 41C, 44C, 45C, 46C, 49C, 50C, 51C, 54C, 4D, 5D, 6D, 9D, 10D, 11D, 14D, 15D, 16D, 9D, 20D, 21D, 24D, 25D, 26D, 28D, 31D, 32D, 33D, 36D, 37D, 38D, 41D, 42D, 43D, 46D, 47D, 48D, 51D, 52D, 53D, 1E, 2E, 1F

*Tabelle 55: Versorgungsspannungs-Zuordnung phyCORE-ST10F168/ST10F269 / Development Board / Erweiterungsplatine*

Signal	phyCORE-ST10Fx	Expansion-Bus	Patchfeld
NC	35A, 50B, 4C, 5C, 38C, 39C, 40C, 38D	51A, 53A, 54A, 55A, 56A, 58A, 59A, 60A, 61A, 63A, 64A, 65A, 66A, 68A, 69A, 70A, 71A, 73A, 74A, 75A, 76A, 78A, 79A, 80A 35A, 50B, 51B, 53B, 54B, 55B, 56B, 58B, 59B, 60B, 61B, 63B, 64B, 65B, 66B, 68B, 69B, 70B, 71B, 73B, 74B, 75B, 76B, 78B, 79B, 80B 51C, 53C, 54C, 55C, 56C, 58C, 59C, 60C, 61C, 63C, 64C, 65C, 66C, 68C, 69C, 70C, 71C, 73C, 74C, 75C, 76C, 78C, 79C, 80C 4C, 5C, 38C, 39C, 40C, 38D 51D, 53D, 54D, 55D, 56D, 58D, 59D, 60D, 61D, 63D, 64D, 65D, 66D, 68D, 69D, 70D, 71D, 73D, 74D, 75D, 76D, 78D, 79D, 80D	27B, 27D, 54D, 27F, 54F 44D, 44F, 45A, 45E, 45B, 45D, 45F, 46A, 46E, 46B, 46F, 47A, 47C, 47E, 47B, 47F, 48A, 48C, 48E, 48B, 48F, 49A, 49E, 49B, 49D, 49F, 50A, 50E, 50B, 50D, 50F, 51A, 51E, 51B, 51F, 52A, 52C, 52E, 52B, 52F, 53A, 53C, 53E, 53B, 53F, 54A, 54E, 54B

*Tabelle 56: Nicht verwendete Pins des phyCORE-ST10F168/ST10F269 beim Development Board und der Erweiterungsplatine*

### 14.3.9 Der Batterieanschluß BAT1

Der Anschluß BAT1 dient zur Bestückung einer Batterie, die die flüchtigen Speicher und die RTC auf dem phyCORE-ST10F168/ST10F269 puffert. Die Umschaltung von der normalen Spannungsversorgung auf Batteriepufferung wird vom Watchdog des phyCORE-ST10F168/ST10F269 automatisch durchgeführt. Die für diesen Anwendungsfall erforderliche Batterie (*siehe auch Kapitel 11*) kann bei der Firma PHYTEC Meßtechnik GmbH bezogen werden.

### 14.3.10 Auslösung des Interrupts /NMI

Durch entsprechende Konfiguration des Jumpers JP28 (*siehe auch Kapitel 14.3.2*) kann über den Boot-Taster auch der nicht maskierbare Hardware-Interrupt (/NMI) des ST10F168/ST10F269 ausgelöst werden.

Jumper	Stellung	Wirkung
JP28	7 + 8	Boot-Taster löst Interrupt /NMI am ST10F168/ST10F269 aus

Tabelle 57: JP28 Aktivierung /NMI Interrupt

### 14.3.11 Nummernchip

Für verschiedene Software-Applikationen im Netzwerkbereich wird für die Definition einer Knotenadresse oder als Kopierschutz ein Nummernchip eingesetzt. Der Nummernchip DS2401 kann je nach Bauform auf Position U10 oder U9 aufgelötet werden.

Der Nummernchip auf dem Development Board phyCORE-HD200 kann über das Portpin P2.1 des ST10F168/ST10F269 Controllers angesprochen werden.

Jumper	Stellung	Wirkung
JP19	geschlossen	Nummernchip mit Port 2.1 des ST10F168/ST10F269 verbunden

Tabelle 58: JP19 Konfiguration des Nummernchip

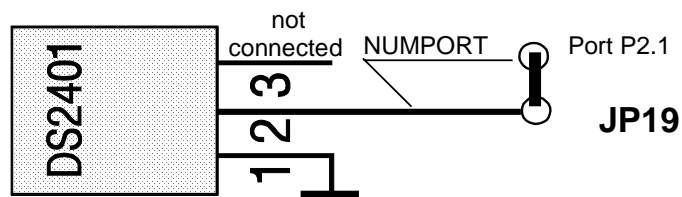


Bild 25: Anschluß des Nummernchips

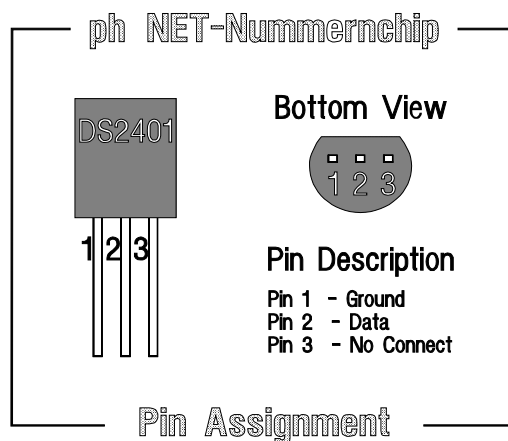


Bild 26: Pinbelegung Nummernchip

### 14.3.12 Stiftleiste X4

Die Stiftleiste X4 ist für den Anschluß einer Spannungsversorgung eines optional an die erste RS-232-Schnittstelle anschließbaren Modemmoduls gedacht. An Pin 1 des Steckers ist die Spannung 5 VDC verfügbar, am Pin 2 ist das Bezugspotential GND der Development Board phyCORE-HD200 angeschlossen. Die maximale Strombelastbarkeit ist abhängig vom verwendeten Netzteil. Es wird empfohlen, nur Geräte mit weniger als 250 mA Stromaufnahme einzusetzen.

## 15 debugCORE-ST10F168/ST10F269

Der debugCORE-ST10F168/ST10F269 ist eine 100 % funktionskompatible Erweiterung des phyCORE-ST10F168/ST10F269. Im Gegensatz zum entsprechenden phyCORE-Modul, das für den OEM-Einsatz in der Serie entwickelt wurde, dient der debugCORE-ST10F168/ST10F269 der einfachen und schnellen Fehlersuche. Hierzu verfügt dieses Modul zusätzlich zu den Funktionen des phyCORE-ST10F168/ST10F269 über Anschlußmöglichkeiten für einen Emulator und LEDs zur Anzeige des Betriebszustandes.

Aufgrund der 100 %igen Funktionskompatibilität zwischen debugCORE-ST10F168/ST10F269 und phyCORE-ST10F168/ST10F269 ist der einfache Austausch der Baugruppen zur einfachen und schnellen Fehlersuche direkt in der Kundenapplikation möglich (*siehe Bild 27*).

### 15.1 Zusatzfunktionen des debugCORE

Im Vergleich zum phyCORE-ST10F168/ST10F269 ist das debugCORE um zwei 80-polige SMD-Stecker (X2) erweitert, auf die alle notwendigen Signale geführt sind. Auf die Stecker X2 wird der nachfolgend beschriebene debugADAPTER-167 gesteckt.

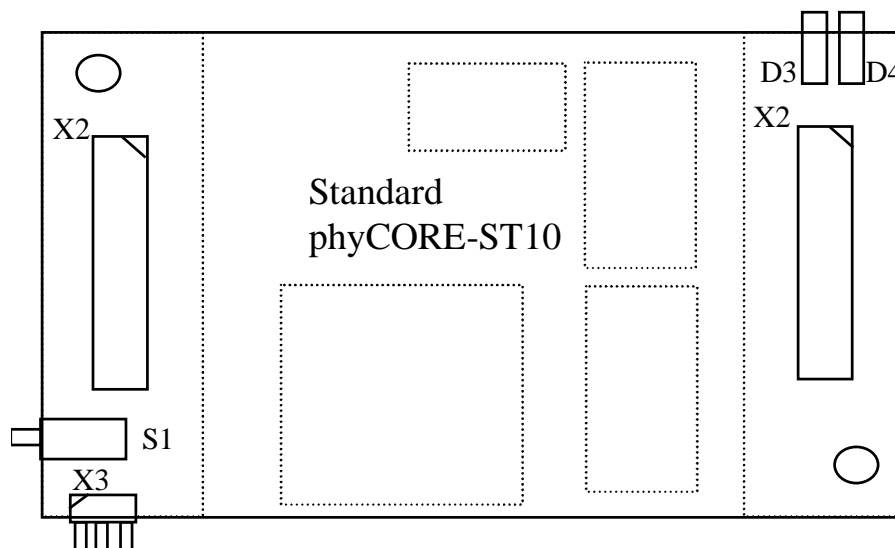


Bild 27: Lage der Anschlüsse auf dem debugCORE-ST10F168/ST10F269

Auf dem debugCORE befindet sich ein Reset-Taster S1 und eine abgewinkelte Stiftleiste X3, an der die Spannungspegel und das Resetsignal abgegriffen werden können.

Belegung der Stiftleiste X3:

Pin	Signal
1	VCC
2	GND
3	/RESET

Tabelle 59: Belegung Stiftleiste X3 des debugCORE-ST10F168/ST10F269

Für die Statusanzeige sind zusätzlich zwei LEDs D3, D4 vorhanden. LED D3 zeigt dabei an, ob sich der Controller im Adapt Mode befindet, d.h. ob die Emulation über den Emulator erfolgt. LED D4 zeigt an ob die EINIT (End of Initialization) Anweisung ausgeführt wurde.

## 15.2 debugADAPTER-167

Der debugADAPTER-167 wird auf zwei 80-polige SMD-Stecker (X2) des debugCORE gesteckt. Auf dem Adapter ist ein Quad-Connector vorhanden. Der Quad-Connector ermöglicht den direkten Anschluss eines Emulators ohne zusätzliche Erweiterungen.

Auf dem debugADAPTER ist eine Reihe von Jumpfern ausgeführt. Diese Jumper sind nur bei Betrieb auf einem debugMODUL-164 relevant. Für den Betrieb mit einem debugCORE-ST10F168/ST10F269 sind diese Jumper ohne Funktion.

### 15.2.1 Quad-Connector auf dem debugADAPTER-167

Der Quad-Connector ist die sicherste und zuverlässigste Methode für den Anschluss der Debug-Hardware zu einem Hitex- oder NOHAU-Emulator. Diese Schnittstelle beinhaltet alle ST10F168/ST10F269 Prozessorsignale und Versorgungspins. Zusätzlich ist der Quad-Connector auch die preiswerteste Alternative bei der Realisierung einer Emulationsschnittstelle.

Quad-Connector X2 A

Pin	Signal	Signal	Pin
1	NC	NC	2
3	P6.0	P6.1	4
5	P6.2	P6.3	6
7	P6.4	/HLD-P	8
9	P6.6	P6.7	10
11	P8.0	P8.1	12
13	P8.2	P8.3	14
15	P8.4	P8.5	16
17	P8.6	P8.7	18
19	VCC	GND	20
21	P7.0	P7.1	22
23	P7.2	P7.3	24
25	P7.4	P7.5	26
27	P7.6	P7.7	28
29	P5.0	P5.1	30
31	P5.2	P5.3	32
33	P5.4	P5.5	34
35	P5.6	P5.7	36
37	P5.8	P5.9	38
39	NC	NC	40

Quad-Connector X2 B

Pin	Signal	Signal	Pin
41	NC	NC	42
43	VREF	VGND	44
45	P5.10	P5.11	46
47	P5.12	P5.13	48
49	P5.14	P5.15	50
51	GND	VCC	52
53	P2.0	P2.1	54
55	P2.2	P2.3	56
57	P2.4	P2.5	58
59	P2.6	P2.7	60
61	GND	VCC	62
63	P2.8	P2.9	64
65	P2.10	P2.11	66
67	P2.12	P2.13	68
69	P2.14	P2.15	70
71	P3.0	P3.1	72
73	P3.2	P3.3	74
75	P3.4	P3.5	76
77	GND	VCC	78
79	NC	NC	80

Quad-Connector X2 C

Pin	Beschreibung	Beschreibung	Pin
81	NC	NC	82
83	P3.6	P3.7	84
85	P3.8	P3.9	86
87	P3.10	P3.11	88
89	/WRH	P3.13	90
91	P3.15	VCC	92
93	GND	VPP	94
95	A16	A17	96
97	A18	A19	98
99	A20	A21	100
101	A22	A23	102
103	VCC	GND	104
105	/RD-P	/WRL	106
107	/RDY-P	ALE	108
109	/EA	D0	110
111	D1	D2	112
113	D3	D4	114
115	D5	D6	116
117	D7	D8	118
119	NC	NC	120

Quad-Connector X2 D

Pin	Beschreibung	Beschreibung	Pin
121	NC	NC	122
123	VCC	GND	124
125	D9	D10	126
127	D11	D12	128
129	D13	D14	130
131	D15	A0	132
133	A1	A2	134
135	A3	A4	136
137	A5	A6	138
139	A7	VCC	140
141	GND	A8	142
143	A9	A10	144
145	A11	A12	146
147	A13	A14	148
149	A15	VCC	150
151	XTO	XTI	152
153	GND	/RES-P	154
155	/RESO-P	/NMI-P	156
157	GND	VCC	158
159	NC	NC	160

Tabelle 60: Anschlußbelegung des Quad-Connectors (X6) auf dem debugADAPTER-167



### 15.3 Mechanische Abmessungen debugCORE/debugADAPTER

Durch die Integration der Erweiterungsleiste X2 auf dem debugCORE-ST10F168/ST10F269 sind die mechanischen Abmessungen dieses Moduls gegenüber dem phyCORE-Basismodul grösser. Dieser Fakt ist besonders beim Einsatz des debugCORE in der Kundenapplikation zu berücksichtigen.

Abmessungen:

debugCORE-ST10F168/ST10F269	80 x 53 mm
debugADAPTER-167	81 x 66 mm

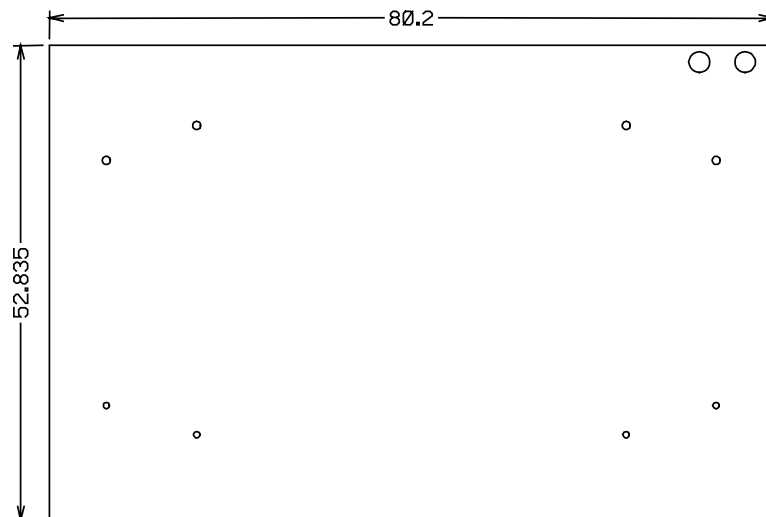


Bild 28: Mechanische Abmessungen debugCORE-ST10F168/ST10F269



## Index

<b>/</b>	
/CS0 .....	35
/NMI .....	89
<b>A</b>	
A/D-Wandler .....	21
A20 .....	20
Abmessungen .....	49
ADDRESELx .....	35
Adressleitung A18 .....	19
Aktivierung Bootstrap Loader...	62
Anschlußbelegung .....	7
Anschlüsse .....	7
<b>B</b>	
Batteriepufferung .....	47
Betriebsspannungen .....	50
Betriebstemperaturbereich .....	50
Buchse P1A .....	64
Buchse P1B .....	66
BUSCONx .....	35
Bus-Timing .....	38
<b>C</b>	
CANRx .....	40
CAN-Schnittstelle .....	26, 40
CAN-Transceiver .....	26
CANTx .....	40
Chip-Select Signale .....	35
<b>D</b>	
Development Board	
Anschlüsse .....	55
Development Board	
Funktionsgruppen .....	60
Jumper .....	57
<b>E</b>	
E <sup>2</sup> PROM .....	24
E <sup>2</sup> PROM	
Adresse .....	26
Schreibschutz .....	25
Echtzeituhr .....	41
EEPROM, seriell .....	43
EMV .....	1
ESD .....	1
Expansion-Bus .....	81
<b>F</b>	
Features .....	4
Flash .....	20
Flash Speicher .....	46
FRAM	
Adresse .....	26
Schreibschutz .....	25
FRAM, seriell .....	43
<b>G</b>	
Gewicht .....	50
<b>H</b>	
Hinweise zum Umgang .....	51
<b>I</b>	
I <sup>2</sup> C-Bus .....	24
Initialisierung des Systems .....	31
Initialisierungsroutine .....	31
<b>J</b>	
J1 .....	19
J10 .....	24
J11 .....	24
J12 .....	25
J13 .....	25
J14 .....	25

J15 .....	26
J16 .....	26
J17 .....	26
J18 .....	26
J19 .....	26
J2 .....	20
J20 .....	28
J3 .....	20
J4 .....	21
J5 .....	21
J6 .....	22
J7 .....	23
J8 .....	23
J9 .....	24
Jumper .....	15
Jumperbelegung .....	18

## **K**

Konzept des Development Board .....	53
Kurzübersicht .....	3

## **L**

Lagertemperaturbereich .....	50
LED D3 .....	81
Luftfeuchtebereich .....	50

## **M**

Modulgröße .....	50
------------------	----

## **N**

Nummernchip .....	89
-------------------	----

## **P**

P3.0 .....	25
P3.1 .....	25
P3.1/TxD0 .....	25
P3.3 .....	24
P3.4 .....	24
Patchfeld .....	81
PCA82C251 .....	40
phyCORE-Connector .....	10

Pin-Belegungen .....	81
Pinout .....	13
Port 0 .....	32
Programmierspannung .....	22
Programmspeicher, Auswahl ....	20

## **Q**

Quad-Connector .....	94
----------------------	----

## **R**

R/W-Delay .....	36
Referenzspannung .....	21
Remote Download Quelle .....	28
Remote Supervisor Chip .....	45
Reset .....	31
Reset-Taster .....	57
RS-232 Schnittstelle .....	39
RS-232-Treiber .....	39
RTC .....	24
Interrupt-Ausgang .....	24
RTC-8563 .....	41

## **S**

Serielle Schnittstelle .....	28
Si9200EY .....	40
SMT-Stiftleisten .....	7
Spannungsversorgung .....	60
Speichermodelle .....	35
Stecker P2A .....	71
Stecker P2B .....	76
Stromaufnahme .....	50
Systemeinstellungen .....	34
System-Konfiguration .....	31
System-Startup-Konfiguration ..	31

## **T**

Technische Daten .....	49
------------------------	----

## **U**

U1 .....	46
U10 .....	24, 41
U11 .....	40

---

U12.....	40	<b>V</b>	
U13.....	47	V <sub>AGND</sub> .....	21
U5.....	39	V <sub>AREF</sub> .....	21
U6.....	39	Voltage Supervisor Chip .....	47
U7.....	39	VPD .....	47
U8.....	45	VPP.....	22
U9.....	24, 43	<b>W</b>	
UART.....	39	Waitstate .....	36
extern.....	23		
on-chip.....	23		



---

**Dokument:** phyCORE-ST10F168/ST10F269

**Dokumentnummer:** L-614d\_3, Juli 2002

---

**Wie würden Sie dieses Handbuch verbessern?**

---

---

---

---

**Haben Sie in diesem Handbuch Fehler entdeckt?**

Seite

---

---

---

---

**Eingesandt von:**

Kundennummer: \_\_\_\_\_

Name: \_\_\_\_\_

Firma: \_\_\_\_\_

Adresse: \_\_\_\_\_

\_\_\_\_\_

**Einsenden an:**

PHYTEC Technologie Holding AG

Postfach 100403

D-55135 Mainz, Germany

Fax : +49 (6131) 9221-33

Published by

---

**PHYTEC**

© PHYTEC Meßtechnik GmbH 2002

Ordering No. L-614d\_3  
Printed in Germany