phyCORE-XC161

Hardware Manual

Preliminary

Edition February 2004

A product of a PHYTEC Technology Holding company

In this manual are descriptions for copyrighted products that are not explicitly indicated as such. The absence of the trademark (™) and copyright (©) symbols does not imply that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, PHYTEC Meßtechnik GmbH assumes no responsibility for any inaccuracies. PHYTEC Meßtechnik GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. PHYTEC Meßtechnik GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages which might result.

Additionally, PHYTEC Meßtechnik GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. PHYTEC Meßtechnik GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

© Copyright 2004 PHYTEC Meßtechnik GmbH, D-55129 Mainz.
Rights - including those of translation, reprint, broadcast, photomechanical or similar reproduction and storage or processing in computer systems, in whole or in part - are reserved. No reproduction may occur without the express written consent from PHYTEC Meßtechnik GmbH.

	EUROPA	NORTH AMERICA
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 order@phytec.de	1 (800) 278-9913 sales@phytec.com
Technische Hotline:	+49 (6131) 9221-31 support@phytec.de	1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	1 (206) 780-9135
Web Seite:	http://www.phytec.de	http://www.phytec.com

4th Edition February 2004

Preface 1

1 Introduction 3

1.1 Block Diagram..6

1.2 View of the phyCORE-XC161..7

2 Pin Description..9

3 Jumpers 17

3.1 J1, J2 A/D Reference Voltage 23

3.2 J3 Internal or External Program Memory..23

3.3 J4 SRAM U6/U7 Chip Select..24

3.4 J5 SRAM Memory Size..24

3.5 J6 E²PROM/FRAM Supply Voltage 25

3.6 J7 Write Protection of E²PROM /FRAM 25

3.7 J8, J9 Address of the Serial E²PROM/ FRAM..26

3.8 J10 RS-232 Transceiver R1O TTL Output..26

3.9 J11, J12, J15, J16 CAN Interfaces..27

3.10 J13, J14, J17, J18 CAN Transceiver..28

3.11 J19 RTC Interrupt Output..29

3.12 J20, J21 Configuration of P9.4 and P9.5 for I²C Bus..29

3.13 J22 RTC (U13) Clock Output..30

3.14 J23, J24 First Serial Interface 30

3.15 J25, J26 Second Serial Interface..31

3.16 J27 Ethernet Controller Chip Select 32

3.17 J28 Ethernet Controller /SBHE Configuration..32

3.18 J29 Ethernet Controller Sleep Mode..33

3.19 J30 Ethernet Controller IRQ Signal..33

3.20 J32 Write Protection of Flash 34

4 System Configuration..35

4.1 System Startup Configuration..35

5 Memory Models..39

5.1 Bus Cycle Phases 42

6 Serial Interfaces..45

6.1 RS-232 Interface..45

6.2 CAN Interface..46

7 Real-Time Clock RTC-8564 (U13)..47

8 Serial E²PROM/FRAM (U8) 49

9 Flash Memory (U4)..51

10 Battery Buffer and Voltage Supervisor Chip (U9)......................... 52
11 CS8900A Ethernet Controller... 53
11.1 Fundamentals.. 53
11.2 Memory Mode.. 53
12 Debug Interface.. 55
13 Technical Specifications... 57
14 Hints for Handling the phyCORE-XC161..................................... 59
15 The phyCORE-XC161 on the phyCORE Development Board HD200 XC.. 61
15.1 Concept of the phyCORE Development Board HD200............. 61
15.2 Development Board HD200 XC Connectors and Jumpers........ 63
15.2.1 Connectors... 63
15.2.2 Jumpers on the phyCORE Development Board HD200 XC.. 65
15.2.3 Unsupported Features and Improper Jumper Settings 67
15.3 Functional Components on the phyCORE Development Board HD200 XC.. 68
15.3.1 Power Supply at X1... 69
15.3.2 Activating the Bootstrap Loader 71
15.3.3 First Serial Interface at Socket P1A............................. 73
15.3.4 Power Supply to External Devices via Socket P1A...... 75
15.3.5 Second Serial Interface at Socket P1B.......................... 77
15.3.6 First CAN Interface at Plug P2A.................................. 80
15.3.7 Second CAN Interface at Plug P2B 86
15.3.8 Programmable LED D3... 91
15.3.9 Pin Assignment Summary of the phyCORE, the Expansion Bus and the Patch Field............................... 91
15.3.10 Battery Connector BAT1.. 100
15.3.11 Releasing the /NMI Interrupt 100
15.3.12 DS2401 Silicon Serial Number.................................. 100
15.3.13 Pin Header Connector X4 101
15.3.14 JP40, S3 Multi-Purpose Push Button Configuration... 102
16 Ethernet Port.. 103
17 Revision History 105
Index 107

Index of Figures

Figure 1: Block Diagram phyCORE-XC161 6

Figure 2: View of the phyCORE-XC161 (Top View) 7

Figure 3: View of the phyCORE-XC161 (Bottom View)............................8

Figure 4: Pinout of the phyCORE-Connector (Top View, with Cross Section Insert)...11

Figure 5: Numbering of the Jumper Pads..17

Figure 6: Location of the Jumpers (Controller Side) 17

Figure 7: Location of the Jumpers (Connector Side) 18

Figure 8: Timing Phases for Multiplexed Bus Mode.................................40

Figure 9: Timing Phases for Demultiplexed Bus Mode.............................40

Figure 10: JTAG Interface (Top View) 55

Figure 11: JTAG Interface (Bottom View) 56

Figure 12: Physical Dimensions..57

Figure 13: Modular Development and Expansion Board Concept with the phyCORE-XC161 62

Figure 14: Location of Connectors on the phyCORE Development Board HD200 XC 63

Figure 15: Numbering of Jumper Pads 65

Figure 16: Location of the Jumpers (View of the Component Side) 65

Figure 17: Default Jumper Settings of the phyCORE Development Board HD200 XC with phyCORE-XC161................................66

Figure 18: Connecting the Supply Voltage at X1 69

Figure 19: Pin Assignment of the DB-9 Socket P1A as First RS-232 (Front View) 73

Figure 20: Location of Components at U12 and U13 for Power Supply to External Subassemblies 75

Figure 21: Pin Assignment of the DB-9 Socket P1B as Second RS-232 (Front View) 77

Figure 22: Pin Assignment of the DB-9 Plug P2A (CAN Transceiver on phyCORE-XC161, Front View) 80

Figure 23: Pin Assignment of the DB-9 Plug P2A (CAN Transceiver on Development Board) 82

Figure 24: Pin Assignment of the DB-9 Plug P2A (CAN Transceiver on Development Board with Galvanic Separation)............ 84

Figure 25: Pin Assignment of the DB-9 Plug P2B (CAN Transceiver on phyCORE-XC161) 86

Figure 26: Pin Assignment of the DB-9 Plug P2B (CAN Transceiver on phyCORE-XC161) 87

Figure 27: Pin Assignment of the DB-9 Plug P2B (CAN Transceiver on Development Board with Galvanic Separation)............ 90

Figure 28: Pin Assignment Scheme of the Expansion Bus............ 92

Figure 29: Pin Assignment Scheme of the Patch Field............ 92

Figure 30: Pin Assignment Control Signals for the phyCORE-XC161 / Development Board / Expansion Board............ 97

Figure 31: Pin Assignment Power Supply for the phyCORE-XC161 / Development Board / Expansion Board............ 98

Figure 32: Unused Pins on the phyCORE-XC161 / Development Board / Expansion Board............ 99

Figure 33: JP28 Releasing the /NMI Interrupt............ 100

Figure 34: JP19 Jumper Configuration for Silicon Serial Number Chip 100

Figure 35: Connecting the DS2401 Silicon Serial Number............ 101

Figure 36: Pin Assignment of the DS2401 Silicon Serial Number............ 101

Figure 37: JP40 Multi-Purpose Push Button S3 Configuration............ 102

Figure 38: Ethernet Transformer Module Connector 103

Figure 39: Ethernet Transformer Connector Pinout 103

Figure 40: Jumper for Ethernet Transformer Port 104

Index of Tables

Table 1 Pinout of the phyCORE-Connector X1 15

Table 2: Jumper Settings...22

Table 3: J1, J2 A/D Converter Reference Voltage 23

Table 4: J3 Code Fetch Selection...23

Table 5: J4 SRAM U6/U7 Chip Select Configuration 24

Table 6: J5 SRAM U6/U7 Size Configuration 24

Table 7: J6 E²PROM Supply Voltage Configuration 25

Table 8: J7 Write Protection of E²PROM/FRAM 25

Table 9: J8, J9 E²PROM/FRAM Address Configuration 26

Table 10: J10 RS-232 Transceiver R1O TTL Output................................26

Table 11: J11, J12, J15, J16 CAN Interface Configuration.......................28

Table 12: J19 P2.11 / RTC Interrupt Configuration 29

Table 13: J20, J21 P9.4, P9.5 / I^{2}C Bus Configuration.............................29

Table 14: J22 RTC Clock Output Configuration......................................30

Table 15: J23, J24 First Serial Interface Configuration.............................31

Table 16: J25, J26 Second Serial Interface Configuration 31

Table 17: J27 Ethernet Controller Chip Select Configuration...................32

Table 18: J28 Ethernet Controller /SBHE Configuration..........................32

Table 19: J29 Ethernet Controller Sleep Mode Configuration..................33

Table 20: J30 Ethernet Controller IRQ Signal Configuration 33

Table 21: J32 Write Protection of Flash 34

Table 22: Functional Settings on Port P0 for System Startup Configuration...36

Table 23: System Startup Configuration Registers.....................................38

Table 24: Bus Timing Calculation Examples 41

Table 25: Memory Device Options for U8 49

Table 26: E²PROM/FRAM Write Protection 49

Table 27: E²PROM/FRAM Address..50

Table 28: Improper Jumper Setting for JP16 on the Development Board....................67

Table 29: Improper Jumper Setting for JP30/33 on the Development Board....................67

Table 30: JP9, JP16 Configuration of the Supply Voltages VCC/VCC2....................69

Table 31: JP9, JP16 Improper Jumper Settings for the Supply Voltages....................70

Table 32: JP28 Configuration of the Boot Button....................71

Table 33: JP28 Configuration of a Permanent Bootstrap Loader Start.....72

Table 34: JP22, JP23, JP10 Configuration of Boot via RS-232................72

Table 35: Improper Jumper Settings for Boot via RS-232....................72

Table 36: Jumper Configuration for the First RS-232 Interface................73

Table 37: Improper Jumper Settings for DB-9 Socket P1A as First RS-232....................74

Table 38: JP24 Power Supply to External Devices Connected to P1A on the Development Board....................76

Table 39: Jumper Configuration for the Second RS-232 Interface............77

Table 40: Improper Jumper Settings for DB-9 Socket P1B (2nd RS-232)....................78

Table 41: Jumper Configuration of the DB-9 Socket P1B (no Second RS-232)....................78

Table 42: Improper Jumper Settings for DB-9 Socket P1B (no Second RS-232)....................79

Table 43: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the phyCORE-XC161....................80

Table 44: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the Development Board....................81

Table 45: Improper Jumper Settings for the CAN Plug P2A (CAN Transceiver on the Development Board)....................82

Table 46: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the Development Board with Galvanic Separation....................83

Table 47: JP39 CAN Bus Voltage Supply Reduction....................84

Table 48: Improper Jumper Settings for the CAN Plug P2A (CAN Transceiver on Development Board with Galvanic Separation)....................85

Table 49: Jumper Configuration for CAN Plug P2B using the CAN Transceiver on the phyCORE-XC161....................86

Table 50: Jumper Configuration for CAN Plug P2B using the CAN Transceiver on the Development Board....................87

Table 51: Improper Jumper Settings for the CAN Plug P2B (CAN Transceiver on the Development Board)....................88

Table 52: Jumper Configuration for CAN Plug P2B using the CAN Transceiver on the Development Board with Galvanic Separation....................89

Table 53: Improper Jumper Settings for the CAN Plug P2B (CAN Transceiver on Development Board with Galvanic Separation)....................90

Table 54: JP17 Configuration of the Programmable LED D3....................91

Table 55: Pin Assignment Data/Address Bus for the phyCORE-XC161 / Development Board / Expansion Board....................93

Table 56: Pin Assignment Port P2, P3, P4 for the phyCORE-XC161 / Development Board / Expansion Board....................94

Table 57: Pin Assignment Port P5, P6, P7, P9 for the phyCORE-XC161 / Development Board / Expansion Board....................95

Table 58: Pin Assignment Interface Signals for the phyCORE-XC161 / Development Board / Expansion Board....................96

Preface

This phyCORE-XC161 Hardware Manual describes the board's design and functions. Precise specifications for Infineon's XC161CJ microcontroller series controller can be found in the enclosed microcontroller Data Sheet/User's Manual. If software is included please also refer to additional documentation for this software.

In this hardware manual and in the attached schematics, low active signals are denoted by a "/" in front of the signal name (i.e.: /RD). A "0" indicates a logic-zero or low-level signal, while a "1" represents a logic-one or high-level signal.

Declaration of Electro Magnetic Conformity of the PHYTEC phyCORE-XC161

PHYTEC Single Board Computers (henceforth products) are designed for installation in electrical appliances or as dedicated Evaluation Boards (i.e.: for use as a test and prototype platform for hardware/software development) in laboratory environments.

Caution:
PHYTEC products lacking protective enclosures are subject to damage by ESD and, hence, may only be unpacked, handled or operated in environments in which sufficient precautionary measures have been taken in respect to ESD-dangers. It is also necessary that only appropriately trained personnel (such as electricians, technicians and engineers) handle and/or operate these products. Moreover, PHYTEC products should not be operated without protection circuitry if connections to the product's pin header rows or connectors are longer than 3 meters.

PHYTEC products fulfill the norms of the European Union's Directive for Electro Magnetic Conformity only in accordance to the descriptions and rules of usage indicated in this hardware manual (particularly in respect to the pin header rows or connectors, power connector and serial interface to a host-PC).

Implementation of PHYTEC products into target devices, as well as user modifications and extensions of PHYTEC products, is subject to renewed establishment of conformity to, and certification of, Electro Magnetic Directives. Users should ensure conformance following any modifications to the products as well as implementation of the products into target systems.

The phyCORE-XC161 is one of a series of PHYTEC Single Board Computers that can be populated with different controllers and, hence, offers various functions and configurations. PHYTEC supports all common 8- and 16-bit controllers in two ways:

(1) as the basis for Rapid Development Kits which serve as a reference and evaluation platform

(2) as insert-ready, fully functional micro-, mini- and phyCORE OEM modules, which can be embedded directly into the user's peripheral hardware, design.

PHYTEC's microcontroller modules allow engineers to shorten development horizons, reduce design costs and speed project concepts from design to market.

1 Introduction

The phyCORE-XC161 belongs to PHYTEC's phyCORE Single Board Computer module family. The phyCORE SBCs represent the continuous development of PHYTEC Single Board Computer technology. Like its mini-, micro- and nanoMODUL predecessors, the phyCORE boards integrate all core elements of a microcontroller system on a subminiature board and are designed in a manner that ensures their easy expansion and embedding in peripheral hardware developments.

As independent research indicates that approximately 70 % of all EMI (Electro Magnetic Interference) problems stem from insufficient supply voltage grounding of electronic components in high frequency environments the phyCORE board design features an increased pin package. The increased pin package allows dedication of approximately 20 % of all pin header connectors on the phyCORE boards to Ground. This improves EMI and EMC characteristics and makes it easier to design complex applications meeting EMI and EMC guidelines using phyCORE boards even in high noise environments.

phyCORE boards achieve their small size through modern SMD technology and multi-layer design. In accordance with the complexity of the module, 0402-packaged SMD components and laser-drilled Microvias are used on the boards, providing phyCORE users with access to this cutting edge miniaturization technology for integration into their own design.

The phyCORE-XC161 is a subminiature (60 x 53 mm) insert-ready Single Board Computer populated with Infineon’s XC161CJ microcontroller. Its universal design enables its insertion in a wide range of embedded applications. All controller signals and ports extend from the controller to high-density pitch (0.635 mm) connectors aligning two sides of the board, allowing it to be plugged like a “big chip” into a target application.

Precise specifications for the controller populating the board can be found in the applicable controller User’s Manual or Data Sheet. The descriptions in this manual are based on the Infineon XC161CJ. No description of compatible microcontroller derivative functions is included, as such functions are not relevant for the basic functioning of the phyCORE-XC161.

The phyCORE-XC161 offers the following features:

- subminiature Single Board Computer (60 x 53 mm) achieved through modern SMD technology
- populated with the Infineon XC161CJ microcontroller (TQFP-144 packaging) featuring two on-chip 2.0B CAN modules
- improved interference safety achieved through multi-layer PCB technology and dedicated Ground pins
- controller signals and ports extend to two 100-pin high-density (0.635 mm) Molex connectors aligning two sides of the board, enabling it to be plugged like a "big chip" into target application
- 16-bit, demultiplexed bus mode
- 20 – 40 MHz clock frequency (50 ns – 25 ns instruction cycle)
- 16 MByte address space
- 256 kByte to 2 MByte external Flash on-board[1]
- on-board Flash programming, no dedicated Flash programming voltage required through use of 5 V Flash devices
- 256 kByte to 1 MByte RAM on-board[1]
- 512 kByte fast SRAM (15 ns access time) on-board[1]
- up to 2[1] CAN interfaces with Philips 82C251 CAN transceiver, or Infineon TLE6250
- I²C Real-Time Clock with internal quartz
- 4 to 32 kByte I^{2}C E²PROM[1], or 512 Byte to 8 kByte FRAM[1]
- Voltage Supervisory Chip for Reset logic and power supervision
- free Chip Select signals for easy connection of peripheral devices[2]
- operates with two supply voltages, 5 V and 2.5 V / <220 mA
- RS-232 transceiver for two serial interfaces
- optional CS8900A 10Base-T Ethernet controller

[1] : Please contact PHYTEC for more information about additional modul configurations.

[2] : Number of available /CS signals depends on configuration of the phyCORE module.

1.1 Block Diagram

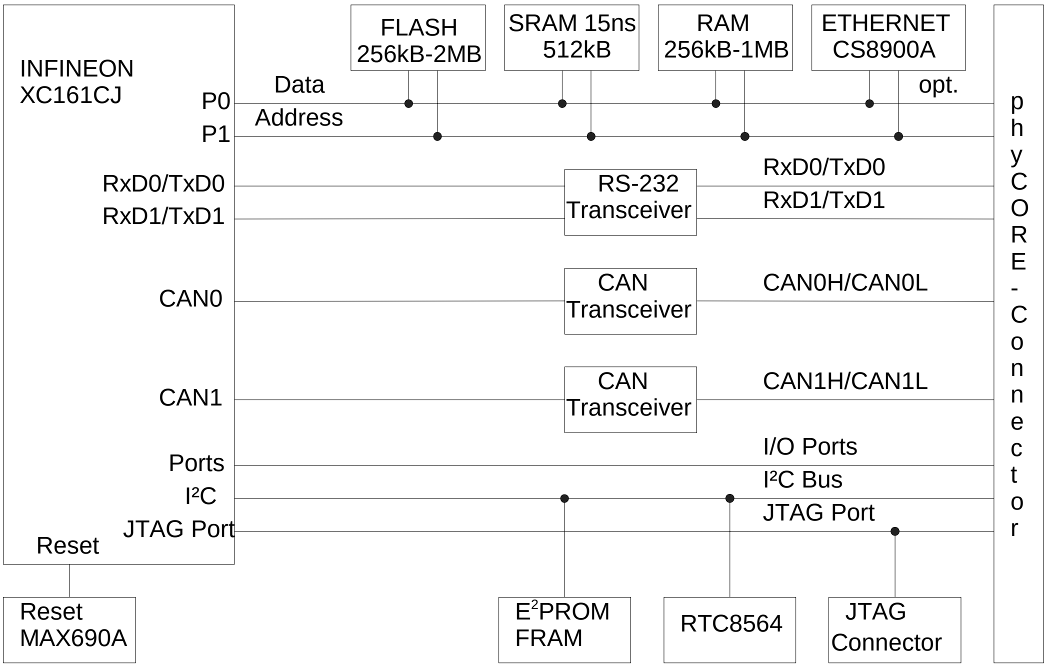


Figure 1: Block Diagram phyCORE-XC161

1.2 View of the phyCORE-XC161

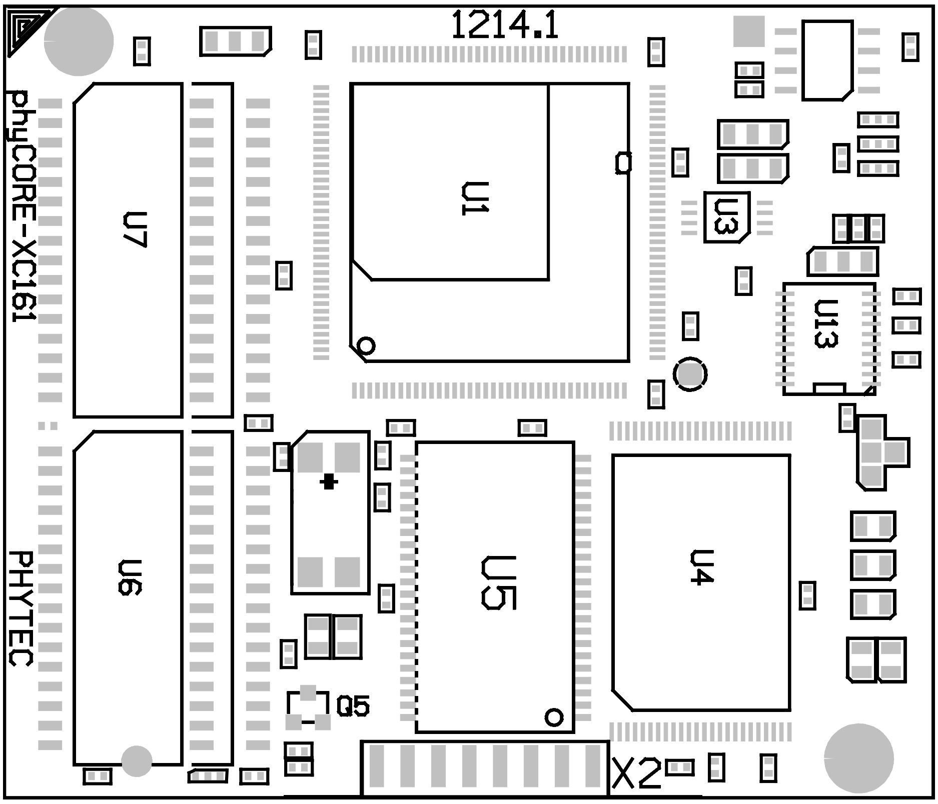


Figure 2: View of the phyCORE-XC161 (Top View)

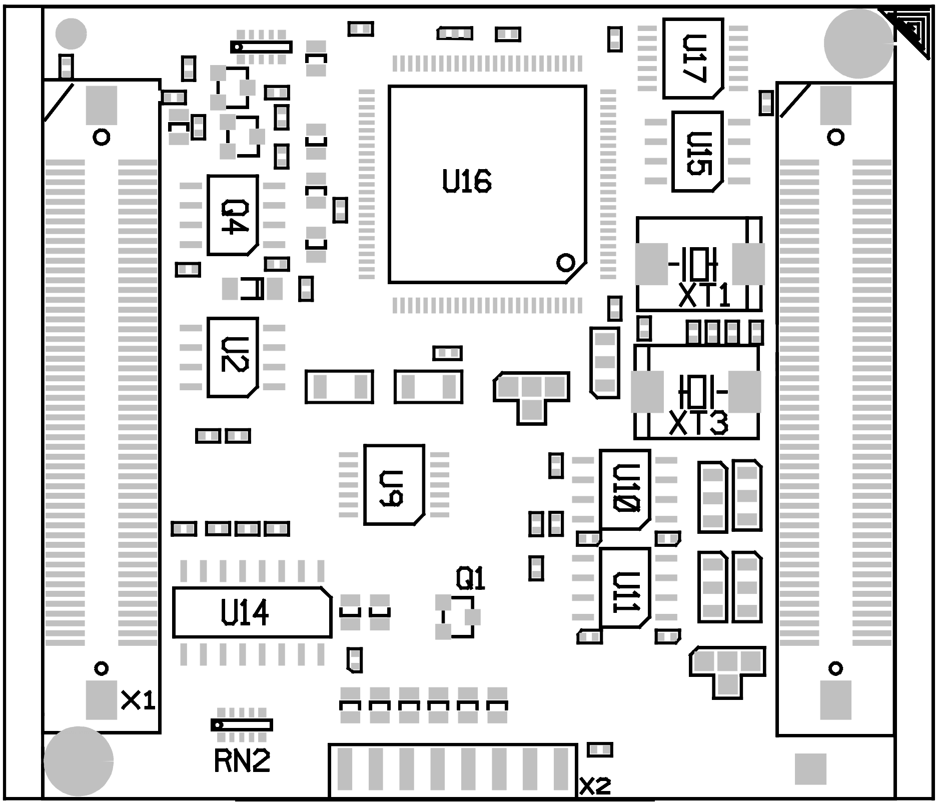


Figure 3: View of the phyCORE-XC161 (Bottom View)

2 Pin Description

Please note that all module connections are not to exceed their expressed maximum voltage or current. Maximum signal input values are indicated in the corresponding controller manuals/data sheets. As damage from improper connections varies according to use and application, it is the user’s responsibility to take appropriate safety measures to ensure that the module connections are protected from overloading through connected peripherals.

As *Figure 4* indicates, all controller signals extend to surface mount technology (SMT) connectors (0.635 mm) lining two sides of the module (referred to as phyCORE-connector). This allows the phyCORE-XC161 to be plugged into any target application like a "big chip".

A new numbering scheme for the pins on the phyCORE-connector has been introduced with the phyCORE specifications. This enables quick and easy identification of desired pins and minimizes errors when matching pins on the phyCORE module with the phyCORE-connector on the appropriate PHYTEC Development Board or in user target circuitry.

The numbering scheme for the phyCORE-connector is based on a two dimensional matrix in which column positions are identified by a letter and row position by a number. Pin 1A, for example, is always located in the upper left hand corner of the matrix. The pin numbering values increase moving down on the board. Lettering of the pin connector rows progresses alphabetically from left to right (*refer to Figure 4*).

The numbered matrix can be aligned with the phyCORE-XC161 (viewed from above; phyCORE-connector pointing down) or with the socket of the corresponding phyCORE Development Board/user target circuitry. The upper left-hand corner of the numbered matrix (pin 1A) is thus covered with the corner of the phyCORE-XC161 marked with a white triangle. The numbering scheme is always in relation to the PCB as viewed from above, even if all connector contacts extend to the bottom of the module.

The numbering scheme is thus consistent for both the module's phyCORE-connector as well as mating connectors on the phyCORE Development Board or target hardware, thereby considerably reducing the risk of pin identification errors.

Since the pins are exactly defined according to the numbered matrix previously described, the phyCORE-connector is usually assigned a single designator for its position (X1 for example). In this manner the phyCORE-connector comprises a single, logical unit regardless of the fact that it could consist of more than one physical socketed connector. The location of row 1 on the board is marked by a white triangle on the PCB to allow easy identification.

The following figure (*Figure 4*) illustrates the numbered matrix system. It shows a phyCORE-XC161 with SMT phyCORE-connectors on its underside (defined as dotted lines) mounted on a Development Board. In order to facilitate understanding of the pin assignment scheme, the diagram presents a crossview of the phyCORE module showing these phyCORE-connectors mounted on the underside of the module's PCB.

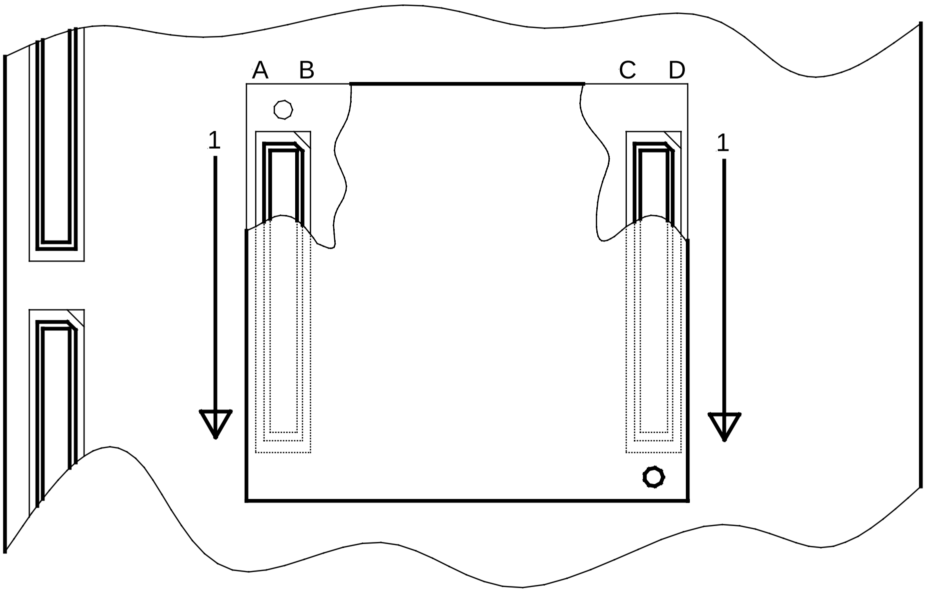


Figure 4: Pinout of the phyCORE-Connector (Top View, with Cross Section Insert)

Many of the controller port pins accessible at the connectors along the edges of the board have been assigned alternate functions that can be activated via software.

Fehler! Verweisquelle konnte nicht gefunden werden. provides an overview of the pinout of the phyCORE-connector , as well as descriptions of possible alternative functions. *Please refer to the Infineon XC161 User's Manual/Data Sheet for details on the functions and features of controller signals and port pins.*

Pin Number	Signal	I/O	Description
Pin Row X1A			
1A	CLKIN	I	Optional external clock generator
2A, 7A, 12A, 17A, 22A, 27A, 32A, 37A, 42A, 47A	GND	-	Ground 0 V
3A	P2.9	I/O	CAPCOM1: CC9 Capture Input/Compare Output Fast external Interrupt 1 Input (I)
4A	/NMI	I	Non-masked interrupt input
5A	P6.4, /CS4	O	Chip Select #4
6A	ALE	O	Address latch enable
8A	/WR (def.)/WRL	O	/WRL (/WR) signal of the microcontroller
9A, 10A, 11A, 13A, 14A, 15A, 16A, 18A, 24A, 25A, 26A, 28A	A1, A2, A4, A7, A9, A10, A12, A15, A17, A18, A20, A23	O	Address lines of the microcontroller
19A, 20A, 21A, 23A, 29A, 30A, 31A, 33A	D1, D2, D4, D7, D9, D10, D12, D15	I/O	Data lines of the microcontroller
34A	/READY	I	Microcontroller READY signal input
35A, 38A ,39A	NC	-	Not connected. These contacts should remain unconnected on the target hardware side.
36A	P6.6, /HLDA	I/O	Acknowledge output (master mode)/ input (slave mode)
40A, 41A	P7.4, P7.6	I/O	CAPCOM2:CC28 Capture Input/Compare Output CAPCOM2:CC30 Capture Input/Compare Output
43A	P3.9	I/O	SSC Master transmit/Slave receive
44A	P3.0	I	CAPCOM1 Timer T0 Counter input
45A	P3.1	O	GPT12E Timer T6 latch output
46A	P3.3	O	GPT12E Timer T3 latch output
48A	P3.6	I	GPT12E Timer T3 counter input
49A	P6.0, /CS0	O	Chip Select #0
50A	P6.1, /CS1	O	Chip Select #1

Pin Number	Signal	I/O	Description
Pin Row X1B			
1B	P3.15/CLKOUT	O	CLKOUT system clock output
2B, 3B	P2.8, P2.10	I/O	CAPCOM1: CC8 Capture Input/Compare Output Fast external Interrupt 0 Input (I) CAPCOM1: CC10 Capture Input/Compare Output Fast external Interrupt 2 Input (I)
4B, 9B, 14B, 19B, 24B, 29B, 34B, 39B, 44B, 49B	GND	-	Ground 0 V
5B	P6.3, /CS3	O	Chip Select #3
6B	P6.2, /CS2	O	Chip Select #2
7B	/RD	O	/RD signal of the microcontroller
8B, 10B, 11B, 12B, 13B, 15B, 16B, 17B, 23B, 25B, 26B, 27B,	A0, A3, A5, A6, A8, A11, A13, A14, A16, A19, A21, A22	O	Address lines of the microcontroller
18B, 20B, 21B, 22B, 28B, 30B, 31B, 32B	D0, D3, D5, D6, D8, D11, D13, D14	I/O	Data lines of the microcontroller
33B	P3.12, /BHE	O	Microcontroller /WRH (or /BHE) signal
35B	P6.5, /HOLD	I	Microcontroller /HOLD signal
36B	P6.7, /BREQ	O	Microcontroller /BREQ signal
37B, 38B	NC	-	Not connected. These contacts should remain unconnected on the target hardware side.
40B, 41B	P7.5, P7.7	I/O	CAPCOM2:CC29 Capture Input/Compare Output CAPCOM2:CC31 Capture Input/Compare Output
42B, 43B, 45B, 46B, 47B, 48B	P3.8, P3.13, P3.2, P3.4, P3.5, P3.7	I/O	Port 3 of the microcontroller (*see corresponding Data Sheet*)
50B	/CS_ETH	O	Chip Select Ethernet Controller (*refer to J26*)

Pin Number	**Signal**	**I/O**	**Description**
Pin Row X1C			
1C, 2C	VCC	-	Voltage input +5 VDC
3C, 7C, 12C, 17C, 22C, 27C, 32C, 37C	GND	-	Ground 0 V
4C, 5C	VCC2	-	Voltage input +2.5 VDC
6C	VBAT	I	Battery input for back-up of RTC
8C	/PFO	O	MAX 690/ Power Fail output
9C	BOOT	I	Input for starting Bootstrap mode
10C	/RESET	I	/RESET input of the phyCORE-XC161
11C	/RESOUT	O	/RESOUT signal of µC
13C, 14C, 15C	P9.2, P9.4, P9.5	I/O	Port 9 of the microcontroller (*see corresponding Data Sheet*)
16C, 30C, 39C. 40C	NC	-	Not connected These contacts should remain unconnected on the target hardware side.
18C	CAN-H1	I/O	Differential CANH line of second CAN transceiver
19C	RxD1_TTL	I	Input of the second serial interface of the phyCORE-XC161, TTL level
20C	TxD1_TTL	O	Output of the second serial interface of the phyCORE-XC161, TT: level
21C	RxD1_RS-232	I	Input of the second serial interface of the phyCORE-XC161, RS-232 level
23C	TxD1_RS-232	O	Output of the second serial interface of the phyCORE-XC161, RS-232 level
24C	TDI	O	Test Data Input TDI (JTAG)
25C	TDO	I	Test Data Output TDO (JTAG)
26C	TMS	I	Test Mode Select TMS (JTAG)
28C	TCK	I	Test Clock TCK (JTAG)
29C	/TRST	I	Test Reset (JTAG)
31C	SCL	O	CLK line I^{2}C bus
33C	LINK_LED	O	LINK_LED (Ethernet)
34C	LAN_LED	I	LAN_LED (Ethernet)
35C,	RxD-	O	RxD- (Ethernet)
36C	TxD-	I	TxD- (Ethernet)
38C	P212	I/O	CAPCOM1: CC12 Capture input/Compare output Fast external interrupt 4 input (I)
41C, 43C, 44C, 45C, 46C, 48C, 49C, 50C	P5.14, P5.11, P5.9, P5.8, P5.6, P5.3, P5.1, P5.0	I/O	Port 5 of the microcontroller (*see corresponding Data Sheet*)
42C, 47C	VAGND	-	Analog Ground of the microcontroller

Pin Number	Signal	I/O	Description
Pin Row X1D			
1D, 2D	VCC	-	Voltage input +5 VDC
3D, 9D, 14D, 19D, 24D, 29D, 34D	GND	-	Ground 0 V
4D, 5D, 15D	NC	-	Not connected These contacts should remain unconnected on the target hardware side.
6D	VPD	O	Output of back-up voltage supply for buffering of external components
7D	PFI	I	MAX 690 power fail input. If this input is unused, it must be connected to VCC or GND
8D	WDI	I	MAX 690 Watchdog input
10D	/RESET	I	/RESET input of the phyCORE-XC161
11D, 12D, 13D	P9.0, P9.1, P9.3	I/O	Port 9 of the microcontroller (*see corresponding Data Sheet*)
15D	Plex	O	TTL level output on the RS-232 transceiver (only if Jumper J10 is closed, refer to section 3.8)
16D	P3.11, RxD0	I	Input of the first serial interface, TTL level
17D	P3.10, TxD0	O	Output of the first serial interface, TTL level
18D	CAN-L1	I/O	Differential CANL line of the 2nd CAN transceiver
20D	CAN-L0	I/O	Differential CANL line of the 1st CAN transceiver
21D	CAN-H0	I/O	Differential CANH line of the first CAN transceiver
22D	RxD0_RS-232	I	Input of the first serial interface, RS-232 level
23D	TxD0_RS-232	O	Output of the first serial interface, RS-232 level
25D, 26D	P2.14, P2.15	I/O	CAPCOM1: CC14 Capture Input/Compare Output, Fast ext. Interrupt 6 Input (I) CAPCOM1: CC15 Capture Input/Compare Output Fast external Interrupt 7 Input (I) T7IN Timer T7 Count Input (I)
27D	/BRKIN	I	Break input, only available in Break mode
28D	/BRKOUT	O	Break output, only available in Break mode
30D	RTC_CLKOUT	O	RTC Clock Output
31D	IRQ_ETH	O	Interrupt output of the Ethernet controller
32D	SDA	O	Data line I^{2}C bus
33D	/IRQ_RTC	O	Interrupt output of the RTC
35D,	RxD+	I	RxD+ (Ethernet)
36D	TxD+	O	TxD+ (Ethernet)
37D	P2.13	I/O	CAPCOM1: CC13 Capture Input/Compare Output Fast ext. Interrupt 5 Input (I)
38D	P2.11	I/O	CAPCOM1: CC11 Capture Input/Compare Output Fast ext. Interrupt 3 Input (I)
39D, 44D, 49D	VAGND	-	Analog Ground
40D, 41D, 42D, 43D, 45D, 46D, 47D, 48D,	P5.15,P5.13,P5.12, P5.10,P5.7, P5.5, P5.4, P5.2	I	Port 5 of the microcontroller (*see corresponding Data Sheet*)
50D	VAREF	I	Reference voltage input for A/D converter

Table 1 Pinout of the phyCORE-Connector X1

3 Jumpers

For configuration purposes, the phyCORE-XC161 has 32 solder jumpers, some of which have been installed prior to delivery. *Figure 5* illustrates the numbering of the jumper pads, while *Figure 7* indicates the location of the jumpers on the board.

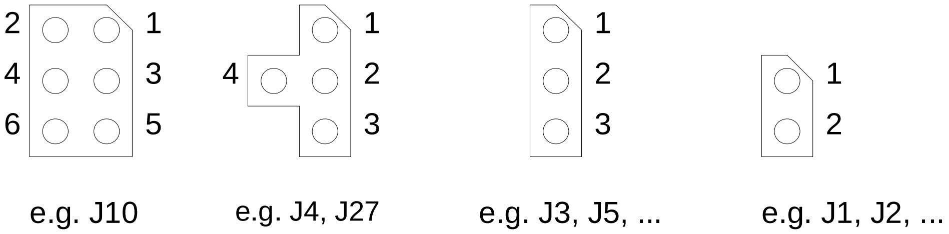


Figure 5: Numbering of the Jumper Pads

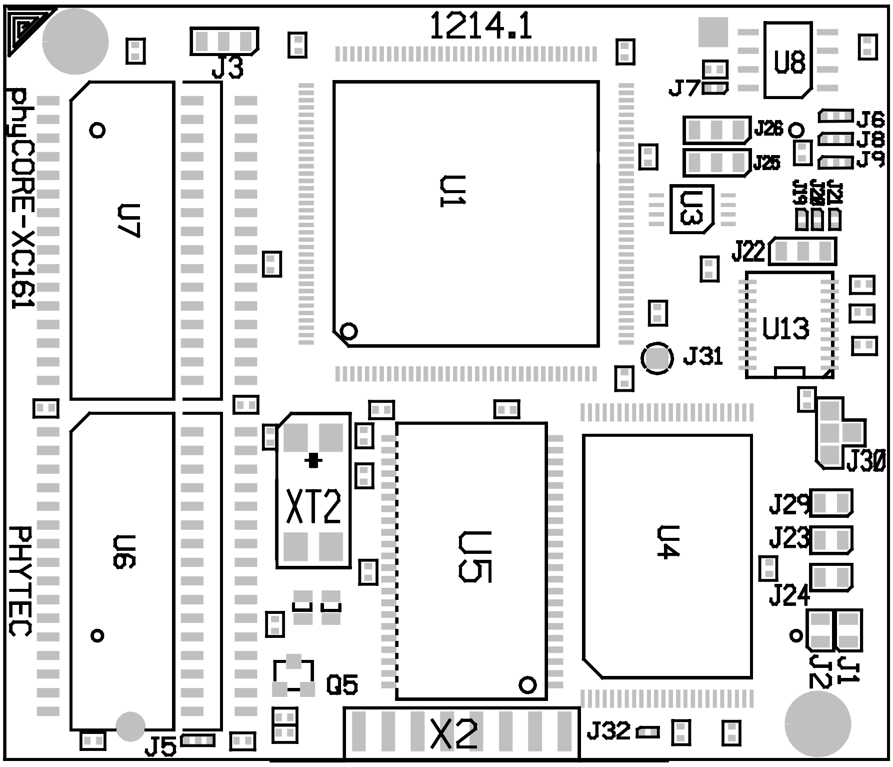


Figure 6: Location of the Jumpers (Controller Side)

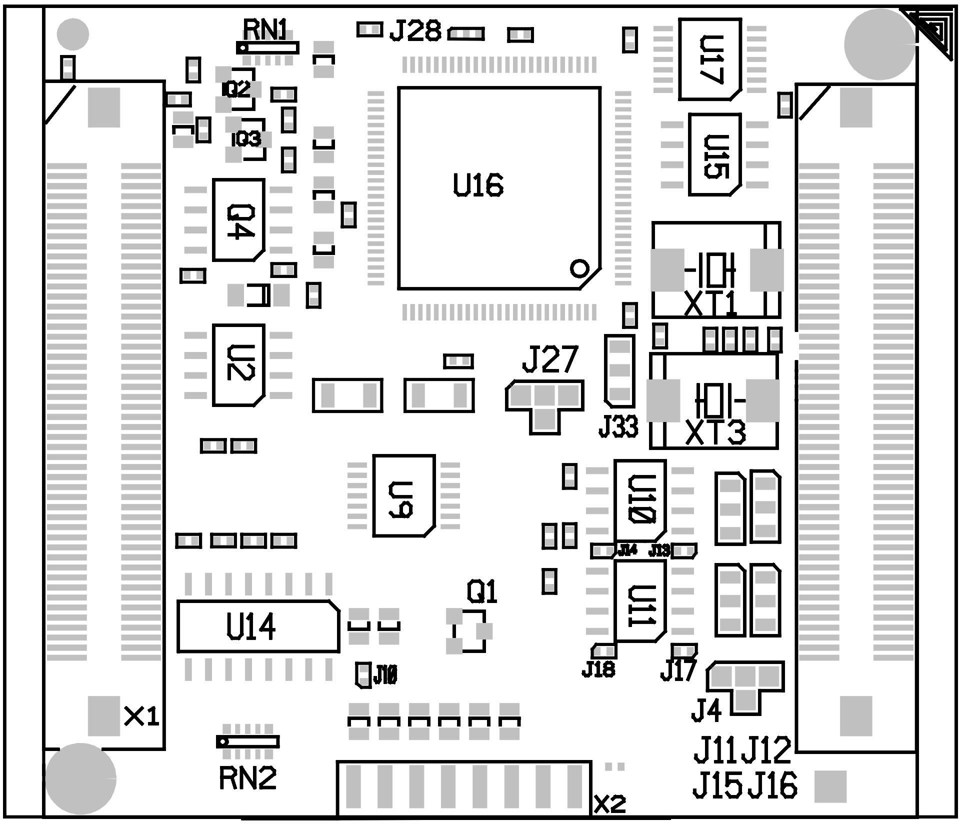


Figure 7: Location of the Jumpers (Connector Side)

The jumpers (J = solder jumper) have the following functions:

	Default Setting[1]		**Alternative Setting**	
J1	(closed)	VAREF derived from supply voltage VCC	(open)	VAREF from external voltage source via pin X1D50
J2	(closed)	VAGND derived from digital ground GND	(open)	VAGND from external ground via pins X1C42, X1C47, X1D39, X1D44 and X1D49[2]
J3	(2 + 3)	external ROM/ Flash active	(1 + 2)	internal ROM/Flash EPROM is active
J4	(1 + 2)	/CS3 from µC selects RAM at U6/U7	(2 + 3) (2 + 4)	/CS1 selects RAM U6/U7 /CS0 selects RAM U6/U7
J5	(1 + 2)	Address line A18 connected with SRAM devices at U6/U7	(2 + 3)	VPD connected with pin 30 on SRAM devices at U6/U7
J6	(1 + 2)	VCC powers E^{2}PROM at U8	(2 + 3)	VPD powers FRAM at U8
J7	(open)	deactivates write protection of the E^{2}PROM memory	(closed)	optional write protection of the E^{2}PROM memory is activated (*see Data Sheet*)
J8	(1 + 2)	address line A1 of the serial memory device at U7 set to low (*see Data Sheet of memory device*)	(2 + 3)	address line A1 of the serial memory device at U7 set to high (*see Data Sheet of memory device*)
J9	(1 + 2)	address line A2 of the serial memory device at U7 set to high (*see Data Sheet of memory device*)	(2 + 3)	address line A2 of the serial memory device at U7 set to low (*see Data Sheet of memory device*)
J10	(open)	Output signal R1O from RS-232 transceiver (pin 12) at U14 disconnected from Molex pin X1D15	(closed)	Output signal R1O from RS-232 transceiver at U14 connects to Molex pin X1D15

[1]: Applies to standard modules without optional features.

[2]: These pins are solely connected to GND of the Development Board when using the phyCORE module on a phyCORE Development Board HD200. It is not possible to attach an external GND potential in this configuration.

	Default Setting		Alternative Setting	
J11	(1 + 2)	CAN0 transmit line (CANTx) of the CAN transceiver at U10 con nected to P9.3 of the microcontroller (*see controller Data Sheet*)	(2 + 3)	CAN0 transmit line (CANTx) of the CAN transceiver at U10 con nected to P4.6 (A22) of the microcontroller (*see controller Data Sheet*)
J12	(1 + 2)	CAN0 receive line (CANRx) of the CAN transceiver at U10 connected to P9.2 of the microcontroller (*see controller Data Sheet*)	(2 + 3)	CAN0 receive line (CANRx) of the CAN transceiver at U10 connected to Port P4.5 (A21) of the microcontroller (*see controller Data Sheet*)
J13	(open)	CAN0_TxD signal connected to on-board CAN transceiver U10	(closed)[1]	CAN0_TxD signal with TTL level available at X1D21 (for use with external transceiver)
J14	(open)	CAN0_RxD signal connected to on-board CAN transceiver U10	(closed)[1]	CAN0_RxD signal with TTL level available at X1D20 (for use with external transceiver)
J15	(1 + 2)	CAN1 transmit line (CANTx) of the CAN transceiver at U11 connected to P9.1 of the microcontroller (*see controller Data Sheet*)	(2 + 3)	CAN1 transmit line (CANTx) of the CAN transceiver at U11 connected to P4.7 (A23) of the microcontroller (*see controller Data Sheet*)
J16	(1 + 2)	CAN1 receive line (CANRx) of the CAN transceiver at U11 connected to P9.0 of the microcontroller (*see controller Data Sheet*)	(2 + 3)	CAN1 receive line (CANRx) of the CAN transceiver at U11 connected to Port P4.4 (A20) of the microcontroller (*see controller Data Sheet*)

[1] : **Note:** Only applicable if on-board CAN transceivers are not populated.

	Default Setting		**Alternative Setting**	
J17	(open)	CAN1_TxD signal connected to on-board CAN transceiver U11	(closed)[1]	CAN1_TxD signal with TTL level available at X1C18 (for use with external transceiver)
J18	(open)	CAN1_RxD signal connected to on-board CAN transceiver U11	(closed)[1]	CAN1_RxD signal with TTL level available at X1D18 (for use with external transceiver)
J19	(closed)	IRQ of the RTC connected to pin P2.11 of the microcontroller	(open)	P2.11 of the controller is freely available as standard I/O at X1D38
J20	(closed)	P9.4 of the microcontroller connected to SDA of the I^2C bus	(open)	P9.4 of the microcontroller is freely available as standard I/O at X1C14
J21	(closed)	P9.5 of the microcontroller connected to SCL of the I^2C bus	(open)	P9.5 of the microcontroller is freely available as standard I/O at X1C15
J22	(1 + 2)	RTC clock output disabled	(2 + 3)	RTC clock output enabled
J23	(closed)[2]	P3.11 used as RxD0 and conected to RS-232 transceiver U14	(open)	P3.11 of the controller is freely available as standard I/O at connector pin X1D16
J24	(closed)[2]	P3.10 used as TxD0 and conected to RS-232 transceiver U14	(open)	P3.10 of the controller is freely available as standard I/O at connector pin X1D17
J25	(2 + 3)	RS-232 transceiver (RxD) of the second serial interface connected to P3.1 of the controller	(open)	P3.1 of the controller is freely available as standard I/O at connector pin X1A44
J26	(2 + 3)	RS-232 transceiver (TxD) of the second serial interface con nected to P3.0 of the controller	(open)	P3.0 of the controller is freely available as standard I/O at connector pin X1A43

[1] : **Note:** Only applicable if on-board CAN transceivers are not populated.

[2] : **Note:** These jumpers must remain closed on the phyCORE-XC161. If they are open, no serial communication is possible, hence PHYTEC FlashTools or the BOOT monitor will not function properly.

	Default Setting		**Alternative Setting**	
J27	(1 + 2)	Chip Select for Ethernet controller connected to /CS2 of the XC161	(2 + 4) (2 + 3)	Chip Select for Ethernet connected to /CS3 Chip Select for Ethernet connected to /CS4
J28	(1 + 2)	/CS_ETH connected to CS8900A Ethernet controller /SBHE input (*see J27*)	(2 + 3)	Address line A0 connected to /SBHE input on the Ethernet controller
J29	(open)	P2.12 of the controller is freely available as standard I/O at X1C38	(closed)	Sleep mode on Ethernet controller controlled with port pin P2.12
J30	(open)	IRQ from Ethernet controller connected to connector pin X1D31	(1 + 2) (2 + 3) (2 + 4)	IRQ from Ethernet controller routed to P2.13 IRQ from Ethernet controller routed to P2.14 IRQ from Ethernet controller routed to P2.15
J32	(open)	Boot sector of Flash at U4 not write protected	(closed)	Flash write protected active (*refer to Flash data sheet*)

Table 2: Jumper Settings

3.1 J1, J2 A/D Reference Voltage

The A/D converter on the phyCORE-XC161 requires an upper and lower reference voltage connected at pins 41 and 42 (V_{AREF}, V_{AGND}). The reference voltage source can be selected using Jumpers J1 and J2.

A/D Reference Voltage Source Selection	**J1**	**J2**
External reference voltage source (V_{AREF} at X1D50, V_{AGND} at X1C42, X1C47, X1D39, X1D44 and X1D49,)	open	open[1]
V_{AREF} derived from voltage supply VCC	closed*	
V_{AGND} derived from digital ground GND potential		closed*

* = Default setting

Table 3: J1, J2 A/D Converter Reference Voltage

3.2 J3 Internal or External Program Memory

At the time of delivery, Jumper J3 is closed at 2+3. This default configuration means that the program stored in the external program memory is executed after a hardware reset. In order to allow the execution of a specific controller's internal program memory, Jumper J3 must be closed at 1+2.

The following configurations are possible:

Code Fetch Selection	**J3**
Execution from external program memory	2 + 3*
Execution from internal program memory	1 + 2

* = Default setting

Table 4: J3 Code Fetch Selection

[1]: These pins are solely connected to GND of the Development Board when using the phyCORE module on a phyCORE Development Board HD200. It is not possible to attach an external GND potential in this configuration.

3.3 J4 SRAM U6/U7 Chip Select

Jumper J4 configures the Chip Select signal for access to the on-board SRAM devices at U6 and U7. The setting of J3 depends on the configuration of the phyCORE-XC161 as shown below.

The following configurations are possible:

Module Configuration	**J4**
Flash, fast SRAM & Ethernet available on the phyCORE module, /CS3 is used to access U6/U7	1 + 2*
no external Flash, /CS0 is used	2 + 4
no external fast SRAM at U5, /CS1 is used	2 + 3

* = Default setting

Table 5: J4 SRAM U6/U7 Chip Select Configuration

3.4 J5 SRAM Memory Size

Jumper J5 configures the size of the SRAM devices installed at U6 and U7. If the phyCORE-XC161 is populated with external SRAM devices with a total capacity of 2*512 kByte, J5 must be closed at position 1+2. This connects of address line A18 of the microcontroller to pin 30 of the SRAM. If an SRAM memory configuration of 2*128 kByte is used, pin 30 on the SRAM must be connected to VPD.

The following configurations are possible:

SRAM U6/U7 Size	**J5**
2 x 512 kB SRAM	1 + 2*
2 x 128 kB SRAM	2 + 3

* = Default setting

Table 6: J5 SRAM U6/U7 Size Configuration

3.5 J6 E²PROM/FRAM Supply Voltage

The device at U8 can be connected to VCC or VPD using Jumper J6. As default, a serial E²PROM populates U8 with voltage supply pins connected to VCC. Alternatively, a serial FRAM device can also populate U8, in order to support frequent write cycles, for instance. If mounted with an FRAM device, the circuit supply pins can be applied to the battery voltage VPD for purposes of data buffering.

The following configurations are possible:

Supply Voltage for U8	**J6**
U8 (E²PROM) supplied with VCC	1 + 2*
U8 (FRAM) supplied with VPD	2 + 3

*= Default setting

Table 7: J6 E²PROM Supply Voltage Configuration

3.6 J7 Write Protection of E²PROM /FRAM

Various types of E²PROM/FRAM can populate space U8. Some of these devices provide a write protection function[1]. Closing Jumper J7 connects pin 7 of the serial E²PROM/FRAM with VCC and thus activates write protection.

The following configurations are possible:

Write Protection E²PROM/FRAM	**J7**
Write protection of E²PROM/FRAM deactivated	open*
Write protection of E²PROM/FRAM activated	closed

* = Default setting

Table 8: J7 Write Protection of E²PROM/FRAM

[1] : *Refer to the corresponding E²PROM/FRAM Data Sheet for more information on the write protection function.*

3.7 J8, J9 Address of the Serial E²PROM/ FRAM

Jumpers J8 and J9 configure the serial E²PROM/FRAM address. The default configuration sets the address to 0xA8.

The following configurations are possible:

E²PROM/FRAM Address	J8	J9
0xA8	1 + 2*	1 + 2*
0xA0	1 + 2	2 + 3
0xA4	2 + 3	2 + 3
0xAC	2 + 3	1 + 2

* = Default setting

Table 9: J8, J9 E²PROM/FRAM Address Configuration

3.8 J10 RS-232 Transceiver R1O TTL Output

Jumper J10 configures the signal available at Molex pin X1D15. It connects the TTL level output signal R1O on the RS-232 transceiver at U14 (pin 12) to the phyCORE-connector X1D15. This TTL level signal can then be used by external devices.

The following configurations are possible:

R1O TTL Level Signal at U14	J10
Disconnected from Molex pin X1D15	open*
Connected to Molex pin X1D15 (Plex)	closed

* = Default setting

Table 10: J10 RS-232 Transceiver R1O TTL Output

3.9 J11, J12, J15, J16 CAN Interfaces

The first CAN interface of the phyCORE-XC161 is available at the port pins P4.5 (CAN0Rx) and P4.6 (CAN0Tx). The second CAN interface is located at port pins P4.4 (CAN1Rx) and P4.7 (CAN1Tx). The XC161 controller also offers a rerouting feature for CAN interface signals to port P9. Using this feature makes the signals of the CAN interface CAN0Rx available at P9.2 and CAN0Tx at P9.3 while CAN1Rx is accessible at P9.0 and CAN1Tx at P9.1.

These signals extend to the two CAN transceivers at U10 and U11 (PCA82C251, alternately TLE6250). The CAN transceivers generate the corresponding CANH0, CANL0, CANH1, and CANL1 signals. These signals can be directly connected to a CAN dual-wire bus. Generation of the CAN signals requires correct setting of solder Jumpers J11, J12, J15 und J16.

Direct access to the CAN1Rx, CAN1Tx, CAN2Rx and CAN2Tx signals is also available at the module’s X1 pin header row if soldering jumpers J11, J12, J15 and J16 are open. This enables use of an external CAN transceiver.

In order to utilize the full 16 MByte linear address space of the microcontroller, the CAN interface signals can be optionally routed to port 9. In this case Jumpers J11, J12, J15 and J16 must be closed at positions 1+2. *Please refer to the Infineon XC161 User’s Manual/Data Sheet for details on the functions and features of controller signals and port pins*.

The following CAN interface configurations are possible:

Interface CAN1	J11	J12
P9.2 (CAN0Rx) P9.3 (CAN0Tx)	1 + 2*	1 + 2*
P4.5 (CAN0Rx) P4.6 (CAN0Tx)	2 + 3	2 + 3

Interface CAN2	J15	J16
P9.0 (CAN1Rx) P9.1 (CAN1Tx)	1 + 2*	1 + 2*
P4.4 (CAN1Rx) P4.7 (CAN1Tx)	2 + 3	2 + 3

* = Default setting

Table 11: J11, J12, J15, J16 CAN Interface Configuration

3.10 J13, J14, J17, J18 CAN Transceiver

Jumpers J13, J14, J17 and J18 are only closed if the CAN transceiver that can populate U10 and U11 are not mounted. In this case the controller’s CAN signals with their TTL level are routed to the phyCORE Connector X1.

Note:
J13, J14, J17 and J18 are configured at time of delivery of the phyCORE module and must not be altered at a later time by the user.

3.11 J19 RTC Interrupt Output

Jumper J19 determines whether the interrupt output of the RTC (U13) is connected to port pin P2.11 of the microcontroller. If Jumper J19 remains open, P2.11 can be used as a port pin at X1D38.

The following configurations are possible:

Port P2.11 Configuration	**J19**
Port P2.11 as /INT input for RTC	closed*
Port P2.11 as I/O pin at X1D38	open

* = Default setting

Table 12: J19 P2.11 / RTC Interrupt Configuration

3.12 J20, J21 Configuration of P9.4 and P9.5 for I²C Bus

The phyCORE-XC161 is equipped with a Real-Time Clock at U13 and a serial E²PROM/FRAM at U8. Both the Real-Time Clock and the serial E²PROM/FRAM are accessed by means of an I²C interface. With Jumpers J20 and J21, this interface can be connected to port pins P9.4 and P9.5. Use of these pins as standard I/O lines requires opening of the corresponding jumpers.

The following configurations are possible:

Port P9.4 and P9.5 Configuration	**J20**	**J21**
Port P9.4 used as I/O pin at X1C14	open	
Port P9.4 used as I²C SDA	closed*	
Port P9.5 used as I/O pin at X1C15		open
Port P9.5 used as I²C SCL		closed*

* = Default setting

Table 13: J20, J21 P9.4, P9.5 / I²C Bus Configuration

3.13 J22 RTC (U13) Clock Output

Jumper J22 can be used to activate the clock output of the RTC populating U13. This clock output is disabled by default. Closing J22 at position 2+3 enables the clock output. *Please refer to the RTC Data Sheet for details on the RTC clock output function*.

The following configurations are possible:

RTC Clock Output	**J22**
RTC_CLKOUT disables	1 + 2*
RTC_CLKOUT enabled and available at connector pin X1D30	2 + 3

* = Default setting

Table 14: J22 RTC Clock Output Configuration

3.14 J23, J24 First Serial Interface

Jumpers J23 and J24 connect the signals of the first asynchronous serial interface to the on-board RS-232 transceiver (U14). The interface signals are then available with RS-232 level at the phyCORE-connector pins X1D22 (RxD0) and X1D23 (TxD0). If the jumpers are opened, the applicable controller pins P3.10 and P3.11 can be used with their alternative functions or the serial interface signals are available with their TTL level at phyCORE-connector pins X1D17 and X1D16.

Note:
These jumpers must remain closed on the phyCORE-XC161. If they are open, no serial communication is possible, hence PHYTEC FlashTools or the BOOT monitor will not function properly.
If the jumpers are closed we recommend **not** to use the interface signals with their TTL level as this will cause damage to the on-board components.

The following configurations are possible:

Signal Quality Serial Interface 1	**J23**	**J24**
TxD0 and RxD0 available with their RS-232 level	closed*	closed*
P3.10 and P3.11 available as I/O pin or TxD0 and RxD0 interface signals with TTL level	open	open

* = Default setting

Table 15: J23, J24 First Serial Interface Configuration

3.15 J25, J26 Second Serial Interface

Jumpers J25 and J26 connect the signals of the second asynchronous serial interface to the on-board RS-232 transceiver (U14). The interface signals are then available with RS-232 level at the phyCORE-connector pins X1C21 (RxD1) and X1C23 (TxD1). If the jumpers are opened, the applicable controller pins P3.0 and P3.1 can be used as standard I/O pins at X1A44 (P3.0) and X1A45 (P3.1).

The following configurations are possible:

Signal Quality Serial Interface 2	**J25**	**J26**
TxD1 and RxD1 available with their RS-232 level	2 + 3*	2 + 3*
P3.0 and P3.1 available as I/O pins at X1A45 and X1A44	open	open
Not allowed!	*1 + 2*	*1 + 2*

* = Default setting

Table 16: J25, J26 Second Serial Interface Configuration

3.16 J27 Ethernet Controller Chip Select

Jumper J27 configures the source of the Chip Select signal that controls the Ethernet controller. Configuration of J27 also depends on the module configuration of the phyCORE-XC161.

The following configurations are possible:

Chip Select for Ethernet Controller	**J27**
/CS2 connected to Ethernet Controller U16	1 + 2*
/CS4 connected to Ethernet Controller U16	2 + 3
/CS3 connected to Ethernet Controller U16	2 + 4

* = Default setting

Table 17: J27 Ethernet Controller Chip Select Configuration

3.17 J28 Ethernet Controller /SBHE Configuration

Jumper J28 configures which signal connects to the /SBHE input of the Ethernet controller at U16. If J28 is closed at position 1+2 configuration of Jumper J27 (*refer to section 3.16*) defines which of the three available /CS signals from the microcontroller extends to the /SBHE input.

The following configurations are possible:

/SBHE Input Configuration	**J28**
Chip Select from the XC161 connected to /SBHE input, *refer to J27*	1 + 2*
Address line A0 connected to /SBHE input	2 + 3

* = Default setting

Table 18: J28 Ethernet Controller /SBHE Configuration

3.18 J29 Ethernet Controller Sleep Mode

Closing Jumper J29 connects the microcontroller port pin P2.12 to the Ethernet controller. This port pin can then be used to render the Ethernet controller in sleep mode.

The following configurations are possible:

Ethernet Controller Sleep Mode	**J29**
Sleep mode can not be activated	open*
Sleep mode can be activated via port P2.12	closed

* = Default setting

Table 19: J29 Ethernet Controller Sleep Mode Configuration

3.19 J30 Ethernet Controller IRQ Signal

Jumper J33 connects the IRQ output of the Ethernet controller with port pins P2.13, P2.14 or P2.15 of the microcontroller. If these port pins are not used as Ethernet IRQ, they are available as standard I/O signal at phyCORE-connector pins X1D37, X1D25 and X1D26.

The following configurations are possible:

IRQ Signal of the Ethernet controller	**J30**
P2.13, P2.14 and P2.15 available as I/O pins at X1D37, X1D25 und X1D26	open*
P2.13 connected to IRQ_ETH	1 + 2
P2.14 connected to IRQ_ETH	2 + 3
P2.15 connected to IRQ_ETH	2 + 4

* = Default setting

Table 20: J30 Ethernet Controller IRQ Signal Configuration

3.20 J32 Write Protection of Flash

Various types of Flash can populate space U4. Some of these devices provide a write protection function[1] for the Flash boot sector. Closing Jumper J32 connects pin 14 of the Flash with GND and thus activates write protection.

The following configurations are possible:

Write Protection Flash	**J32**
Write protection of Flash deactivated	open*
Write protection of Flash activated	closed

* = Default setting

Table 21: J32 Write Protection of Flash

[1] *Refer to the corresponding Flash Data Sheet for more information on the write protection function.*

4 System Configuration

Following a hardware or software reset, the microcontroller starts program execution from address 00:0000H. At this address a jump instruction to an application-specific initialization routine is located. This routine configures certain features of the microcontroller. Initialization is carried out in a privileged mode and completed by an EINIT instruction. After that, access to specific registers and execution of certain instructions are limited.

Although most features of the XC161 microcontroller are configured and/or programmed during the initialization routine, other features, which influence program execution, must be configured prior to initialization.

4.1 System Startup Configuration

The system startup configuration sets the features of the microcontroller that have a direct influence on program execution and, hence, the correct execution of the initialization routine as well. Of particular importance to the system startup configuration are the characteristics of the external bus interface which supports the module's memory (for example data width, multiplexed- or demultiplexed mode).

During the system startup configuration, certain pins comprising port P0 are latched by the controller during the reset procedure. The signal level on the corresponding input pins configures the resulting characteristics of the controller. The system startup configuration can be set by connecting desired pins at port 0 with a pull-down resistor (resulting in logical 0), or by leaving the connections open (resulting in logical 1).

A 4.7 kΩ pull-down resistor is recommended, although the resistor value is also dependent upon the external circuitry that is connected to the data bus of the module.

The individual pins of port P0 have the following functions:

Function of port P0 during system reset (high byte)							
Bit H7	H6	H5	H4	H3	H2	H1	Bit H0
CLKCFG			SALSEL		CSSEL		WRC
R10,	*R9,*	*R8*	*R7*	*R6*	*R5*	*R4*	
0	*0*	*1*	*0(1[1])*	*0*	*1*	*1*	1

Function of port P0 during system reset (low byte)							
Bit L7	L6	L5	L4	L3	L2	L1	Bit L0
BUSTYP		SMOD				ADP	EMU
	R3		*R2*				
1	*0*	*Pin 21B*	*0*	*Pin 20B*	*Pin 20A*	*Pin 19A*	*Pin 18B*

Table 22: Functional Settings on Port P0 for System Startup Configuration

In order to ensure proper functioning of the microcontroller, reserved pins must remain at high-level (logical 1).
Configuration on these pins must not be changed.

[1]: On modules with a memory configuration featuring 2 MByte Flash memory the register SALSEL must be configured with the values 1 (H4) 0 (H3).

The following table provides detailed comments to these system startup functions:

Name	**Value**	**Function**	**Comment**
CLKCFG	0 0 0	CPU clock = ext. clock * 0.5	defines CPU clock
	0 0 1*	CPU clock = ext. clock * 2.5	
	0 1 0	CPU clock = ext. clock * 2.5	
	0 1 1	CPU clock = ext. clock * 1	
	1 0 0	CPU clock = ext. clock * 5	
	1 0 1	CPU clock = ext. clock * 2	
	1 1 0	CPU clock = ext. clock * 4.5	
	1 1 1	CPU clock = ext. clock * 3	
SALSEL	0 0	address lines, A16..A19, I/O pins P4.4..P4.7	defines function of port pins P4.0..P4.7
	0 1	no address lines, I/O pins P4.0..P4.7	
	1 0[1]	address lines A16..A23, no I/O pins	
	1 1	address lines A16..A17, I/O pins P4.2..P4.7	
CSSEL	0 0	Chip Selects /CS0../CS2, I/O pins P6.3..P6.4	defines function of port pins P6.0..P6.4
	0 1	Chip Selects /CS0../CS1, I/O pins P6.2..P6.4	
	1 0	no Chip Selects, I/O pins P6.0..P6.4	
	1 1	Chip Selects /CS0../CS4, no I/O pins	
WRC	0	/WRL and /WRH	defines function of pins /WR and P3.12
	1	/WR and /BHE	

[1]: On modules with a memory configuration featuring 2 MByte Flash memory the register SALSEL must be configured with the values 1 (H4) 0 (H3).

Name	Value	Function	Comment
BUSTYP	0 0	8-bit demultiplexed bus	defines bus interface for /CS0
	0 1	8-bit multiplexed bus	
	1 0	16-bit demultiplexed bus	
	1 1	16-bit multiplexed bus	
BSL	0	Bootstrap loader active	
	1	Bootstrap loader inactive	
ADP	0	adapter mode active	
	1	adapter mode inactive	
EMU	0	emulation mode active	
	1	emulation mode inactive	

Table 23: System Startup Configuration Registers

Default system startup configuration of the phyCORE-XC161

The initial setting of the system startup configuration can be modified during the initialization routine. Certain functions can not be configured during startup, such as selection of the number of wait states for individual memory devices and Chip Select signals, as well as the location of these devices within the controller’s address space.

Several software development tools utilize a special file which allows easy definition of system settings. This configuration file can be easily included in the translation and link procedures (such as the ***start_v2.a66*** used within the Keil software development tool chain).

5 Memory Models

The XC161 controller provides up to five Chip Select signals at port P6 for easy selection of external peripherals or memory banks. Depending on the number of memory devices installed on the phyCORE-XC161, as well as the availability of the Ethernet controller, up to four Chip Select signals are used internally. /CS0 (P6.0) selects the Flash memory installed on U4 with a total memory of either 256 kByte, 512 kByte, 1 MByte or 2 MByte. The external data memory consists of a fast 512 kByte SRAM at U5 and two RAM banks at U6 and U7. Spaces U6 and U7 can house memory devices of 512 kByte in an SO44 package.

/CS1 (P6.1) selects the fast SRAM on U5 while either /CS0 (P6.0), /CS1 (P6.1) or /CS3 (P6.3) selects the RAM banks at U6/U7. See description of Jumper J4 (*section 3.3*) for details. If the Ethernet controller populates the module at U16 then either /CS2 (P6.2), /CS3 (P6.3) or /CS4 (P6.4) can be used to control this device. See description of Jumper J27 (*section 3.16*) for details.

The Chip Select signals must be enabled during reset (*refer to section 4*). The assignment of the Chip Select signals to specific address areas is done with the corresponding ADDRESELx, FCONCSx and TCONCSx register. Note that ADDRESELx must be configured prior activating of the Chip Select signal with register FCONCSx. Ensure that the memory areas do not overlap in order to avoid conflicts when accessing the desired code or data memory. Program code must remain accessible via /CS0.

Prior to definition of the ADDRESELx, FCONCSx and the TCONCSx register, only /CS0 (P6.0 connected to Flash at U4) is active in the entire address space and remains active for all areas not assigned to another Chip Select signal.

The timing of the bus access is controlled by the Timing CONfiguration registers for CSx (TCONCSx), which specify the timing of the bus cycle with the lengths of the different access phases.

The following paragraph contains important information on timing characteristics. All information refers to a XC161 controller with a 16-bit bus, demultiplexed, at 40 MHz CPU clock time (F_{osz}), 25 ns clock cylce:

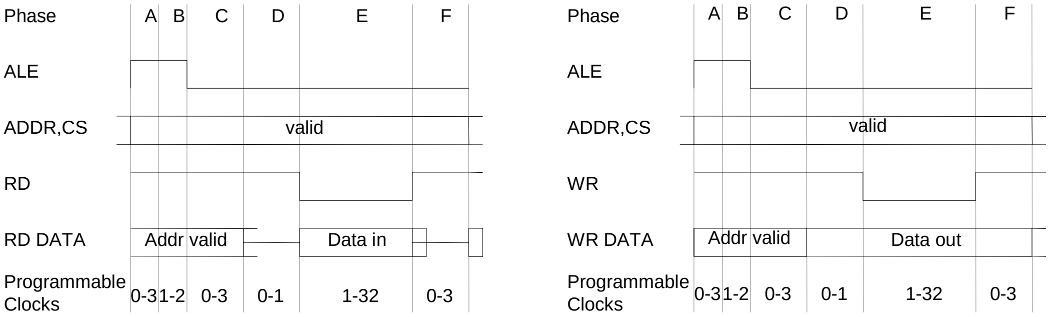


Figure 8: Timing Phases for Multiplexed Bus Mode

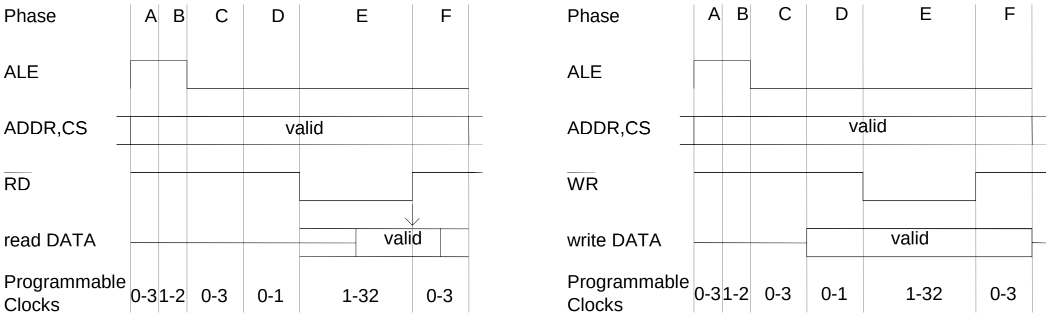


Figure 9: Timing Phases for Demultiplexed Bus Mode

Bus timing calculation

The worst case scenario as described by the individual memory or other peripheral component manufacturer should be assumed when calculating the bus timing phases. It should also be noted that applicable controller signals are not always present on the XC161 outputs at the same time. Signal transition times (H-to-L or L-to-H) of up to 4 ns have to be considered. The H-to-L transitions delay for /CS, /RD and /WR signals can be as long as 10 ns while the rise time (L-to-H) for the same signals can be up to 6 ns.

When using the module at 40 MHz with demultiplexed bus mode (25 ns clock cylce) the fastest possible timing can be achieved with one clock cylce for phase B and one cycle for phase E. The total of two cylces results in a 50 ns instruction cycle.

The maximum delay time for the /RD signal is 11 ns. Then the memory device needs 4 ns to apply its data to the bus. Now the data have to be valid on the bus for at least 15 ns before they can be read from the controller. Based on the values above phase E calculates to a minimum length of 11 ns + 4 ns + 15 ns = 30 ns. Configuration of two clock cyles (25 ns each) for phase E is necessary to achieve proper access.

Examples:

Demultiplexed bus at 40 MHz

Memory	**TCONCSx**	**PHA**	**PHB**	**PHC**	**PHD**	**PHE**	**RDPHF**	**WRPHF**
SRAM 15ns, U5	0x0040	0	1	0	0	2	0	0
Flash 70ns, U4	0x0080	0	1	0	0	3	0	0
SRAM 70ns, U6/7	0x0080	0	1	0	0	3	0	0
Ethernet at U16	0x240	0	1	0	0	10	0	0

Table 24: Bus Timing Calculation Examples

5.1 Bus Cycle Phases

Phase A – Chip Select Change Phase

The A phase can take 0-3 clocks. It is used for tristating data bus drivers from the previous cycle (tristate wait states after chip select switch). A phase cycles are not inserted at every access cycle, but only when changing the CS. If an access using one CS (CSx) ends and the next access with a different CS (CSy) is started, then A phase cycles are performed according to the bits set in the **first** CS (CSx). This feature is used to optimize wait states with devices having a long turn-off delay at their databus drivers, such as EPROMs and flash memories. **The A phase cycles are inserted while the addresses and ALE of the next cycle are already applied.** If there are some idle cycles between two accesses, these clocks are taken into account and the A phase is shortened accordingly. For example, if there are three tristate cycles programmed and two idle cycles occur, then the A phase takes only one clock.

Phase B - Address Setup / ALE Phase

The B phase can take 1-2 clocks. It is used for addressing devices before giving a command, and defines the length of time that ALE is active. In multiplexed bus mode, the address is applied for latching.

Phase C - Delay Phase

The C phase is similar to the A and B phases but ALE is already low. It can take 0-3 clocks. In multiplexed bus mode, the address is held in order to be latched safely. Phase C cycles can be used to delay the command signals (R/W delay).

Phase D - Write Data Setup / Mux Tristate Phase

The D phase can take 0-1 clocks. It is used to tristate the address on the multiplexed bus when a read cycle is performed. For all write cycles, it is used to ensure that the data are valid on the bus before the command is applied.

Phase E - RD/WR Commando Phase

The E phase is the command or access phase, and takes 1-32 clocks. Read data are fetched, write data are put onto the bus, and the command signals are active. Read data are registered with the terminating clock of this phase. The READY function lengthens this phase, too. READY-controlled access cycles may have an unlimited cycle time.

Phase F - Address / Write Data Hold Phase

The F phase is at the end of an access. It can take 0-3 clocks. Addresses and write data are held while the command is inactive. The number of wait states inserted during the F phase is independently programmable for read and write accesses. The F phase is used to program tristate wait states on the bidirectional data bus in order to avoid bus conflicts.

6 Serial Interfaces

6.1 RS-232 Interface

One RS-232 transceiver is located on the phyCORE-XC161 at U14. This device converts the signal levels for the P3.11/RxD0 and P3.10/TxD0 lines, as well as those of the second serial interface, P3.1/RxD1 and P3.0/TxD1 from TTL level to RS-232 level.

The RS-232 interface enables connection of the module to a COM port on a host-PC. In this instance the RxD0 line of the transceiver is connected to the TxD line of the COM port; while the TxD0 line is connected to the RxD line of the COM port. The Ground potential of the phyCORE-XC161 circuitry needs to be connected to the applicable Ground pin on the COM port as well.

The microcontroller's on-chip UART does not support handshake signal communication. However, depending on user needs, handshake communication can be software emulated using port pins on the microcontroller. Use of an RS-232 signal level in support of handshake communication requires use of an external RS-232 transceiver not located on the module.

Note:
Jumpers J23 and J24 must remain closed on the phyCORE-XC161. If they are open, no serial communication is possible, hence PHYTEC FlashTools or the BOOT monitor will not function properly.

6.2 CAN Interface

The phyCORE-XC161 is designed to house two CAN transceivers at U10 and U11 (either PCA82C251 or TLE6250). The CAN bus transceiver devices support signal conversion of the CAN transmit (CANTx) and receive (CANRx)lines. The CAN transceiver supports up to 110 nodes on a single CAN bus. Data transmission occurs with differential signals between CANH and CANL. A Ground connection between nodes on a CAN bus is not required, yet is recommended to better protect the network from electromagnetic interference (EMI). In order to ensure proper message transmission via the CAN bus, a 120 Ohm termination resistor must be connected to each end of the CAN bus.

For larger CAN bus systems, an external opto-coupler should be implemented to galvanically separate the CAN transceiver and the phyCORE-XC161. This requires the CANTx and CANRx lines to be separated from the on-board CAN transceivers by opening Jumpers J11, J12, J15, and J16. For connection of the CANTx and CANRx lines to an external transceiver we recommend using a Hewlett Packard HCPL06xx or a Toshiba TLP113 HCPL06xx fast opto-coupler. Parameters for configuring a proper CAN bus system can be found in the DS102 norms from the CiA[1] (CAN in Automation) User and Manufacturer’s Interest Group.

[1]: CiA: CAN in Automation. Founded in March 1992, CiA provides technical, product and marketing information with the aim of fostering Controller Area Network’s image and providing a path for future developments of the CAN protocol.

7 Real-Time Clock RTC-8564 (U13)

For real-time or time-driven applications, the phyCORE-XC161 is equipped with an RTC-8564 Real-Time Clock at U13. This RTC device provides the following features:

- Serial input/output bus (I^2C)
- Power consumption

Bus active:	max. 50 mA
Bus inactive, CLKOUT = 32 kHz :	max. 1.7 µA
Bus inactive, CLKOUT = 0 kHz :	max. 0.75 µA

- Clock function with four year calendar
- Century bit for year 2000-compliance
- Universal timer with alarm and overflow indication
- 24-hour format
- Automatic word address incrementing
- Programmable alarm, timer and interrupt functions

If the phyCORE-XC161 is buffered by battery, the Real-Time Clock runs independently of the board’s power supply.

Programming the Real-Time Clock is done via the I^2C bus (address 0xA2 = 10100010), which is connected to port P9.5 (SCL) and port P9.4 (SDA). The Real-Time Clock also provides an interrupt output that extends to port P2.11 via Jumper J19. An interrupt occurs in case of a clock alarm, timer alarm, timer overflow and event counter alarm. An interrupt must be cleared by software. With the interrupt function, the Real-Time Clock can be utilized in various applications. *For more information on the features of the RTC-8564, refer to the corresponding Data Sheet.*

Note:
After connection of the supply voltage, or after a reset, the Real-Time Clock generates **no** interrupt. The RTC must first be initialized (*see RTC Data Sheet for more information*).

8 Serial E^{2}PROM/FRAM (U8)

The phyCORE-XC161 is populated with a non-volatile memory with a serial interface (I^{2}C interface) to store configuration data. According to the memory configuration of the module, an E^{2}PROM (4 to 32 kByte) or FRAM can be mounted at U8.

A description of the I^{2}C memory protocol of the specific memory component at U8 can be found in the respective Data Sheet.

Table 25 gives an overview of the memory components that can be used at U8 at the time of printing of this manual.

Device Type	Manufacturer	Size	Component
E^{2}PROM	Microchip	32 kByte	MIC24LC256
	Catalyst	4 kByte	CAT24WC32
		8 kByte	CAT24WC64
	ST	4 kByte	M24C32
		8 kByte	M24C64
FRAM	Ramtron	512 Byte	FM24C04
		8 kByte	FM24C64

Table 25: Memory Device Options for U8

Various available E^{2}PROM/FRAM types provide a write protection function[1]. Jumper J7 is used to activate this function. If this jumper is closed, then pin 7 of the serial E^{2}PROM/FRAM is connected to VCC.

Write Protection E^{2}PROM/FRAM	J7
Write protection of E^{2}PROM/FRAM deactivated	open*
Write protection of E^{2}PROM/FRAM activated	closed

* = Default setting

Table 26: E^{2}PROM/FRAM Write Protection

Jumpers J8 and J9 configure the address of the serial E²PROM/FRAM. The default configuration sets the address to 0xA8.

E^2PROM/FRAM Address	**J8**	**J9**
0xA8	1 + 2*	1 + 2*
0xA0	1 + 2	2 + 3
0xA4	2 + 3	2 + 3
0xAC	2 + 3	1 + 2

* = Default setting

Table 27: E²PROM/FRAM Address

[1] *Refer to the corresponding E²PROM/FRAM Data Sheet for more information on the write protection function.*

9 Flash Memory (U4)

Use of Flash as non-volatile memory on the phyCORE-XC161 provides an easily reprogrammable means of code storage. The following Flash devices can populate the phyCORE-XC161:

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

These Flash devices are programmable with 5 V. No dedicated programming voltage is required.

Use of a Flash device as the only code memory results in no or only a limited usability of the Flash memory as non-volatile memory for data. This is due to the internal structure of the Flash device as, during the Flash-internal programming process, the reading of data from Flash is not possible. Hence, for Flash programming, program execution must be transferred out of Flash (such as into von Neumann RAM). This usually equals the interruption of a "normal" program execution cycle.

As of the printing of this manual, Flash devices generally have a life expectancy of at least 100,000 erase/program cycles.

10 Battery Buffer and Voltage Supervisor Chip (U9)

The battery that buffers the memory is not essential to the functioning of the phyCORE-XC161. However, this battery buffer embodies an economical and practical means of storing non-volatile data. It is necessary to preserve data from the Real-Time Clock in case of a power failure.

The VBAT input at pin X1C6 of the board is provided for connecting the external battery. The negative polarity pin on the battery must be connected to GND on the phyCORE-XC161. As of the printing of this manual, a lithium battery is recommended as it offers relatively high capacity at low discharge. In the event of a power failure at VCC, the RTC will be buffered by a connected battery via VBAT. The RTC device is generally supplied via VPD in order to preserve data by means of the battery back-up in the absence of a power supply via VCC.

The Voltage Supervisor Chip populating U9 controls switching between VCC supply and the back-up battery. *The basic characteristics of this IC are described in the appropriate Data Sheet, which is available on the Spectrum CD.*

11 CS8900A Ethernet Controller

11.1 Fundamentals

The CS8900A is a IEEE 802.3 Single-Chip Ethernet-Controller that is operated in memory mode on the phyCORE-XC161. The configuration data for the Ethernet controller are stored in a E²PROM located at U22.

The CS8900A Ethernet controller provides the following features:

- power consumption: 55 mA
- industrial temperature range available (CS8900A-IQ)
- I/O- and memory mode
- Full-Duplex operation
- On-chip RAM buffer for transmit and receive frames
- 10 Base-T Port with analog filters (automatic polarization recognition and correction)
- AUI-Port for 10Base2, 10Base5 and 10Base-F
- LED driver for LINK status and LAN activity
- Sleep mode

Additional technical data for the CS8900A Ethernet controller are available in the corresponding data sheet.

11.2 Memory Mode

Following a hardware reset the CS8900A Ethernet controller is in I/O mode. In order to render the chip into memory mode the following settings are required in the E²PROM U15:

register 0116h bit A = 1

The base address of the CS8900A is set in register 002Ch. It is recommended to configure the value "0000 0000h".

12 Debug Interface

The phyCORE-XC161 is equipped with a JTAG interface for downloading program code into the external Flash or for debugging programs in the external SRAM. The JTAG interface extends to 2 mm pitch pin header pads at X2 located on the controller side of the module. *Figure 10* and *Figure 11* show the position of the debug interface (JTAG connector X2) on the phyCORE module.

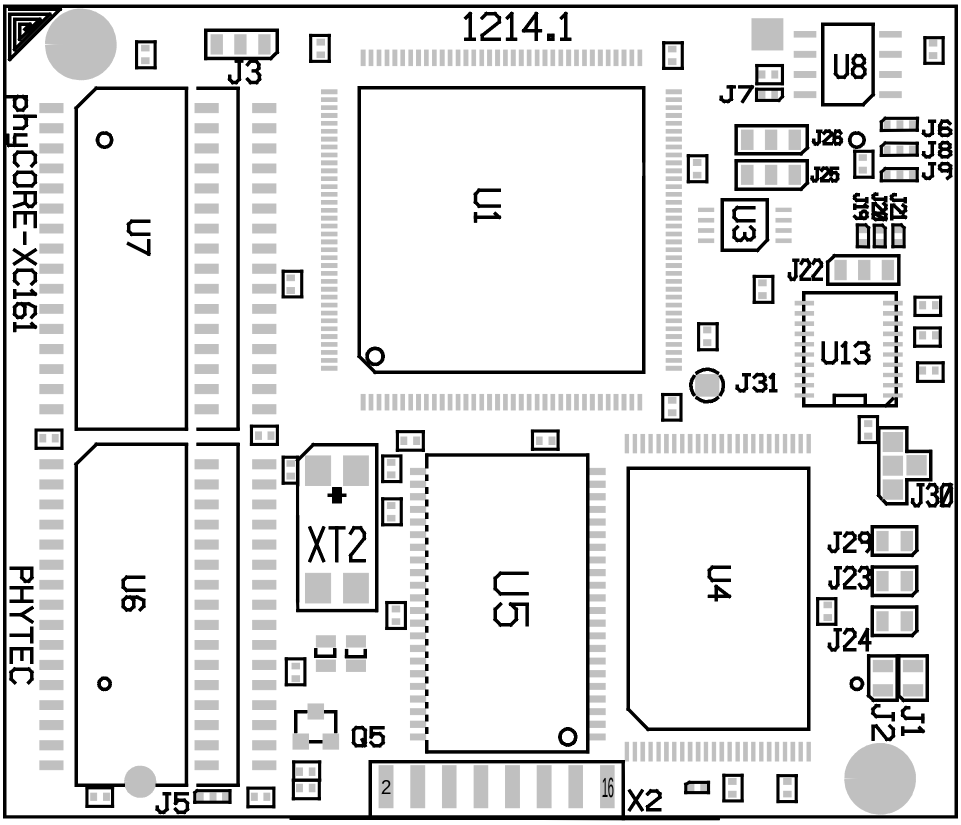


Figure 10: JTAG Interface (Top View)

Pin 1 of the JTAG connector X2 is marked by a black pad on the connector side of the PCB.

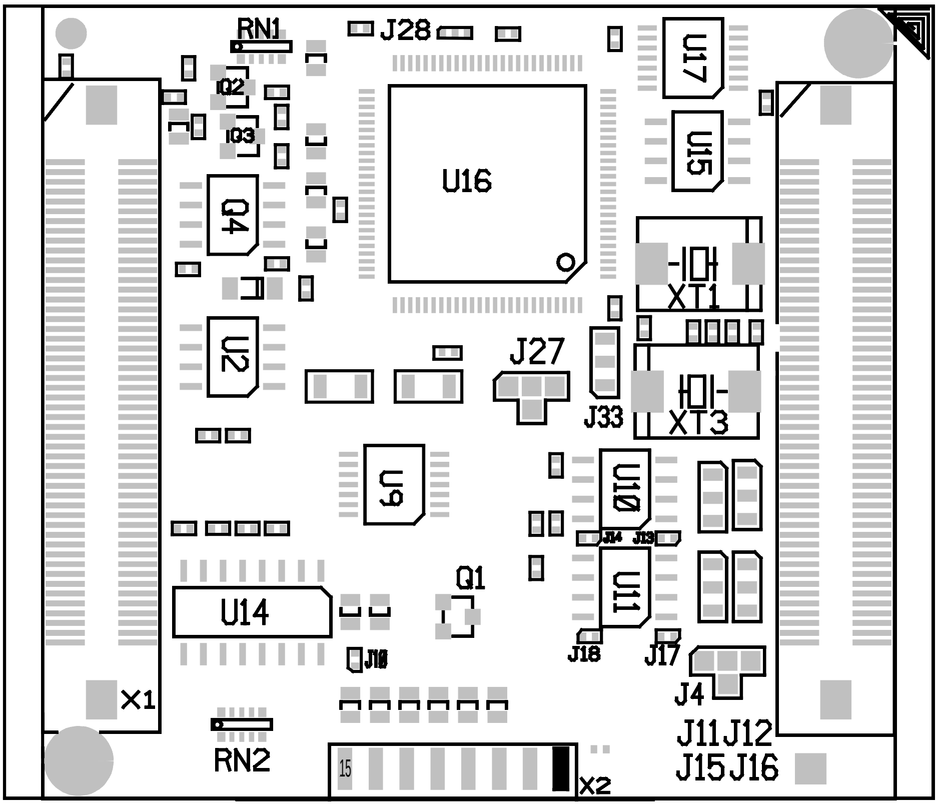
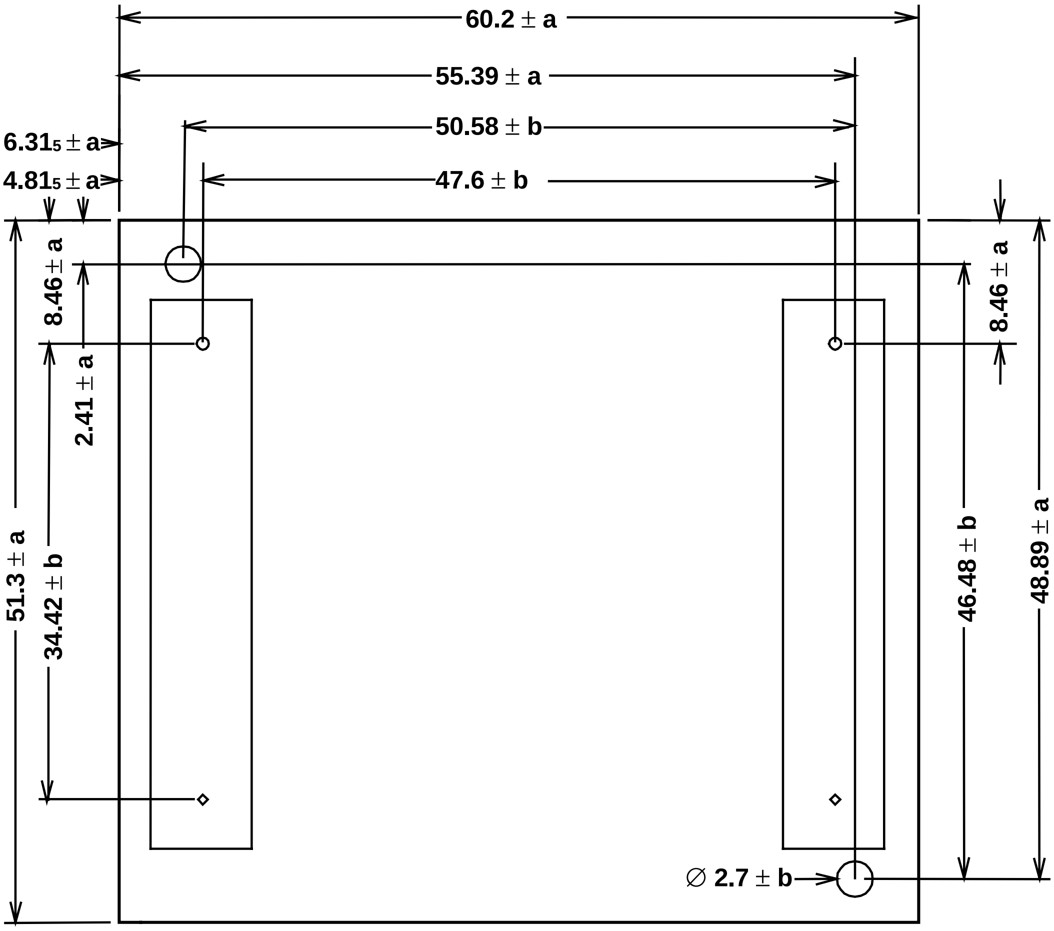


Figure 11: JTAG Interface (Bottom View)

Note:
The JTAG connector X2 is usually not populated on standard versions of the phyCORE-XC161 modules since they are intended for OEM implementation. The applicable pin header connector for X2 is available through PHYTEC (order code VL094) and included in all Rapid Development Kits (order code KPCM-020). In addition, all JTAG signals are also accessible at the phyCORE-connector X1 (Molex connectors). We recommend integration of a standard (2.54 mm pitch) pin header connector in the user target circuitry to allow easy program updates via the JTAG interface. *See **Fehler! Verweisquelle konnte nicht gefunden werden.** for details on the JTAG signal pin assignment.* Additional accessories for connecting the JTAG interface to third party OCDS tools (e.g. Keil ULINK device, HITEX JProbeXC, etc.) are also available through PHYTEC. Please contact us for details.

13 Technical Specifications

The physical dimensions of the phyCORE-XC161 are represented in *Figure 12*. The module's profile is ca. 6 mm thick, with a maximum component height of 2.0 mm on the backside of the PCB and approximately 2.5 mm on the front side. The board itself is approximately 1.6 mm thick.



Tolerance	a	b
in [mm]	0.20	0.05

Figure 12: Physical Dimensions

© PHYTEC Meßtechnik GmbH 2004

Additional specifications:

•	Dimensions:	60 mm x 53 mm
•	Weight:	approximately 25 g with all optional components mounted on the circuit board
•	Storage temperature:	-40°C to +90°C
•	Operating temperature:	standard: 0°C to +70°C extended: -40°C to +85°C
•	Humidity:	95 % r.F. not condensed
•	Operating voltage:	5 V ±5 %, VBAT 3 V ±20 %, 2.5 V ±5%
•	Power consumption:	Conditions:
	maximum 220 mA	**VCC = 5 V, VBAT = 0 V**,
	typical 110 mA	256 kByte RAM, 5 MHz quartz, 20°C
	maximum 100 µA	**VCC = 0 V, VBAT = 3 V**,
	typ. 1 µA Real-Time Clock	20°C

These specifications describe the standard configuration of the phyCORE-XC161 as of the printing of this manual.

14 Hints for Handling the phyCORE-XC161

All XC161 compatible controllers can populate the phyCORE-XC161 module at U1. Please note that, if using a XC161 derivative with an active CAN interface at port 4, only 20 external address lines (A0…A19) and 1 MByte of address space is available on the module. These constraints can be avoided by relocating the CAN interface to port 9[1] (*see controller User's Manual and Data Sheet for details*).

In order to activate the address lines A18…A23 (for more than 256 kByte Flash) the configuration resistors at data lines D12 and D11 of the module must be pulled to GND level (*see section 4, "System Configuration")*.

The address and data bus on the module is not buffered. To connect external components to the data/address bus, as well as the control lines (/RD, /WR), an external buffer (i.e. 74AHCT245) between the modul and the peripheral components should be installed.

The data bus D0…15 (Port 0) should be connected with a 100 kΩ pull-up resistor against VCC. Furthermore, precautions should be taken to allow connection of configuration resistor against GND directly to port 0 (pin 0…15). This enables startup of the XC161 in various configurations since these specific pins are latched during reset (*see controller User's Manual and section 4, "System Configuration"*).

The /NMI input is connected with a pull-up resistor (10 kΩ) against VCC. This enables activation of the NMI signal by means of a high-low signal transition. This can be realized with a push button (switching to GND) and is useful during software development if e.g. a Monitor program is used (*see Monitor User's Manual*).

[1] This function is only available with Infineon's XC161 microcontroller.

Removal of various components, such as the microcontroller and the standard quartz, is not advisable given the compact nature of the module. Should this nonetheless be necessary, please ensure that the board as well as surrounding components and sockets remain undamaged while desoldering. Overheating the board can cause the solder pads to loosen, rendering the module inoperable. Carefully heat neighboring connections in pairs. After a few alternations, components can be removed with the solder-iron tip. Alternatively, a hot air gun can be used to heat and loosen the bonds.

15 The phyCORE-XC161 on the phyCORE Development Board HD200 XC

PHYTEC Development Boards are fully equipped with all mechanical and electrical components necessary for the speedy and secure start-up and subsequent communication to and programming of applicable PHYTEC Single Board Computer (SBC) modules. Development Boards are designed for evaluation, testing and prototyping of PHYTEC Single Board Computers in labratory environments prior to their use in customer designed applications.

15.1 Concept of the phyCORE Development Board HD200

The phyCORE Development Board HD200 XC provides a flexible development platform enabling quick and easy start-up and subsequent programming of the phyCORE-XC161 Single Board Computer module. The Development Board design allows easy connection of additional expansion boards featuring various functions that support fast and convenient prototyping and software evaluation.

This modular development platform concept is depicted in *Figure 13* and includes the following components:

- The actual **Development Board** (1), which offers all essential components and connectors for start-up including: a power socket enabling connection to an **external power adapter** (2) and **serial interfaces** (3) of the SBC module at DB-9 connectors (depending on the module, up to two RS-232 interfaces and up to two RS-485 or CAN interfaces).
- All of the signals from the SBC module mounted on the Development Board extend to two mating receptacle connectors. A strict 1:1 signal assignment is consequently maintained from the phyCORE-connectors on the module to these expansion connectors. Accordingly, the pin assignment of the **expansion bus** (4) depends entirely on the pinout of the SBC module mounted on the Development Board.

- As the physical layout of the expansion bus is standardized across all applicable PHYTEC Development Boards, we are able to offer various **expansion boards** (5) that attach to the Development Board at the expansion bus connectors. These modular expansion boards offer **supplemental I/O functions** (6) as well as peripheral support devices for specific functions offered by the controller populating the **SBC module** (9) mounted on the Development Board.
- All controller and on-board signals provided by the SBC module mounted on the Development Board are broken out 1:1 to the expansion board by means of its **patch field** (7). The required connections between SBC module / Development Board and the expansion board are made using **patch cables** (8) included with the expansion board.

Figure 13 illustrates the modular development platform concept:

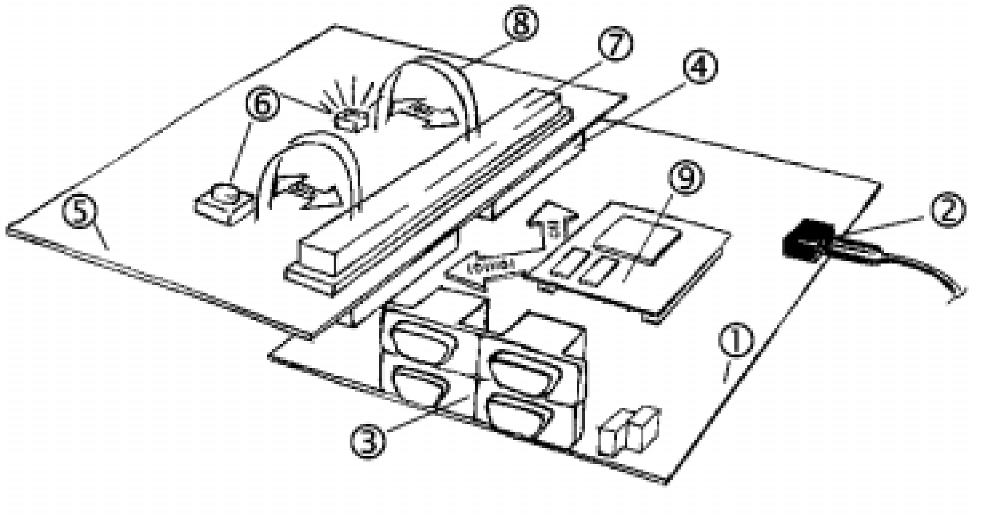


Figure 13: Modular Development and Expansion Board Concept with the phyCORE-XC161

The following sections contain specific information relevant to the operation of the phyCORE-XC161 mounted on the phyCORE Development Board HD200 XC. For a general description of the Development Board, *please refer to the corresponding Development Board Hardware Manual.*

15.2 Development Board HD200 XC Connectors and Jumpers

15.2.1 Connectors

As shown in *Figure 14*, the following connectors are available on the phyCORE Development Board HD200 XC:

X1- low-voltage socket for power supply connectivity
X2- mating receptacle for expansion board connectivity
P1- dual DB-9 sockets for serial RS-232 interface connectivity
P2- dual DB-9 connectors for CAN or RS-485 interface connectivity
X4- voltage supply for external devices and subassemblies
X5- GND connector (for connection of GND signal of measuring devices such as an oscilliscope)
X6- phyCORE-connector enabling mounting of applicable phyCORE modules
X7 - connector for Ethernet transformer module EAD-001
U9/U10- space for an optional silicon serial number chip
BAT1- receptacle for an optional battery

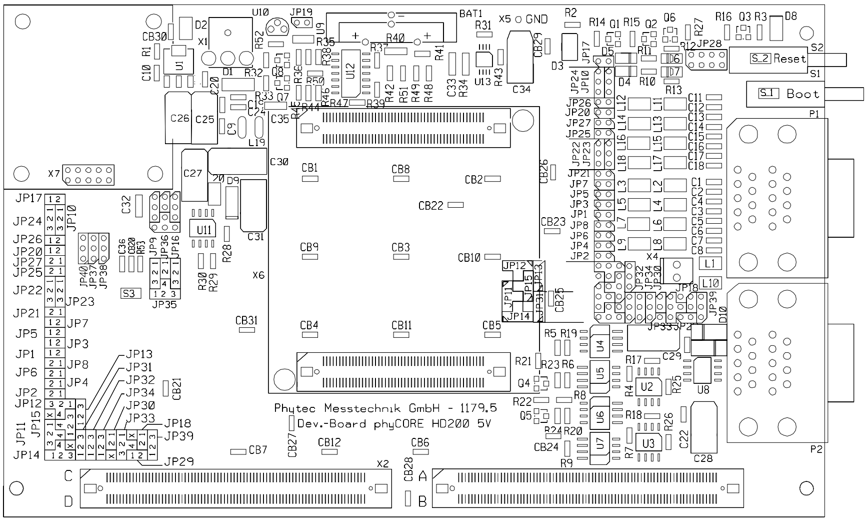


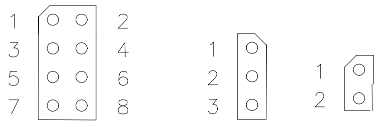
Figure 14: Location of Connectors on the phyCORE Development Board HD200 XC

Please note that all module connections are not to exceed their expressed maximum voltage or current. Maximum signal input values are indicated in the corresponding controller User’s Manual/Data Sheets. As damage from improper connections varies according to use and application, it is the user's responsibility to take appropriate safety measures to ensure that the module connections are protected from overloading through connected peripherals.

15.2.2 Jumpers on the phyCORE Development Board HD200 XC

Peripheral components of the phyCORE Development Board HD200 XC can be connected to the signals of the phyCORE-XC161 by setting the applicable jumpers.

The Development Board's peripheral components are configured for use with the phyCORE-XC161 by means of insertable jumpers. If no jumpers are set, no signals connect to the DB-9 connectors, the control and display units and the CAN transceivers. The Reset input on the phyCORE-XC161 directly connects to the Reset button (S2). *Figure 15* illustrates the numbering of the jumper pads, while *Figure 16* indicates the location of the jumpers on the Development Board.



z.B.: JP28 z.B.: JP23 z.B.: JP24

Figure 15: Numbering of Jumper Pads

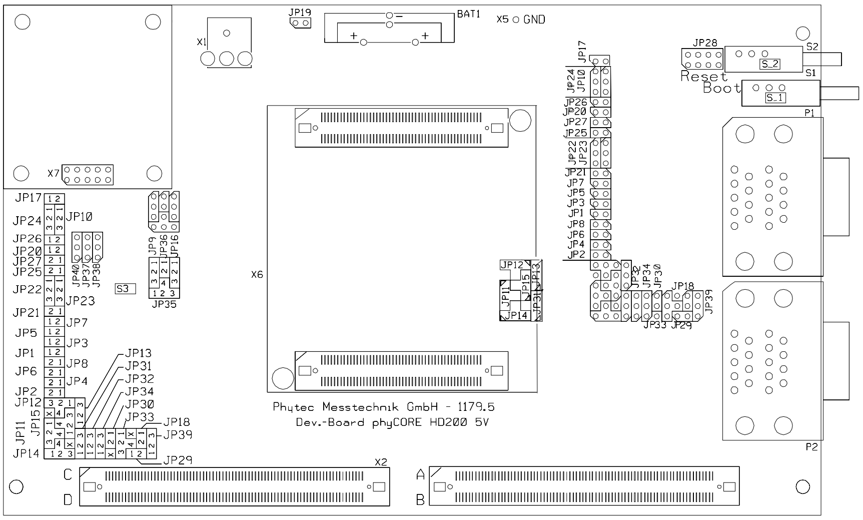


Figure 16: Location of the Jumpers (View of the Component Side)

Figure 17 shows the factory default jumper settings for operation of the phyCORE Development Board HD200 XC with the standard phyCORE-XC161 (standard = XC161 controller, use of two RS-232 interfaces, two CAN interfaces, LED D3, the Boot button on the Development Board). Jumper settings for other functional configurations of the phyCORE-XC161 module mounted on the Development Board are described in *section 15.3*.

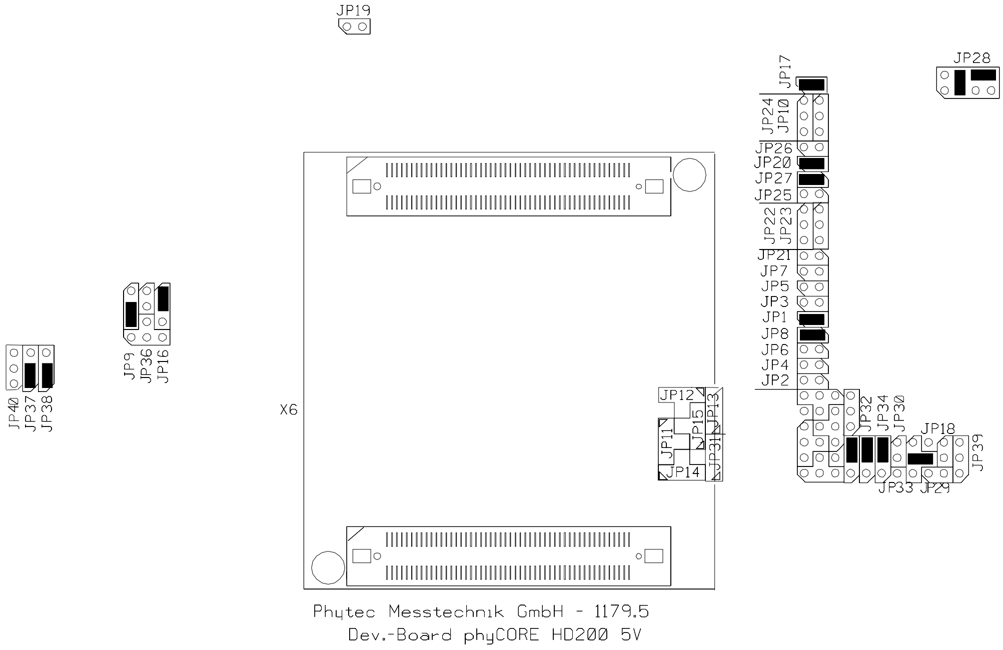


Figure 17: Default Jumper Settings of the phyCORE Development Board HD200 XC with phyCORE-XC161

15.2.3 Unsupported Features and Improper Jumper Settings

The following table contains improper jumper settings for operation of the phyCORE-XC161 on a phyCORE Development Board HD200 XC. Functions configured by these settings are not supported by the phyCORE module.

Supply Voltage:

The phyCORE Development Board HD200 XC supports two main supply voltages for the start-up of various phyCORE modules. When using the phyCORE-XC161, two main supply voltages are required, VCC with 5 V and VCC2 with 2.5 V.

Jumper	Setting	Description
JP16	2 + 3 or open	VCC or no supply voltage routed to pins X1C4 and X1C5 on the phyCORE-XC161

Table 28: Improper Jumper Setting for JP16 on the Development Board

No RS-485 interface:

DB-9 plug P2B on the Development Board can be configured as RS-485 interface as an alternative to a possible second CAN interface. The phyCORE-XC161 does not support an RS-485 interface. For this reason the corresponding jumper settings should never be used.

Jumper	Setting	Description
JP30	closed	TxD signal for second serial interface routed to pin 8 on the DB-9 plug P2B
JP33	1 + 2	RxD signal for second serial interface routed to pin 2 on the DB-9 plug P2B

Table 29: Improper Jumper Setting for JP30/33 on the Development Board

Reference Voltage Source for A/D Converter

Pins X1C42, X1C47, X1D39, X1D44 and X1D49 (VAGND) of the phyCORE-XC161 are solely connected with the phyCORE Development Board HD200 XC GND potential. This makes a separate supply with an alternative VAGND potential impossible. Jumper J2 on the phyCORE-XC161 is therefore without function when the module is mounted on a Development Board HD200 XC. Free definition of the VAGND potential is however available in a customer application board.

15.3 Functional Components on the phyCORE Development Board HD200 XC

This section describes the functional components of the phyCORE Development Board HD200 XC supported by the phyCORE-XC161 and appropriate jumper settings to activate these components. Depending on the specific configuration of the phyCORE-XC161 module, alternative jumper settings can be used. These jumper settings are different from the factory default settings as shown in *Figure 17* and enable alternative or additional functions on the phyCORE Development Board HD200 XC depending on user needs.

15.3.1 Power Supply at X1

Caution:
Do not use a laboratory adapter to supply power to the Development Board! Power spikes during power-on could destroy the phyCORE module mounted on the Development Board! Do not change modules or jumper settings while the Development Board is supplied with power!

Permissible input voltage: +/-5 VDC regulated.

The required current load capacity of the power supply depends on the specific configuration of the phyCORE-XC161 mounted on the Development Board as well as whether an optional expansion board is connected to the Development Board. An adapter with a minimum supply of 500 mA is recommended.

Jumper	Setting	Description
JP9	2 + 3	5 V main supply voltage (VCC) to the phyCORE-XC161
J16	1 + 2	2.5 V as secondary supply voltage (VCC2) to the phyCORE-XC161

Table 30: JP9, JP16 Configuration of the Supply Voltages VCC/VCC2

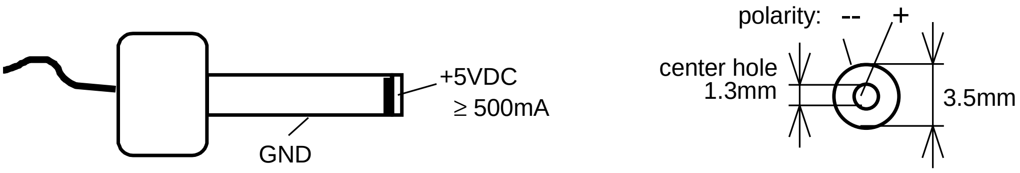


Figure 18: Connecting the Supply Voltage at X1

Caution:

When using this function, the following jumper settings are not allowed:

Jumper	Setting	Description
JP9	1 + 2	2.5 V as main supply voltage (VCC) for the phyCORE-XC161
	open	phyCORE-XC161 not connected to main supply voltage (VCC)
JP16	2 + 3	5 V as secondary supply voltage (VCC2) for the phyCORE-XC161
	open	phyCORE-XC161 not connected to secondary supply voltage (VCC2)

Table 31: JP9, JP16 Improper Jumper Settings for the Supply Voltages

Setting Jumper JP9 to positions 1+2 configures a main power supply to the phyCORE-XC161 of 2.5 V which could destroy the module. If Jumper JP9 is open, no main power supply is connected to the phyCORE-XC161. This jumper setting should therefore not be used.

Setting Jumper JP16 to positions 2+3 configures a secondary power supply to the phyCORE-XC161 of 5 V which could destroy the module. If Jumper JP16 is open, no main power supply is connected to the phyCORE-XC161. This jumper setting should therefore not be used.

15.3.2 Activating the Bootstrap Loader

The Infineon XC161 microcontroller contains an on-chip Bootstrap Loader that provides basic communication and programming functions. The combination of this Bootstrap Loader and the corresponding FlashTools software installed on the PC allows for Flash programming with application code via an RS-232 interface. The Bootstrap Loader is also used by other third party toolpartner software such as the Monitor166 from Keil or CrossView Pro ROM monitor from Altium for debugging functions.

In order to start the on-chip Bootstrap Loader on the phyCORE-XC161, the data line D4 of the microcontroller must be connected to a low-level signal at the time the Reset signal changes from its active to the inactive state. This is achieved by applying a high-level signal at pin X1C9 of the phyCORE-XC161 as the Boot input is high-active.

The phyCORE Development Board HD200 XC provides three different options to activate the on-chip Bootstrap Loader:

1. The Boot button (S1) can be connected to VCC via Jumper JP28 which is located next the the Boot and Reset buttons at S1 and S2. This configuration enables start-up of the on-chip Bootstrap Loader if the Boot button is pressed during a hardware reset or power-on.

Jumper	Setting	Description
JP28	6 + 8 and 3 + 4	Boot button (in conjunction with Reset button or connection of the power supply) starts the Bootstrap Loader on the XC161

Table 32: JP28 Configuration of the Boot Button

2. The Boot input of the phyCORE-XC161 can also be permanently connected to VCC via a pull-up resistor. This pulls the data line D4 to low level via an on-board circuitry which then starts the Bootstrap Loader. This spares pushing the Boot button during a hardware reset or power-on.

Caution:
In this configuration a regular reset, hence normal start of your application, is not possible. The Bootstrap Loader is started every time. This is useful when using an emulator.

Jumper	Setting	Description
JP28	4 + 6	Boot input connected permanently with VCC via pull-up resistor. The Bootstrap Loader is always started with Reset button or with connection of the power supply

Table 33: JP28 Configuration of a Permanent Bootstrap Loader Start

3. It is also possible to start the FlashTools via external signals applied to the DB-9 socket P1A. This requires control of the signal transition on the Reset line via pin 7 while a static high-level is applied to pin 4 for the Boot signal.

Jumper	Setting	Description
JP22	2 + 3	Pin 7 (CTS) of the DB-9 socket P1A as Reset signal for the phyCORE-XC161
JP23	2 + 3	Pin 4 (DSR) of the DB-9 socket P1A as Boot signal for the phyCORE-XC161
JP10	2 + 3	High-level Boot signal connected with the Boot input of the phyCORE-XC161

Table 34: JP22, JP23, JP10 Configuration of Boot via RS-232

Caution:
When using this function, the following jumper setting is not allowed:

Jumper	Setting	Description
JP10	1 + 2	Jumper setting generates low-level on Boot input of the phyCORE-XC161

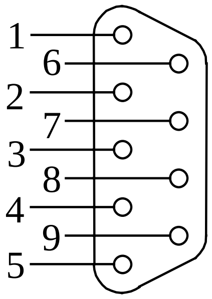
Table 35: Improper Jumper Settings for Boot via RS-232

15.3.3 First Serial Interface at Socket P1A

Socket P1A is the lower socket of the double DB-9 connector at P1. P1A is connected via jumpers to the first serial interface of the phyCORE-XC161. When connected to a host-PC, the phyCORE-XC161 can be rendered in Bootstrap mode via signals applied to the socket P1A (*refer to section 15.3.2*).

Jumper	Setting	Description
JP20	closed[1]	Pin 2 of DB-9 socket P1A connected with RS-232 interface signal TxD0 of the phyCORE-XC161
JP21	open	Pin 9 of DB-9 socket P1A not connected
JP22	open	Pin 7 of DB-9 socket P1A not connected
	2 + 3[2]	Reset input of the module can be controlled via RTS signal from a host-PC
JP23	open	Pin 4 of DB-9 socket P1A not connected
	2 + 3[2]	Boot input of the module can be controlled via DTR signal from a host-PC (**Note**: JP10 must be set to position 2 + 3)
JP24	open	Pin 6 of DB-9 socket P1A not connected
JP25	open	Pin 8 of DB-9 socket P1A not connected
JP26	open	Pin 1 of DB-9 socket P1A not connected
JP27	closed[1]	Pin 3 of DB-9 socket P1A connected with RS-232 interface signal RxD0 from the phyCORE-XC161

Table 36: Jumper Configuration for the First RS-232 Interface



Pin 2:	TxD0
Pin 3:	RxD0
Pin 5:	GND

Figure 19: Pin Assignment of the DB-9 Socket P1A as First RS-232 (Front View)

[1]: This jumper should always be closed because communication with PHYTEC FlashTools requires use of the first serial interface on the phyCORE module.

[2]: Alternative jumper configuration for additional features (*refer to section 15.3.2*). Not required for standard communication functions.

Caution:

When using the DB-9 socket P1A as RS-232 interface on the phyCORE-XC161 the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP20	open	Pin 2 of DB-9 socket P1A not connected, no connection to TxD0 signal from phyCORE-XC161
JP21	closed	Pin 9 of DB-9 socket P1A connected with RTC_CLKOUT from phyCORE-XC161
JP22	1 + 2	Pin 7 of DB-9 socket P1A connected with port P2.15 from phyCORE-XC161
JP23	1 + 2	Pin 4 of DB-9 socket P1A connected with /BRKIN from phyCORE-XC161
JP24	1 + 2	Pin 6 of DB-9 socket P1A connected with /BRKOUT from phyCORE-XC161
JP25	closed	Pin 8 of DB-9 socket P1A connected with port P2.14 from phyCORE-XC161
JP26	closed	Pin 1 of DB-9 socket P1A connected with /IRQ_ETH from phyCORE-XC161
JP27	open	Pin 3 of DB-9 socket P1A not connected, no connection to RxD0 signal from phyCORE-XC161

Table 37: Improper Jumper Settings for DB-9 Socket P1A as First RS-232

If an RS-232 cable is connected to P1A, the voltage level on the RS-232 lines could destroy the phyCORE-XC161.

15.3.4 Power Supply to External Devices via Socket P1A

The phyCORE Development Board HD200 XC can be populated by additional components that provide a supply voltage of 5 V at pin 6 of DB-9 socket P1A. This allows for easy and secure supply of external devices connected to P1A. This power supply option especially supports connectivity to analog and digital modems. Such modem devices enable global communication of the phyCORE -XC161 over the Internet or a direct dial connection.

The following figure shows the location of these components on the Development Board:

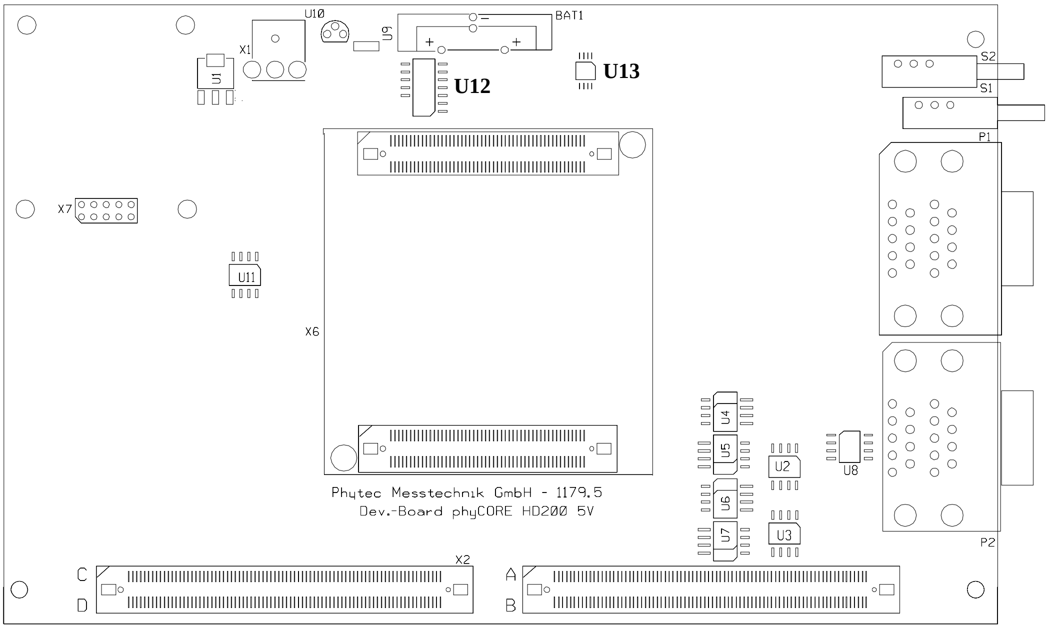


Figure 20: Location of Components at U12 and U13 for Power Supply to External Subassemblies

The components at U12 and U13 guarantee electronic protection against overvoltage and excessive current draw at pin 6 of P1A; in particular:

- Load detection and controlled voltage supply switch-on:

  In order to ensure clear detection of the switch-on condition, the connected device should cause a current draw of at least 10 mA at pin 6. The controlled voltage supply switch-on prevents voltage drop off on the phyCORE Development Board HD200 XC.

- Overvoltage Protection:

  If the voltage at pin 6 exceeds the limiting value that can be provided by the phyCORE Development Board HD200 XC, the voltage at pin 6 will be switched off immediately. This prevents damage to the phyCORE Development Board HD200 XC as well as connected modules and expansion boards.

- Overload Protection:

  If the current draw at pin 6 exceeds the limiting value of approximately 150 mA, the voltage at pin 6 will be switched off immediately. This prevents damage to the phyCORE Development Board HD200 XC and its power adapter caused by current overload.

This configuration option provides the following possibility:

Jumper	Setting	Description
JP24	2 + 3	Electronically protected 5 V at pin 6 for supply of external devices connected to P1A

Table 38: JP24 Power Supply to External Devices Connected to P1A on the Development Board

15.3.5 Second Serial Interface at Socket P1B

Socket P1B is the upper socket of the double DB-9 connector at P1. P1B is connected via jumpers to the second serial interface of the phyCORE-XC161. Depending on the module configuration (*refer to section 3.15*) and different options are available for configuration of socket P1B.

1. Default configuration, second serial interface with RS-232 level:

Jumper	Setting	Description
JP1	closed	Pin 2 of DB-9 socket P1B connected with RS-232 interface signal TxD1 of the phyCORE-XC161
JP2	open	Pin 9 of DB-9 socket P1B not connected
JP3	open	Pin 7 of DB-9 socket P1B not connected
JP4	open	Pin 4 of DB-9 socket P1B not connected
JP5	open	Pin 6 of DB-9 socket P1B not connected
JP6	open	Pin 8 of DB-9 socket P1B not connected
JP7	open	Pin 1 of DB-9 socket P1B not connected
JP8	closed	Pin 3 of DB-9 socket P1B connected with RS-232 interface signal RxD1 from the phyCORE-XC161

Table 39: Jumper Configuration for the Second RS-232 Interface

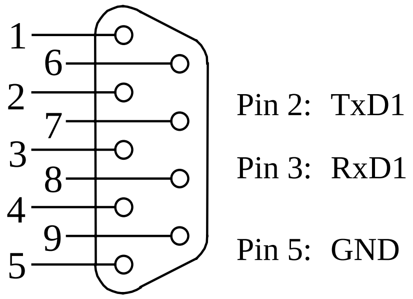


Figure 21: Pin Assignment of the DB-9 Socket P1B as Second RS-232 (Front View)

Caution:
When using the DB-9 socket P1B as RS-232 interface on the phyCORE-XC161 the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP2	closed	Pin 9 of DB-9 socket P1B connected with /TRST signal from the phyCORE-XC161
JP3	closed	Pin 7 of DB-9 socket P1B connected with TDO signal from the phyCORE-XC161
JP4	closed	Pin 4 of DB-9 socket P1B connected with TMS signal from the phyCORE-XC161
JP5	closed	Pin 6 of DB-9 socket P1B connected with TCK signal from the phyCORE-XC161
JP6	closed	Pin 8 of DB-9 socket P1B connected with TDI signal from the phyCORE-XC161
JP7	closed	Pin 1 of DB-9 socket P1B connected with NC pin on the phyCORE-XC161

Table 40: Improper Jumper Settings for DB-9 Socket P1B (2nd RS-232)

2. Optional configuration, no second serial interface, P3.0 and P3.1 available as I/O pins:

Jumper	Setting	Description
JP1	open	Pin 2 of DB-9 socket P1B not connected
JP2	open	Pin 9 of DB-9 socket P1B not connected
JP3	open	Pin 7 of DB-9 socket P1B not connected
JP4	open	Pin 4 of DB-9 socket P1B not connected
JP5	open	Pin 6 of DB-9 socket P1B not connected
JP6	open	Pin 8 of DB-9 socket P1B not connected
JP7	open	Pin 1 of DB-9 socket P1B not connected
JP8	open	Pin 3 of DB-9 socket P1B not connected

Table 41: Jumper Configuration of the DB-9 Socket P1B (no Second RS-232)

In this configuration no second serial interface is available.

Caution:
When using the DB-9 socket P1B with the configuration of the phyCORE-XC161 as described above, the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP1	closed	No TxD1_RS232 signal available from the phyCORE-XC161 (P1B pin 2)
JP2	closed	No RI1_TTL signal available from the phyCORE-XC161 (P1B pin 9)
JP3	closed	No CTS1_RS232 signal available from the phyCORE-XC161 (P1B pin 7)
JP4	closed	No DSR1_RS232 signal available from the phyCORE-XC161 (P1B pin 4)
JP5	closed	No DTR1_RS232 signal available from the phyCORE-XC161 (P1B pin 6)
JP6	closed	No RTS1_RS232 signal available from the phyCORE-XC161 (P1B pin 8)
JP7	closed	No CD1_RS232 signal available from the phyCORE-XC161 (P1B pin 1)
JP8	closed	No RxD1_RS232 signal available from the phyCORE-XC161 (P1B pin 3)

Table 42: Improper Jumper Settings for DB-9 Socket P1B (no Second RS-232)

If an RS-232 cable is connected to P1B by mistake, the voltage level on the RS-232 lines could destroy the phyCORE-XC161.

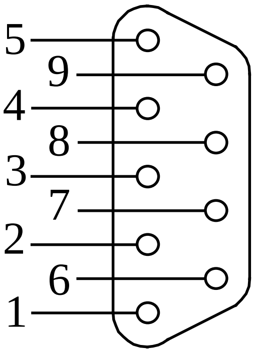
15.3.6 First CAN Interface at Plug P2A

Plug P2A is the lower plug of the double DB-9 connector at P2. P2A is connected to the first CAN interface (CAN0) of the phyCORE-XC161 via jumpers. Depending on the configuration of the CAN transceivers and their power supply, the following three configurations are possible:

1. CAN transceiver populating the phyCORE-XC161 is enabled and the CAN signals from the module extend directly to plug P2A.

Jumper	Setting	Description
JP31	2 + 3	Pin 2 of the DB-9 plug P2A is connected to CAN-L0 from on-board transceiver on the phyCORE module
JP32	2 + 3	Pin 7 of the DB-9 plug P2A is connected to CAN-H0 from on-board transceiver on the phyCORE module
JP11	open	Input at opto-coupler U4 on the phyCORE Development Board HD200 XC open
JP12	open	Output at opto-coupler U5 on the phyCORE Development Board HD200 2.5V open
JP13	open	No supply voltage to CAN transceiver and opto-coupler on the phyCORE Development Board HD200 XC
JP18	open	No GND potential at CAN transceiver and opto-coupler on the phyCORE Development Board HD200 XC
JP29	open	No power supply via CAN bus
JP39	open	No power supply via CAN bus

Table 43: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the phyCORE-XC161



Pin 3:	GND	(Development Board Ground)
Pin 7:	CAN-H0	(not galvanically separated)
Pin 2:	CAN-L0	(not galvanically separated)
Pin 6:	GND	(Development Board Ground)

Figure 22: Pin Assignment of the DB-9 Plug P2A (CAN Transceiver on phyCORE-XC161, Front View)

2. The CAN transceiver populating the phyCORE-XC161 is disabled; CAN signals generated by the CAN transceiver (U2) on the Development Board extending to connector P2A **without galvanic seperation:**

Jumper	Setting	Description
JP31	1 + 2	Pin 2 of DB-9 plug P2A connected with CAN-L0 from CAN transceiver U2 on the Development Board
JP32	1 + 2	Pin 7 of DB-9 plug P2A connected with CAN-H0 from CAN transceiver U2 on the Development Board
JP11	2 + 3	Input at opto-coupler U4 on the Development Board connected to CAN1_Tx (P4.6[1]) of the XC161
	1 + 2	Input at opto-coupler U4 on the Development Board connected to CAN1_Tx (P9.3[2]) of the XC161
JP12	2 + 3	Output at opto-coupler U5 on the Development Board connected to CAN1_Rx (P4.5[3]) of the XC161
	1 + 2	Output at opto-coupler U5 on the Development Board connected to CAN1_Rx (P9.2[4]) of the XC161
JP13	2 + 3	Supply voltage for CAN transceiver and opto-coupler derived from local supply circuitry on the phyCORE Development Board HD200 XC
JP18	closed	CAN transceiver and opto-coupler on the Development Board connected with local GND potential
JP29	open	No power supply via CAN bus
JP39	open	No power supply via CAN bus

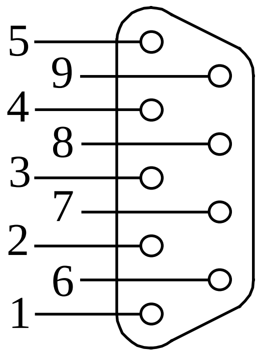
Table 44: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the Development Board

[1] Port P4.6 is the default port for CAN1_Tx (standard).

[2] Port P9.3 is the alternative port for CAN1_Tx (*see Controller User‘s Manual/Data Sheet*).

[3] Port P4.5 is the default port for CAN1_Rx (standard).

[4] Port P9.2 is the alternative port for CAN1_Rx (*see Controller User‘s Manual/Data Sheet*).



Pin 3:	GND	(Development Board Ground)
Pin 7:	CAN-H0	(not galvanically separated)
Pin 2:	CAN-L0	(not galvanically separated)
Pin 6:	GND	(Development Board Ground

Figure 23: Pin Assignment of the DB-9 Plug P2A (CAN Transceiver on Development Board)

Caution:

When using the DB-9 connector P2A as CAN interface and the CAN transceiver on the Development Board the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP31	2 + 3	Pin 2 of DB-9 plug P2A connected with CAN-L0 from on-board transceiver on the phyCORE-XC161
JP32	2 + 3	Pin 7 of DB-9 plug P2A connected with CAN-H0 from on-board transceiver on the phyCORE-XC161
JP11	open	Input at opto-coupler U4 on the Development Board not connected
JP12	open	Output at opto-coupler U5 on the Development Board not connected
JP13	1 + 2	Supply voltage for CAN transceiver and opto-coupler on the Development Board derived from external source (CAN bus) via on-board voltage regulator
JP29	closed	Supply voltage for on-board voltage regulator from pin 9 of DB-9 connector P2A
JP39	*see Table 47*	CAN bus supply voltage reduction for CAN circuitry

Table 45: Improper Jumper Settings for the CAN Plug P2A (CAN Transceiver on the Development Board)

3. The CAN transceiver populating the phyCORE-XC161 is disabled; CAN signals generated by the CAN transceiver (U2) on the Development Board extend to connector P2A **with galvanic separation.** This configuration requires connection of an external CAN supply voltage of 7 to 28 V. The external power supply must be **only** connected to either P2A **or** P2B.

Jumper	Setting	Description
JP31	1 + 2	Pin 2 of DB-9 plug P2A connected with CAN-L0 from CAN transceiver U2 on the Development Board
JP32	1 + 2	Pin 7 of DB-9 plug P2A connected with CAN-H0 from CAN transceiver U2 on the Development Board
JP11	2 + 3	Input at opto-coupler U4 on the Development Board connected to CAN1_Tx (P4.6[1]) of the XC161
	1 + 2	Input at opto-coupler U4 on the Development Board connected to CAN1_Tx (P9.3[2]) of the XC161
JP12	2 + 3	Output at opto-coupler U5 on the Development Board connected to CAN1_Rx (P4.5[3]) of the XC161
	1 + 2	Output at opto-coupler U5 on the Development Board connected to CAN1_Rx (P9.2[4]) of the XC161
JP13	1 + 2	Supply voltage for CAN transceiver and opto-coupler on the Development Board derived from external source (CAN bus) via on-board voltage regulator
JP18	open	CAN transceiver and opto-coupler on the Development Board disconnected from local GND potential
JP29	closed	Supply voltage for on-board voltage regulator from pin 9 of DB-9 plug P2A
JP39	*see Table 47*	CAN bus supply voltage reduction for CAN circuitry

Table 46: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the Development Board with Galvanic Separation

[1]: Port P4.6 is the default port for CAN1_Tx (standard).

[2]: Port P9.3 is the alternative port for CAN1_Tx (*see Controller User's Manual/Data Sheet*).

[3]: Port P4.5 is the default port for CAN1_Rx (standard).

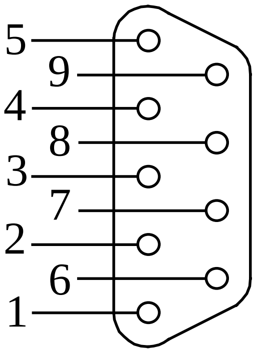
[4]: Port P9.2 is the alternative port for CAN1_Rx (*see Controller User's Manual/Data Sheet*).

CAN Bus Voltage Supply Reduction via JP39:

Depending on the voltage level that is supplied over the CAN bus at P2A or P2B (VCAN_IN1+) JP39 must be configured in order to routed the applicable voltage to the CAN voltage regulator at U8 on the Development Board:

VCAN_IN+	JP39
7 V..18 V	1 + 2
18 V..23 V	2 + 3
23 V..28 V	open

Table 47: JP39 CAN Bus Voltage Supply Reduction



Pin 9:	VCAN+
Pin 3:	VCAN-
Pin 7:	CAN-H0 (galvanically separated)
Pin 2:	CAN-L0 (galvanically separated)
Pin 6:	VCAN-

Figure 24: Pin Assignment of the DB-9 Plug P2A (CAN Transceiver on Development Board with Galvanic Separation)

Caution:
When using the DB-9 plug P2A as CAN interface, and the CAN transceiver on the Development Board with galvanic separation, the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP31	2 + 3	Pin 2 of DB-9 plug P2A connected with CAN-L0 from on-board transceiver on the phyCORE-XC161
JP32	2 + 3	Pin 7 of DB-9 plug P2A connected with CAN-H0 from on-board transceiver on the phyCORE-XC161
JP11	open	Input at opto-coupler U4 on the Development Board not connected
JP12	open	Output at opto-coupler U5 on the Development Board not connected
JP13	2 + 3	Supply voltage for CAN transceiver and opto-coupler derived from local supply circuitry on the phyCORE Development Board HD200 XC
JP18	closed	CAN transceiver and opto-coupler on the Development Board connected with local GND potential
JP29	open	No power supply via CAN bus
JP39	*see Table 47*	Incorrect CAN bus supply voltage reduction for CAN circuitry

Table 48: Improper Jumper Settings for the CAN Plug P2A (CAN Transceiver on Development Board with Galvanic Separation)

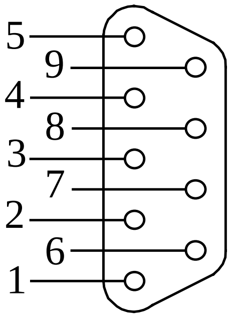
15.3.7 Second CAN Interface at Plug P2B

Plug P2B is the upper plug of the double DB-9 connector at P2. P2B is connected to the second CAN interface (CAN1) of the phyCORE-XC161 via jumpers. Depending on the configuration of the CAN transceivers and their power supply, the following three configurations are possible:

1. CAN transceiver populating the phyCORE-XC161 is enabled and the CAN signals from the module extend directly to plug P2B.

Jumper	Setting	Description
JP33	2 + 4	Pin 2 of the DB-9 plug P2B is connected to CAN-L1 from on-board transceiver on the phyCORE module
JP34	2 + 3	Pin 7 of the DB-9 plug P2B is connected to CAN-H1 from on-board transceiver on the phyCORE module
JP14	open	Input at opto-coupler U6 on the phyCORE Development Board HD200 XC open
JP15	open	Output at opto-coupler U7 on the phyCORE Development Board HD200 XC open
JP13	open	CAN transceiver and opto-coupler on the Development Board disconnected from supply voltage
JP18	open	No GND potential at CAN transceiver and opto-coupler on the phyCORE Development Board HD200 XC
JP29	open	No power supply via CAN bus
JP39	open	No power supply via CAN bus

Table 49: Jumper Configuration for CAN Plug P2B using the CAN Transceiver on the phyCORE-XC161



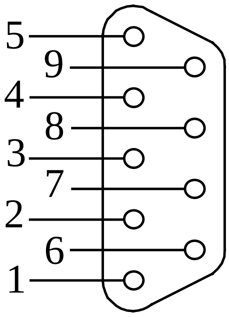
Pin 3:	GND	(Development Board Ground)
Pin 7:	CAN-H1	(not galvanically separated)
Pin 2:	CAN-L1	(not galvanically separated)
Pin 6:	GND	(Development Board Ground)

Figure 25: Pin Assignment of the DB-9 Plug P2B (CAN Transceiver on phyCORE-XC161)

2. The CAN transceiver populating the phyCORE-XC161 is disabled; CAN signals generated by the CAN transceiver (U3) on the Development Board extending to connector P2B **without galvanic seperation:**

Jumper	Setting	Description
JP33	2 + 3	Pin 2 of DB-9 plug P2B connected with CAN-L1 from CAN transceiver U3 on the Development Board
JP34	1 + 2	Pin 7 of DB-9 plug P2B connected with CAN-H1 from CAN transceiver U3 on the Development Board
JP14	2 + 3	Input at opto-coupler U6 on the Development Board connected to CAN2_Tx (P4.7[1]) of the XC161
	1 + 2	Input at opto-coupler U6 on the Development Board connected to CAN2_Tx (P9.1[2]) of the XC161
JP15	2 + 3	Output at opto-coupler U7 on the Development Board connected to CAN2_Rx (P4.4[3]) of the XC161
	1 + 2	Output at opto-coupler U7 on the Development Board connected to CAN2_Rx (P9.0[4]) of the XC161
JP13	open	CAN transceiver and opto-coupler on the Development Board disconnected from supply voltage
JP18	open	No GND potential at CAN transceiver and opto-coupler on the phyCORE Development Board HD200 XC
JP29	open	No power supply via CAN bus
JP39	open	No power supply via CAN bus

Table 50: Jumper Configuration for CAN Plug P2B using the CAN Transceiver on the Development Board



Pin 3:	GND	(Development Board Ground)
Pin 7:	CAN-H1	(not galvanically separated)
Pin 2:	CAN-L1	(not galvanically separated)
Pin 6:	GND	(Development Board Ground)

Figure 26: Pin Assignment of the DB-9 Plug P2B (CAN Transceiver on phyCORE-XC161)

[1] Port P4.7 is the default port for CAN2_Tx (standard).

[2] Port P9.1 is the alternative port for CAN2_Tx (*see XC161 User's Manual/Data Sheet*).

[3] Port P4.4 is the default port for CAN2_Rx (standard).

[4] Port P9.0 is the alternative port for CAN2_Rx (*see XC161 User's Manual/Data Sheet*).

Caution:
When using the DB-9 connector P2B as second CAN interface and the CAN transceiver on the Development Board the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP30	closed	Pin 8 at P2B is connected with TxD1_RS232 from the phyCORE-XC161
JP33	1 + 2	Pin 2 at P2B is connected with P9.5 from the phyCORE-XC161
	2 + 4	Pin 2 at P2B is connected with CAN_L1 from the on-board CAN transceiver on the phyCORE-XC161
JP34	2 + 3	Pin 7 at P2B is connected with CAN_H1 from the on-board CAN transceiver on the phyCORE-XC161
JP14	open	Input at opto-coupler U6 on the Development Board not connected
JP15	open	Output at opto-coupler U7 on the Development Board not connected
JP13	1 + 2	Supply voltage for CAN transceiver and opto-coupler on the Development Board derived from external source (CAN bus) via on-board voltage regulator
JP29	closed	Supply voltage for on-board voltage regulator from pin 9 of DB-9 connector P2A
JP39	*see Table 47*	CAN bus supply voltage reduction for CAN circuitry

Table 51: Improper Jumper Settings for the CAN Plug P2B (CAN Transceiver on the Development Board)

3. The CAN transceiver populating the phyCORE-XC161 is disabled; CAN signals generated by the CAN transceiver (U3) on the Development Board extend to connector P2B **with galvanic separation.** This configuration requires connection of an external CAN supply voltage of 7 to 28 V. The external power supply must be **only** connected to either P2A **or** P2B.

Jumper	**Setting**	**Description**
JP33	2 + 3	Pin 2 of DB-9 plug P2B connected with CAN-L1 from CAN transceiver U3 on the Development Board
JP34	1 + 2	Pin 7 of DB-9 plug P2B connected with CAN-H1 from CAN transceiver U3 on the Development Board
JP14	2 + 3	Input at opto-coupler U6 on the Development Board connected to CAN2_Tx (P4.7[1]) of the XC161
	1 + 2	Input at opto-coupler U4 on the Development Board connected to CAN2_Tx (P9.1[2]) of the XC161
JP15	2 + 3	Output at opto-coupler U7 on the Development Board connected to CAN2_Rx (P4.4[3]) of the XC161
	1 + 2	Output at opto-coupler U7 on the Development Board connected to CAN2_Rx (P9.0[4]) of the XC161
JP13	1 + 2	Supply voltage for CAN transceiver and opto-coupler on the Development Board derived from external source (CAN bus) via on-board voltage regulator
JP18	open	CAN transceiver and opto-coupler on the Development Board disconnected from local GND potential
JP29	closed	Supply voltage for on-board voltage regulator from pin 9 of DB-9 plug P2B or P2A
JP39	*see Table 47*	CAN bus supply voltage reduction for CAN circuitry

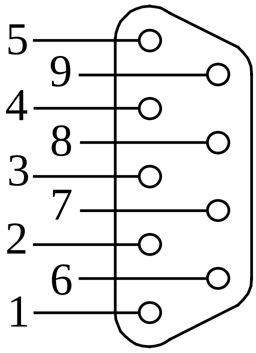
Table 52: Jumper Configuration for CAN Plug P2B using the CAN Transceiver on the Development Board with Galvanic Separation

[1]: Port P4.7 is the default port for CAN2_Tx (standard).

[2]: Port P9.1 is the alternative port for CAN2_Tx (*see XC161 User‘s Manual/Data Sheet*).

[3]: Port P4.4 is the default port for CAN2_Rx (standard).

[4]: Port P9.0 is the alternative port for CAN2_Rx (*see XC161 User‘s Manual/Data Sheet*).



Pin 9:	VCAN+
Pin 3:	VCAN-
Pin 7:	CAN-H1 (galvanically separated)
Pin 2:	CAN-L1 (galvanically separated)
Pin 6:	VCAN-

Figure 27: Pin Assignment of the DB-9 Plug P2B (CAN Transceiver on Development Board with Galvanic Separation)

Caution:
When using the DB-9 plug P2B as second CAN interface, and the CAN transceiver on the Development Board with galvanic separation, the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP30	closed	Pin 8 at P2B is connected with TxD1_RS232 from the phyCORE-XC161
JP33	1 + 2	Pin 2 at P2B is connected with P9.5 from the phyCORE-XC161
	2 + 4	Pin 2 at P2B is connected with CAN_L1 from the on-board CAN transceiver on the phyCORE-XC161
JP34	2 + 3	Pin 7 at P2B is connected with CAN_H1 from the on-board CAN transceiver on the phyCORE-XC161
JP14	open	Input at opto-coupler U6 on the Development Board not connected
JP15	open	Output at opto-coupler U7 on the Development Board not connected
JP13	2 + 3	Supply voltage for CAN transceiver and opto-coupler derived from local supply circuitry on the phyCORE Development Board HD200 XC
JP18	closed	CAN transceiver and opto-coupler on the Development Board connected with local GND potential
JP29	open	No power supply via CAN bus
JP39	*see Table 47*	Incorrect CAN bus supply voltage reduction for CAN circuitry

Table 53: Improper Jumper Settings for the CAN Plug P2B (CAN Transceiver on Development Board with Galvanic Separation)

15.3.8 Programmable LED D3

The phyCORE Development Board HD200 XC offers a programmable LED at D3 for user implementations. This LED can be connected to port pin P9.0 of the phyCORE-XC161 which is available via signal GPIO0 (JP17 = closed). A low-level at port pin P9.0 causes the LED to illuminate, LED D3 remains off when writing a high-level to P9.0.

Jumper	Setting	Description
JP17	closed	Port pin P9.0 (GPIO0) of the XC161 controller controls LED D3 on the Development Board

Table 54: JP17 Configuration of the Programmable LED D3

15.3.9 Pin Assignment Summary of the phyCORE, the Expansion Bus and the Patch Field

As described in *section 15.1*, all signals from the phyCORE-XC161 extend in a strict 1:1 assignment to the Expansion Bus connector X2 on the Development Board. These signals, in turn, are routed in a similar manner to the patch field on an optional expansion board that mounts to the Development Board at X2.

Please note that, depending on the design and size of the expansion board, only a portion of the entire patch field is utilized under certain circumstances. When this is the case, certain signals described in the following section will not be available on the expansion board. However, the pin assignment scheme remains consistent.

A two dimensional numbering matrix similar to the one used for the pin layout of the phyCORE-connector is provided to identify signals on the Expansion Bus connector (X2 on the Development Board) as well as the patch field.

However, the numbering scheme for Expansion Bus connector and patch field matrices differs from that of the phyCORE-connector, as shown in the following two figures:

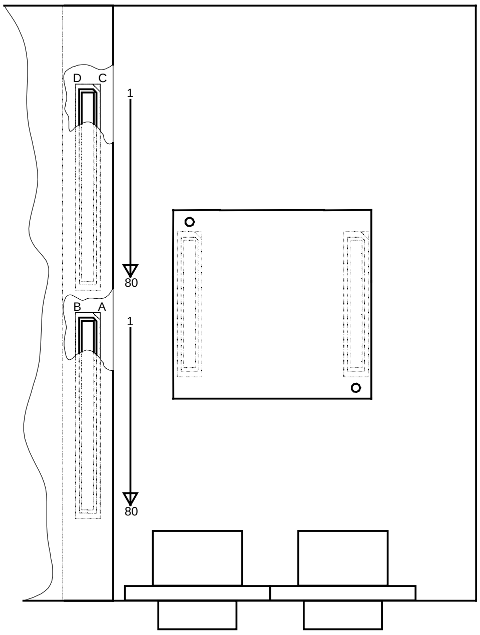


Figure 28: Pin Assignment Scheme of the Expansion Bus

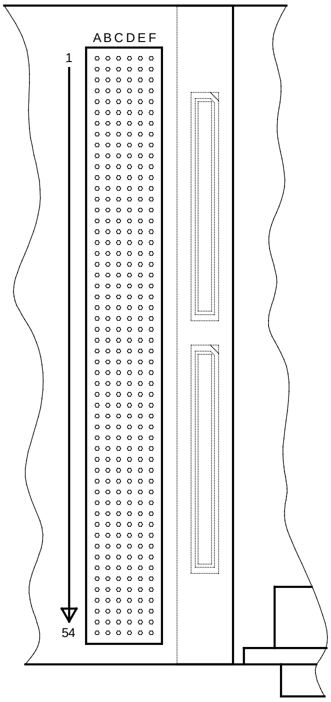


Figure 29: Pin Assignment Scheme of the Patch Field

The pin assignment on the phyCORE-XC161, in conjunction with the Expansion Bus (X2) on the Development Board and the patch field on an expansion board, is as follows:

Signal	**phyCORE-XC161**	**Expansion Bus**	**Patch Field**
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	30F
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	32A
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	16A	16A	33A
P1H.5/A13	16B	16B	33C
P1H.6/A14	17B	17B	33E
P1H.7/A15	18A	18A	33B

Table 55: Pin Assignment Data/Address Bus for the phyCORE-XC161 / Development Board / Expansion Board

Signal	phyCORE-XC161	Expansion Bus	Patch Field
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
P2.8/CC8IO/EX0IN	2B	2B	28E
P2.9/CC9IO/EX1IN	3A	3A	28B
P2.10/CC10IO/EX2IN	3B	3B	28F
P2.11/CC11IO/EX3IN	38D	38D	13E
P2.12 /CC12IO/EX4IN	38C	38C	13A
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/TCK	45A	45A	42B
P3.2/CAPIN/TDI	45B	45B	42F
P3.3/T3OUT/TDO	46A	46A	43A
P3.4/T3EUD/TMS	46B	46B	43C
P3.5/T4IN/ /BRKOUT	47B	47B	43E
P3.6/T3IN	48A	48A	43B
P3.7/T2IN/ /BRKIN	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

Table 56: Pin Assignment Port P2, P3, P4 for the phyCORE-XC161 / Development Board / Expansion Board

Signal	phyCORE-XC161	Expansion Bus	Patch Field
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	45CD	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
P7.4/CC28IO	40A	40A	40F
P7.5/CC29IO	40B	40B	41A
P7.6/CC30IO	41A	41A	41E
P7.7/CC31IO	41B	41B	41B
P9.0/CAN2_RxD	11D	11D	4A
P9.1/CAN2_TxD	12D	12D	4B
P9.2/CAN1_RxD	13C	13C	4F
P9.3/CAN1_TxD	13D	13D	5A
P9.4/CC20IO	14C	14C	5C
P9.5/CC21IO	15C	15C	5E

Table 57: Pin Assignment Port P5, P6, P7, P9 for the phyCORE-XC161 / Development Board / Expansion Board

Signal	**phyCORE-XC161**	**Expansion Bus**	**Patch Field**
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_RS232	23C	23C	8A
RxD0_TTL	16D	16D	6A
TxD0_TTL	17D	17D	6C
RxD1_TTL	19C	19C	6F
TxD1_TTL	20C	20C	7A
SCL	31C	31C	10F
SDA	32D	32D	11C
TCK	28C	28C	9F
TDI	24C	24C	8B
TDO	25C	25C	8D
TMS	26C	26C	9A
/BRKIN	27D	27D	9B
/BRKOUT	28D	28D	10A
/TRST	29C	29C	10C
ETH_LINKLED	33C	33C	11E
ETH_LANLED	34C	34C	11F
ETH_RxD+	35D	35D	12E
ETH_TxD+	36D	36D	12D
ETH_RxD-	35C	35C	12A
ETH_TxD-	36C	36C	12B

Table 58: Pin Assignment Interface Signals for the phyCORE-XC161 / Development Board / Expansion Board

Signal	**phyCORE-XC161**	**Expansion Bus**	**Patch Field**
/ALE	6A	6A	29D
/CS_ETH	50B	50B	44B
/IRQ_ETH	31D	31D	11A
/IRQ_RTC	33D	33D	11B
/NMI	4A	4A	29A
/PFO	8C	8C	3E
/RD	7B	7B	30A
/RESET	10C, 10D	10C, 10D	3D, 3F
/RSTOUT	11C	11C	4E
/WR/ /WRL	8A	8A	30E
BOOT	9C	9C	3B
RTC_CLKOUT	30D	30D	10B
PFI	7D	7D	2F
WDI	8D	8D	3A

Figure 30: Pin Assignment Control Signals for the phyCORE-XC161 / Development Board / Expansion Board

Signal	phyCORE-XC161	Expansion Bus	Patch Field
VCC	1C, 2C, 1D, 2D	1C, 2C, 1D, 2D	1A, 1C
VCC2	4C, 5C	4C, 5C	2A, 1B
XTAL1	1A	1A	28A
VPD	6D	6D	2D
VBAT	6C	6C	2B
VAREF	50D	50D	17E
VAGND	42C, 47C, 39D, 44D, 49D	42C, 47C, 39D 44D, 49D	connected to GND potential
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, 39D, 44D, 49D, 54D, 59D, 64D, 69D, 74D, 79D	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, 19D, 20D, 21D, 24D, 25D, 26D, 28D, 31D, 32D, 33D, 36D, 37D, 38D, 41D, 42D, 43D, 46D, 47D, 48D, 51D, 52D, 53D, 1E, 2E, 1F

Figure 31: Pin Assignment Power Supply for the phyCORE-XC161 / Development Board / Expansion Board

Signal	phyCORE-XC161	Expansion Bus	Patch Field
NC	35A, 38A, 39A, 37B, 38B, 16C, 30C, 39C, 40C, 4D, 5D, 15D	35A, 38A, 39A, 51A, 53A, 54A, 55A, 56A, 58A, 59A, 60A, 61A, 63A, 64A, 65A, 66A, 68A, 69A, 70A, 71A, 73A, 74A, 75A, 76A, 78A, 79A, 80A 37B.38B, 51B, 52B, 53B, 55B, 56B, 57B, 58B, 60B, 61B, 62B, 63B, 65B, 66B, 67B, 68B, 70B, 71B, 72B, 73B, 75B, 76B, 77B, 78B, 80B 16C, 30C, 39C, 40C 51C, 53C, 54C, 55C, 56C, 58C, 59C, 60C, 61C, 63C, 64C, 65C, 66C, 68C, 69C, 70C, 71C, 73C, 74C, 75C, 76C, 78C, 79C, 80C 4D, 5D, 15D 51D, 52D, 53D, 55D, 56D, 57D, 58D, 60D, 61D, 62D, 63D, 65D, 66D, 67D, 68D, 70D, 71D, 72D, 73D, 75D, 76D, 77D, 78D, 80D	18A, 19A, 20A, 21A, 22A, 23A, 24A, 25A, 26A, 27A, 40A, 45A, 46A, 47A, 48A, 49A, 50A, 51A, 52A, 53A, 54A 5B, 13B, 17B, 18B, 19B, 20B, 21B, 22B, 23B, 24B, 25B, 26B, 27B, 40B, 45B, 46B, 47B, 48B, 49B, 50B, 51B, 52B, 53B, 54B 2C, 20C, 21C, 25C, 26C, 47C, 48C, 52C, 53C 1D, 13D, 17D, 18D, 22D, 23D, 27D, 40D, 44D, 45D, 49D, 50D, 54D 10E, 18E, 19E, 20E, 21E, 22E, 23E, 24E, 25E, 26E, 27E, 39E, 40E, 45E, 46E, 47E, 48E, 49E, 50E, 51E, 52E, 53E, 54E 5F, 17F, 18F, 19F, 20F, 21F, 22F, 23F, 24F, 25F, 26F, 27F, 44F, 45F, 46F, 47F, 48F, 49F, 50F, 51F, 52F, 53F, 54F

Figure 32: Unused Pins on the phyCORE-XC161 / Development Board / Expansion Board

15.3.10 Battery Connector BAT1

The mounting space BAT1 (see PCB stencil) is provided for connection of a battery that buffers the RTC on the phyCORE-XC161. The Voltage Supervisor Chip on the phyCORE-XC161 is responsible for switching from a normal power supply to a back-up battery. The optional battery required for this function (*refer to section 10*) is available through PHYTEC (order code BL-011).

15.3.11 Releasing the /NMI Interrupt

The boot button S1 on the phyCORE Development Board HD200 XC can be routed to the non-maskable interrupt (/NMI) of the XC161 controller with applicable configuration of Jumper JP28 (*also refer to section 15.3.2*).

Jumper	Setting	Description
JP28	7 + 8	Boot button S1 can be used to release the /NMI interrrupt of the XC161 controller

Figure 33: JP28 Releasing the /NMI Interrupt

15.3.12 DS2401 Silicon Serial Number

Communication to a DS2401 Silicon Serial Number can be implemented in various software applications for the definition of a node address or as copy protection in networked applications. The DS2401 can be soldered on space U10 or U9 on the Development Board, depending on the type of device packaging being used.

The Silicon Serial Number Chip mounted on the phyCORE Development Board HD200 XC can be connected to port pin P9.1 of the XC161 available at GPIO1 (JP19 = closed).

Jumper	Setting	Description
JP19	closed	Port pin P9.1 (GPIO1) of the XC161 is used to access the Silicon Serial Number

Figure 34: JP19 Jumper Configuration for Silicon Serial Number Chip

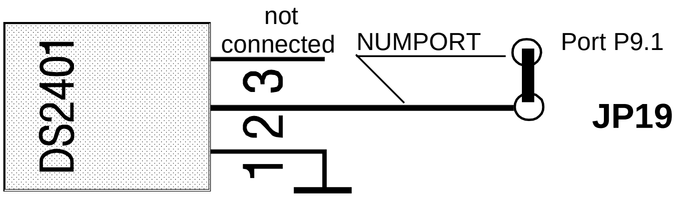


Figure 35: Connecting the DS2401 Silicon Serial Number

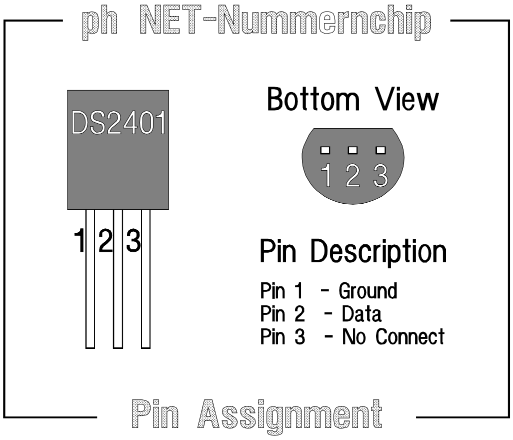


Figure 36: Pin Assignment of the DS2401 Silicon Serial Number

15.3.13 Pin Header Connector X4

The pin header X4 on the Development Board enables connection of an optional modem power supply. Connector X4 supplies 5 VDC at pin 1 and provides the phyCORE Development Board HD200 XC GND potential at pin 2. The maximum current draw depends on the power adapter used. We recommend the use of modems with less than 250 mA current draw.

15.3.14 JP40, S3 Multi-Purpose Push Button Configuration

Push button S3 on the Development Board HD200 can be connected to various input pins of the microcontroller populating the phyCORE module with the help of Jumper JP40. On phyCORE modules featuring an Infineon 16-bit microcontroller this push button can control the controller's NMI input. Push button S3 connects the signal to Ground potential when pushed. A 4.7 kOhm pull-up resistor is used to guarantee a defined high level of the applicable signal line.

The following configurations are possible with JP40:

Multi-Purpose Push Button S3 Configuration	**JP40**
No connection between S3 and any microcontroller input pin	open*
S3 connected with /NMI (BUS5) signal of the XC161 microcontroller	1 + 2
S3 connected with P2.10/EX2IN (BUS4) signal of the XC161 microcontroller	2 + 3

* = Default setting

Figure 37: JP40 Multi-Purpose Push Button S3 Configuration

16 Ethernet Port

The phyCORE Development Board HD200 XC provides a 10-pin header connector at X7 for mounting the PHYTEC Ethernet transformer module. The optional add-on module is available through PHYTEC (order code EAD-001).

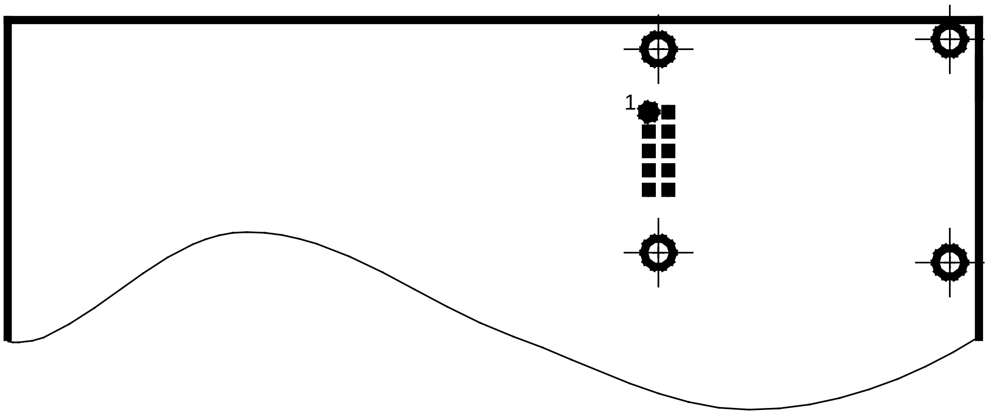


Figure 38: Ethernet Transformer Module Connector

The pinout for the Ethernet transformer connector is shown below:

Pin#	Function	Note
1	ETH_LanLED	Check configuration JP37 on the Development Board!
2	ETH_LinkLED	Check configuration of JP38 on the Development Board!
3	VCC	
4	ETH_TxD+	
5	ETH_TxD-	
6	GND	
7	ETH_RxD+	
8	ETH_RxD-	
9	GND	
10	VCC	

Figure 39: Ethernet Transformer Connector Pinout

The insertable jumpers JP37 und JP38 on the phyCORE Development Board HD200 XC are provided for compatibility reasons in support of the phyCORE-167CR/CS (PCM-009):

Jumper	**phyCORE-XC161**	**phyCORE-XC161 in compatibility mode phyCORE-167CR/CS**
JP37	1 + 2	2 + 3
JP38	1 + 2	2 + 3

Figure 40: Jumper for Ethernet Transformer Port

17 Revision History

Date	**Version numbers**	**Changes in this manual**
13-Nov-2002	Manual L-622e_0 PCM-020 PCB# 1214.0 PCM-997-V2 PCB# 1179.3	First preliminary version.
12-Mar-2003	Manual L-622e_1 PCM-020 PCB# 1214.0 PCM-997-XC PCB# 1179.4	Correct module images added, *Figure 2* and *Figure 5*. Maximum size and address for serial E²PROM corrected. Corrections in section 4.1, System Startup Configuration. Section 13, Debug Interface added. Section 16 adapted to describe Development Board HD200 (PCM-997-XC) with PCB# 1179.4. This revision history table added.
14-May-2003	Manual L-622e_2 PCM-020 PCB# 1214.0 PCM-997-XC PCB# 1179.4	Figure 5 corrected, default setting for J25/J26 added. Corrections in section 3 and 3.9, default settings for J11, J12, J15, J16 changed from 2+3 to 1+2. Corrections in section 5, bus timing description corrected and calculation example added. Corrections in section 14, physical dimensions.
29-Aug-2003	Manual L-622e_3 PCM-020 PCB# 1214.0 PCM-997-XC PCB# 1179.4	Corrections in section 3.15, default settings for J25 and J26 changed from 1+2 to 2+3.
12-Feb-2004	Manual L-622e_4 PCM-020 PCB# 1214.1 PCM-997-XC PCB# 1179.5	New images showing the 1214.1 PCB revision, new figures caption added for *Figure 3* and *Figure 7*. New images showing the 1179.5 PCB revision in *section 15*. Description for new push button S3 and Jumper JP40 added in new *section 15.3.14*.

Index

/

- /NMI 118
- /NMI Interrupt 115

A

- A/D Converter 23

B

- BAT1 115
- Battery Buffer 62
- Battery Connector 115
- Block Diagram 6
- Bootstrap Loader 83

C

- CAN Bus 53
- CAN Interface 30, 53
- CAN Transceiver 31, 53
- CANH 53
- CANL 53
- CANRx 53
- CANTx 53
- Chip Select 25
- Chip Select Signals 45
- Code Fetch Selection 24
- COM Port 52
- Concept of the Development Board 71
- Connector X4 117
- CS8900A 63

D

- Debug Interface 65
- Development Board
  - Connectors and Jumpers 74
- Dimensions 68
- DS2401 115

E

- E^{2}PROM 32
  - Supply Voltage 27
- EMC 1
- Ethernet
  - /SBHE 37
  - Chip Select 37
  - IRQ Signal 38
  - Memory Mode 63
  - Sleep Mode 38
- Ethernet Controller 63
- Expansion Bus 106

F

- Features 5
- First CAN Interface 93
- First Serial Interface 85
- Flash
  - Write Protection 39
- Flash Memory 61
- FRAM
  - Supply Voltage 27
- FRAM
  - Address 29
  - Write Protection 27
- Functional Components on the phyCORE Development Board 80

H

- Humidity 68

I

- I^{2}C Bus 55
- I^{2}C Bus 32
- I^{2}C Interface 58

Initialization Routine 41

J

J1 23
J10 29
J11 30
J12 30
J13 31
J14 31
J15 30
J16 30
J17 31
J18 31
J19 32
J2 23
J20 32
J21 32
J22 34
J25 36
J26 36
J27 37
J28 37
J29 38
J3 23
J30 38
J32 39
J4 25
J5 25
J6 27
J7 27
J8 29
J9 29
JP17 106
JP19 115
JP28 115
JP40 118
JTAG Interface 65
Jumper 17
Jumper Configuration 76
Jumper Settings 22

L

LED D3 106

M

Memory Models 45
Multi-Purpose Push Button 118

O

Operating Temperature 68
Operating Voltage 68

P

P3.0 36
P3.1 36
P3.10 34
P3.11 34
P9.4 32
P9.5 32
Patch Field 106
PCA82C251 53
phyCORE-connector 9, 12
Physical Dimensions 67
Pin Assignment 106
Pin Description 9
Pinout 15
Plug P2A 93
Plug P2B 100
Port 0 42
Power Consumption 68
Power Supply 81
Power Supply to External
  Devices via Socket P1A 88

R

Real-Time Clock 55
Reference Voltage 23
Reset Button 76
RS-232 .... 34, 36
RS-232 Interface 52
RS-232 Level 52

RS-232 Transceiver 52
RTC 32, 62
  Clock Output 34
RTC Interrupt Output 32

S

S3 118
SCL 33, 55
SDA 33, 55
Second CAN Interface 100
Second Serial Interface 90
Serial E^{2}PROM
  Address 29
  Write Protection 27
Serial E^{2}PROM/FRAM 58
Serial Interface 1 34
Serial Interface 2 36
Serial Interfaces 52
Silicon Serial Number 115
SMT Connector 9
Socket P1A (First RS-232) 85
Socket P1B (Second RS-232) 90
SRAM 25
Storage Temperature 68
System Configuration 41
System Initialization 41
System Startup
  Configuration 41

T

Technical Specifications 67
TLE6250 53

U

U1 69
U10 .... 32, 53
U11 53
U13 .... 34, 55
U14 .... 34, 36, 52
U4 61
U6 25
U7 25
U8 .... 32, 58
UART, on-chip 52

V

V_{AGND} 23
V_{AREF} 23
VBAT 62
Voltage Supervisor Chip 62
VPD 62

W

Weight 68

X

X2 65

Document: **phyCORE-XC161**
Document number: **L-622e_4, February 2004**

How would you improve this manual?

____

____

____

Did you find any mistakes in this manual? page

____

____

____

Submitted by:
Customer number: ______________________________

Name: ______________________________

Company: ______________________________

Address: ______________________________

____

Return to:

PHYTEC Technologie Holding AG
Postfach 100403
D-55135 Mainz, Germany
Fax : +49 (6131) 9221-33

Published by

PHYTEC

© PHYTEC Meßtechnik GmbH 2004

Ordering No. L-622e_4
Printed in Germany