phyCORE-P8xC591

Hardware Manual

Edition June 2001

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.

	EUROPE	NORTH AMERICA
Address:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 D-55129 Mainz GERMANY	PHYTEC America LLC 255 Ericksen Avenue NE Bainbridge Island, WA 98110 USA
Ordering Information:	+49 (800) 0749832 order@phytec.de	1 (800) 278-9913 info@phytec.com
Technical Support:	+49 (6131) 9221-31 support@phytec.de	1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	1 (206) 780-9135
Web Site:	http://www.phytec.de	http://www.phytec.com

5th Edition: June 2001

Index of Figures and Tables

Preface

This phyCORE-P8xC591 Hardware Manual describes the board’s design and functions. Precise specifications for the P8xC591 microcontroller 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 for the PHYTEC phyCORE-P8xC591

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.

Note:
PHYTEC products lacking protective enclosures are subject to damage by Electro Static Discharge (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 are longer than 3 m.

PHYTEC products fulfill the norms of the European Union’s Directive for Electro Magnatic Conformity only in accordance to the descriptions and rules of usage indicated in this hardware manual (particularly in respect to the pin header row 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-P8xC591 is one of a series of PHYTEC Single Board Computers (SBCs) 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-P8xC591 belongs to PHYTEC's phyCORE Single Board Computer (SBC) 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 Electro Magnetic Interference (EMI) 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 modules achieve their small size through advanced SMD technology and multi-layer design. In accordance with the complexity of the module, 0402-packaged SMD and laser-drilled Microvias components are used on the boards, providing phyCORE users with access to this cutting edge miniaturization technology for integration into their own design.

The phyCORE-P8xC591 is a subminiature (55 x 40 mm) insert-ready Single Board Computer populated with the Philips P8xC591 microcontroller featuring on-chip 2.0B CAN with extended Philips PeliCAN. Its universal design enables its insertion in a wide range of embedded applications. All controller signals and ports extend from the controller to standard-width (2.54 mm / 0.1 in.) pin header rows 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 Philips 8xC591 controller. No description of compatible microcontroller derivative functions is included, as such functions are not relevant for the basic operation of the phyCORE-P8xC591.

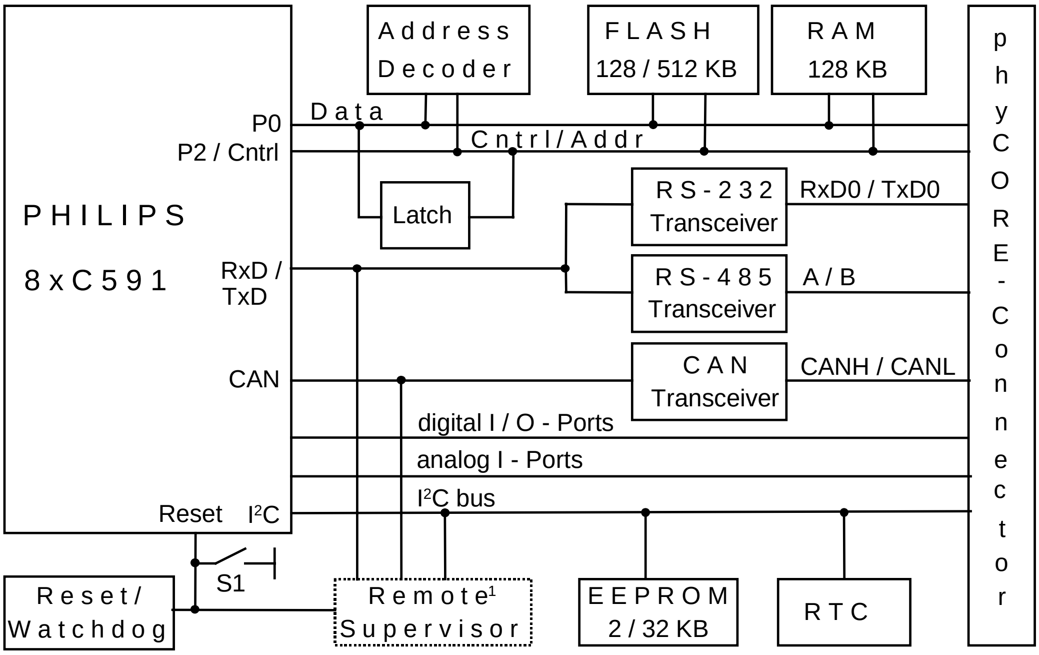
The phyCORE-P8xC591 offers the following features:

- subminiature Single Board Computer (55 x 40 mm) achieved through advanced SMD technology
- populated with the Philips P8xC591 microcontroller (PLCC-44 packaging) featuring 2.0B CAN with extended Philips PeliCAN
- instruction cycle time of 500 ns at 12 MHz clock speed (standard)
- PLCC-socketed controller enables easy emulator connectivity (optional soldered controller)
- improved interference safety achieved through multi-layer PCB technology and dedicated Ground pins
- controller signals and ports extend to standard-width (2.54 mm) pins aligning two sides of the board, enabling it to be plugged like a "big chip" into target applications
- 128 to 512 kByte external Flash on-board[1], enabling In-System Programming (ISP) with PHYTEC FlashTools
- no dedicated Flash programming voltage required through use of 5 V Flash devices
- 128 kByte external SRAM on-board (SMD)
- flexible software-configurable address decoding via a complex logic device
- bank latches for Flash and SRAM integrated in address decoder
- RS-232 or RS-485 interface, user-configurable
- CAN interface with on-board CAN transceiver
- I²C Real-Time Clock with internal quartz
- 2 to 8 kByte I²C EEPROM
- Watchdog device for reset logic and battery control
- Remote Supervisory Circuit[2]
- 3 free Chip Select signals for easy connection of peripheral devices
- requires single 5 V / < 200 mA supply voltage

[1] : North America: Support Hotline: 1-800-278-9913 ● http://www.phytec.com
Europe: Support Hotline: 0 800-0-749-832 ● http://www.phytec.de

[2] : This feature is under development and not available yet.

1.1 Block Diagram



[1]: This feature is under development and not available yet.

Figure 1: Block Diagram

1.2 View of the phyCORE-P8xC591

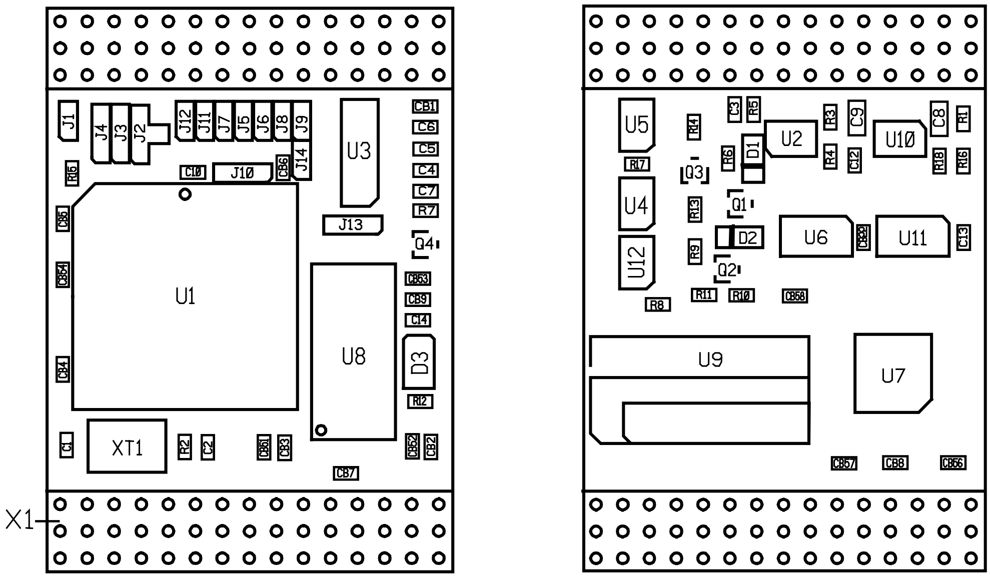


Figure 2: View of the phyCORE-P8xC591

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 User's Manual/Data Sheets located on the Spectrum CD. 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 3* indicates, all controller signals extend to standard-width (2.54 mm / 0.10 in.) pin rows lining two sides the board (referred to as phyCORE-connector). This allows the phyCORE-P8xC591 to be plugged into any target application like a "big chip".

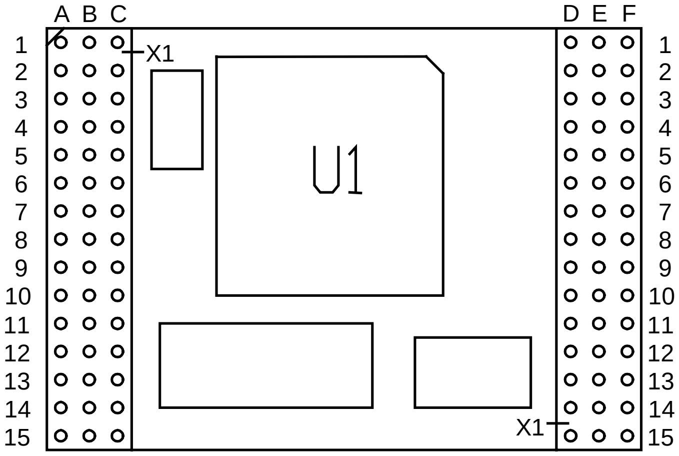


Figure 3: Pinout of the phyCORE-P8xC591 (Top View)

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 receptacle socket on the appropriate PHYTEC phyCORE Development Board LD 5V or your OEM application.

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-P8xC591 (viewed from above; phyCORE-connector header pins pointing down) or with the socket of the phyCORE Development Board LD 5V / target circuitry. The upper left hand corner of the numbered matrix (Pin 1A) is thus covered with the corner of the phyCORE-P8xC591 marked with a white triangle. The numbering scheme is always in relation to the PCB as viewed from above, even if all contacts extend to the bottom of the board.

The numbering scheme is thus consistent for both the module's phyCORE-connector as well as mating connectors on the phyCORE Development Board LD 5V 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's receptacle socket 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 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 (*see Figure 4*) illustrates the numbered matrix system. It shows a phyCORE-P8xC591 mounted on a phyCORE Development Board LD 5V. The shaded area of the phyCORE-connectors shown below indicates the remaining pins not used in conjunction with the phyCORE-P8xC591 which, when plugged onto the Development Board, does not span the entire length of the receptacle socket. The phyCORE Development Board LD 5V can house all phyCORE modules with standard-width (2.54 mm/ 0.10 in.) pin header rows and a maximum of 32 pins per pin header row, A, B, C, D, E and F.

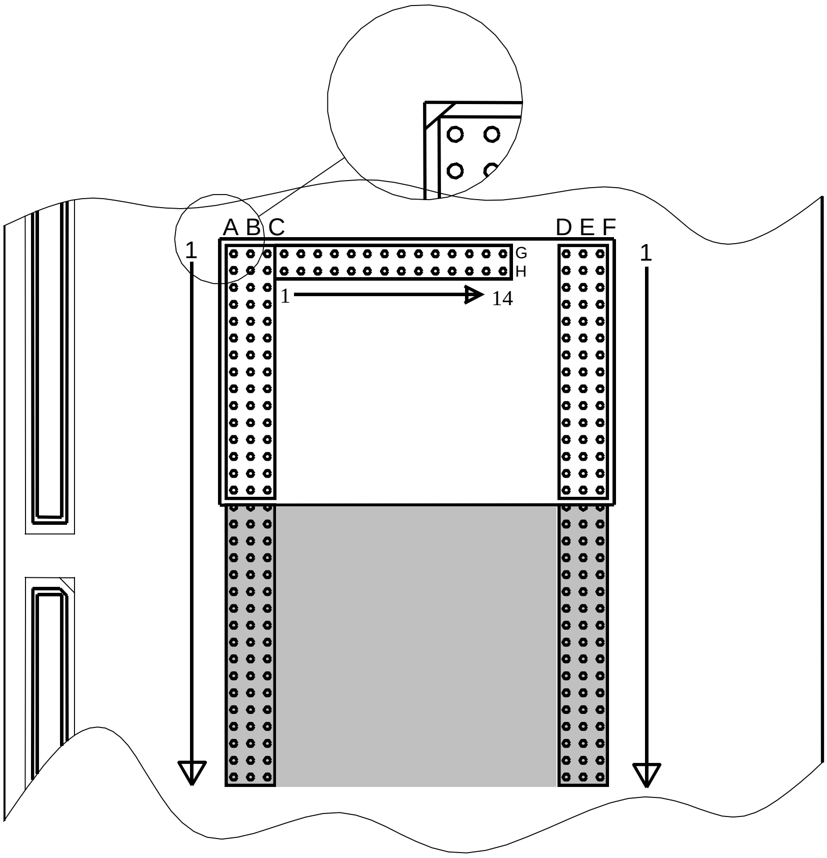


Figure 4: Numbered Matrix Overview of the phyCORE-Connector (Viewed from Above)

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

Table 1 provides an overview of the pinout of the phyCORE-connector and shows possible alternative functions of the pins. *Please refer to the microcontroller User's Manual/Data Sheet for details on the functions and features of controller signals and port pins.*

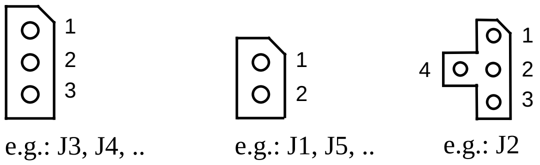
Pin Number	Signal	I/O	Description
Pin Row X1A			
1A	ClkIn	I	Optional external clock generator input connected directly to XTAL1 of µC
2A	P3.3/INT1	I/O	Port pin µC
3A	P3.5/T1	I/O	Port pin µC
4A	/CS2	O	Pre-decoded Chip Select signal #2
5A	/RD	O	/RD signal
6A, 7A, 8A, 9A, 10A, 11A, 12A	A0, A3, A5, A7, A10, A12, A15	O	Address bus from address latch (A0, A3, A5, A7, A10, A12, A15)
13A, 14A, 5A	AD1, AD3, AD6	O	Multiplexed address/data bus from µC
Pin Row X1B			
1B	NC	-	Not used
2B, 3B, 5B, 7B, 8B, 10B, 12B, 13B, 15B	GND	-	Ground 0 V
4B	ALE	O	Address Latch Enable output µC
6B, 9B, 11B	A1, A8, A13	O	Address bus from address latch (A1; A8; A13)
14B	AD4	I/O	Multiplexed address/data bus from µC
Pin Row X1C			
1C, 2C	P3.2 / /INT0, P3.4 / T0	I/O	Port pins µC
3C, 4C	/CS1, /CS3	O	Pre-decoded Chip Select signal #1, #3
5C	/WR, P3.6	I/O	/WR signal µC
6C, 7C, 8C, 9C, 10C, 11C	A2, A4, A6, A9, A11, A14	O	Address bus from address latch (A2; A4; A6; A9; A11; A14)
12C, 13C, 4C, 15C	AD0, AD2, AD5, AD7	I/O	Multiplexed address/data bus µC

Pin Number	Signal	I/O	Description
Pin Row X1D			
1D	VCC	-	Voltage input +5 V =
2D, 3D	NC	-	Not used
4D	VBAT	I	Input for connection to external buffer battery (+)
5D	WDI	I	WDI input of the Reset controller
6D	BOOT	I	Boot = 1 during Reset → starts the Boot sequence
7D, 8D, 9D, 10D, 11D	P1.0 (RxDC), P1.1 (TxDC), P1.3, P1.6 (SCL), P3.0 (RxD)	I/O	Port pins µC
12D, 13D	PWM1, PWM2	O	Port pins µC
14D	CANL	I/O	CANL signal of the CAN transceiver
15D	CANH	I/O	CANH signal of the CAN transceiver
Pin Row X1E			
1E	VCC	-	Voltage input + 5 V
2E, 3E	NC	-	Not used
4E	VPD	O	Voltage output for external buffer
5E, 7E, 8E, 10E, 12E, 13E, 15E	GND	-	Ground 0 V
6E	/RESET	O	Reset output of the module, directly connected with Reset input
9E, 11E	P1.4, P3.1 (TxD)	I/O	Port pins µC
14E	A	I/O	Differential A signal of the RS-485 transceiver
Pin Row X1F			
1F, 2F, 3F	GND	-	Ground 0 V
4F	PFI	I	Power Fail Input of Reset IC
5F	/PF0	O	Power Fail Output of Reset IC
6F	/RESin	I	Reset input of the module
7F	NC	-	Not used
8F, 9F, 10F	P1.2, P1.5, P1.7 (SDA)	I/O	Port pins µC
11F	VAGND	-	Analog GND (connected to GND via J9)
12F	VAREF	-	Reference voltage (connected to VCC via J8)
13F	B	I/O	Differential B signal of the RS-485 transceiver
14F	TxD0	O	Transmit output of the RS-232 transceiver
15F	RxD0	I	Receive input of the RS-232 transceiver

Table 1: Pinout of the phyCORE-Connector X1

3 Jumpers

For configuration purposes, the phyCORE-P8xC591 has 14 solder jumpers, some of which have been installed prior to delivery. *Figure 5* illustrates the numbering of the jumper pads, while *Figure 6* indicates the location of the jumpers on the board. All solder jumpers (Jxx) are located on the top side of the phyCORE-P8xC591.



e.g.: J3, J4, ..   e.g.: J1, J5, ..   e.g.: J2

Figure 5: Numbering of the Jumper Pads

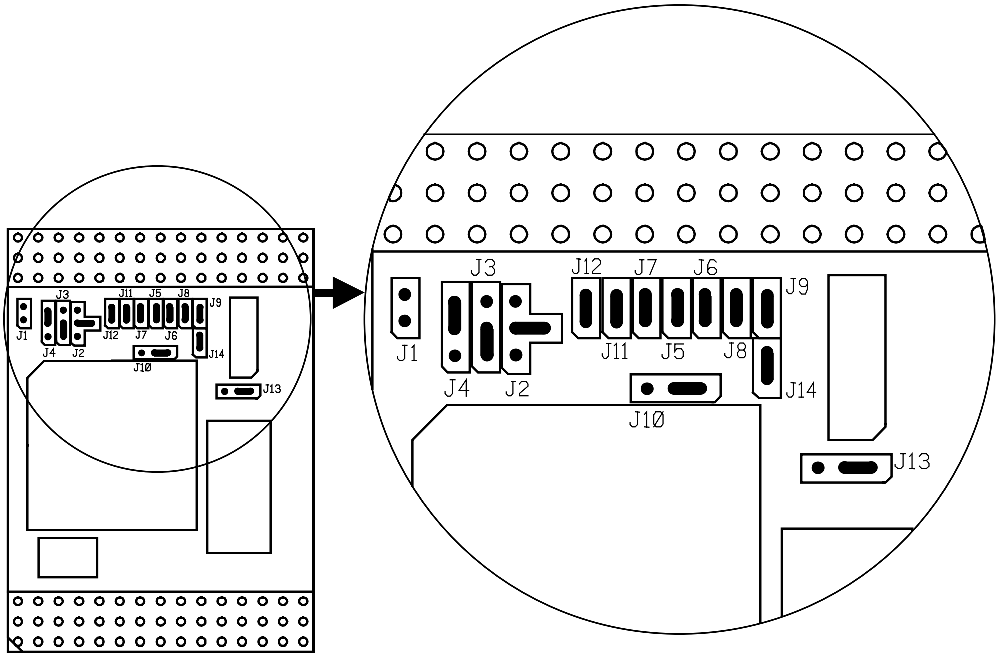


Figure 6: Location of the Jumpers (Top View)

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

	Default Setting		**Alternative Setting**	
J1	(open)	external ROM/ Flash active	(closed)	internal ROM/Flash active
J2	(2+4)	RS-232 / RS-485 for Remote Boot	(2+1) (2+3)	I^{2}C for Remote Boot CAN for Remote Boot
J3	(1+2)	RxD from RS-232	(2+3)	RxD from RS-485
J4	(2+3)	RTC interrupt on /INT1	(1+2)	RTC interrupt on /INT0
J5, **J6**	(closed) (closed)	I^{2}C EEPROM and I^{2}C RTC on controller	(open) (open)	EEPROM and RTC disconnected from controller
J7	(open)	RS-485 not active	(closed)	(bridged with 100 Ω resistor!) P1.5 activates RS-485
J8, **J9**	(closed) (closed)	V_{AREF} and V_{AGND} derived from supply voltage VCC and GND	(open) (open)	V_{AREF} and V_{AGND} derived from external voltage source via phyCORE-connector (X1F12/X1F11)
J10	(1+2)	EEPROM on VCC	(2+3)	EEPROM on VPD
J11, **J12**	(closed) (closed)	on-board CAN transceiver connected to P1.0 and P1.1	(open) (open)	an external CAN transceiver can be attached or P1.0 and P1.1 are available as standard I/O at X1D7 and X1D8
J13	(1+2)	Boot → AD0 pulled high via pull-up resistor R7	(2+3)	Boot → /EA pulled high via pull-up resistor R7
J14	(closed)	on-board RS-232 / 485 transceiver on P3.1	(open)	port P3.1 is available as standard I/O at X1E11

Table 2: Jumper Settings Overview

3.1 J1 Internal or External Program Memory

At the time of delivery, Jumper J1 is open. This default configuration means that the program stored in the external program memory (Flash) is executed after a hardware reset. In order to allow the execution of any code stored in the controller's on-chip memory, Jumper J1 must be closed.

The following configurations are possible:

Code Execution	**J1**
from external program memory	open*
from internal program memory	closed

*= Default setting

Table 3: J1 Access to External or Internal Program Memory

3.2 J2 Remote Download Source

Space U8 on the module is intended to be populated by a Remote Supervisory Chip[1]. This IC can initiate a boot sequence via various serial interfaces (*refer to section 10*). Jumper J2 is reserved for future use and remains open as default.

The following configurations are possible:

Download Source	**J2**
Not available	open*
RS-232 / RS-485 – RxD	2 + 4
I²C – SDA	1 + 2
CAN – RxD	2 + 3

*= Default setting

Table 4: J2 Remote Download Source Configuration

[1]: The Remote Supervisor IC is under development. Hence Jumper J2 has no function at this time.

3.3 J3 Serial Interface

Jumper J3 connects the controller RxD0 input port P3.0 to the on-board RS-232 or RS-485 transceiver. Optionally, P3.0/RxD can be used as standard I/O at pin X1D11 of the phyCORE-connector. This requires opening Jumper J3.

The following configurations are possible:

P3.0 - Serial Interface	**J3**
P3.0 as RS-232	1 + 2*
P3.0 as RS-485	2 + 3
P3.0 as port pin	open

*= Default setting

Table 5: J3 Serial Interface Configuration

3.4 J4 Interrupt Output of the RTC

Jumper J4 determines if the interrupt output of the RTC (U11) extends to port 3.2 or to port 3.3. Alternatively, these port pins can be used as standard I/O signals at pins X1C1 and X1A2 of the phyCORE-connector. This requires opening Jumper J4.

The following configurations are possible:

Interrupt Output of the RTC	**J4**
Interrupt output is connected to port 3.2 (/INT0)	1 + 2
Interrupt output is connected to port 3.3 (/INT1)	2 + 3*
P3.2 and P3.3 available as port pins	open

*= Default setting

Table 6: J4 RTC Interrupt Configuration

3.5 J5, J6 Configuration of P1.6 and P1.7 for I²C Bus

Two I²C interface devices - a Real-Time Clock (RTC) at U11 and an EEPROM at U10 - are available on the phyCORE-P8xC591. These devices are connected via Jumpers J5 and J6 to port pins P1.6 and P1.7. Use of these pins as standard I/O requires opening the corresponding jumpers. *Refer to section 7* and *section 8 for details on these I²C interface devices.*

The following configurations are possible:

Configuration P1.6 / P1.7	J5	J6
Port P1.7 used as I/O pin		open
Port P1.7 used as I²C SDA		closed*
Port P1.6 used as I/O pin	open	
Port P1.6 used as I²C SCL	closed*	

*= Default setting

Table 7: J5 and J6 I²C Interface Configuration

3.6 J7 RS-485 Interface Control

The transmission circuit of the RS-485 transceiver (U4) is disabled at time of delivery through a pull-up resistor (R14). Closing Jumper J7, using a 100 Ω / 0805-shape resistor, enables control of the transceiver IC's transmit function via port 1.5 (P1.5 High → RS-485 inactive, P1.5 Low → RS-485 active).

The following configurations are possible:

Configuration RS-485 Interface Control	J7
P1.5 High →RS-485 transmitter inactive P1.5 Low →RS-485 transmitter active	closed
RS-485 transmitter inactive / P1.5 as port pin	open*

*= Default setting

Table 8: J7 RS-485 Interface Control Configuration

3.7 J8, J9 A/D Reference Voltage

The A/D converter of the phyCORE-P8xC591 requires a reference voltage ($V_{AREF,}$ V_{AGND}) supplied to the controller at pin 44 and pin 1. The source of the reference voltage can be chosen with Jumpers J8 and J9.

The following configurations are possible:

A/D Reference Voltage	**J8**	**J9**
External reference voltage (V_{AREF} at X1F12, V_{AGND} at X1F11)	open	open
VCC as V_{AREF}	closed*	
GND as V_{AGND}		closed*

*= Default settings

Table 9: J8 and J9 A/D Reference Voltage Configuration

3.8 J10 EEPROM Supply Voltage

The device at U10 can be connected to VCC or VPD using Jumper J10. As default, U10 is populated with a serial EEPROM with voltage supply pins connected to VCC. Alternatively, a serial FRAM device can also populate U10, 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 U10	**J10**
U10 supplied with VCC	1 + 2*
U10 supplied with VPD	2 + 3

*= Default setting

Table 10: J10 EEPROM Supply Voltage Configuration

3.9 J11, J12 CAN Interface

The CAN interface of the P8xC591 is available at port P1.0 and P1.1. These signals extend to the CAN transceiver populating U5 (e.g. PCA82C251), which generates the signals CANH (Pin X1D15) and CANL (Pin X1D14). These signals can be directly connected to a CAN bus using a dual-wire cable. This requires that Jumpers J11 and J12 are closed.

Direct access to the signals RxDC/P1.0 and TxDC/P1.1 is also available at the module's X1D7 and X1D8 pins if solder Jumpers J11 and J12 are open. This enables use of either an external CAN transceiver or use of P1.0 or P1.1 as standard I/O's.

For detailed descriptions of the CAN interface *please refer to the appropriate controller User's Manual from Philips, as well as to the accompanying CAN transceiver Data Sheet.*

The following configurations are possible:

CAN Transceiver	**J11**	**J12**
on-board CAN transceiver	closed*	closed*
External CAN transceiver or P1.0 and P1.1 as I/O	open	open

*= Default setting

Table 11: J11 and J12 CAN Interface Configuration

3.10 J13 Boot Sequence Selection

A boot sequence can be initiated through the module’s Boot pin or the Remote Supervisory IC. Solder Jumper J13 (1+2) applies pin AD0 to high-level (required for starting PHYTEC FlashTools) or (2+3) pin /EA to VCC (for execution of any code stored in on-chip memory).

The following configurations are possible:

Boot Sequence Selection	**J13**
AD0 to pull-up resistor	1 + 2*
/EA to VCC	2 + 3

*= Default setting

Table 12: J13 Boot Sequence Configuration

3.11 J14 P3.1 as TxD Signal

Jumper J14 determines whether P3.1 connects as a TxD signal line to the on-board RS-232 or RS-485 transceiver, or whether it is available as dedicated port pin P3.1 at pin X1E11 of the phyCORE-connector.

The following configurations are possible:

P3.1 Configuration	**J14**
P3.1 connected to the on-board RS-232 or RS-485 transceiver	closed*
P3.1 available for free use on X1E11	open

*= Default setting

Table 13: J14 Port P3.1 Configuration

4 Memory Models

The phyCORE-P8xC591 allows for flexible address decoding which can be configured by software to different memory models. A hardware reset activates a default memory configuration that is suitable for a variety of applications. However, this memory model can be changed or adjusted at the beginning of a particular application.

Configuration of the memory is done within the address decoder by means of 4 internal decoder registers: two Control Registers, one Address Register and one Mask Register. All registers are carried out as write-only registers with access through the controller's XDATA memory space. There are two distinct address areas - selectable by means of the bit IO-SW in Control Register 1 - by which the registers can be accessed (*refer to the description of the bit IO-SW below*). Due to a lack of read access, a copy of all register contents should be maintained within your application. Reserved bits may not be changed during the writing of the register; contents must remain at 0. A hardware reset erases all registers while preserving the configuration of the default memory model.

Note:
In the event that you use FlashTools – PHYTEC's proprietary firmware allowing convenient on-board Flash programming - the address FA16 is preset at the start of your application software (*refer to section 4.1, "Control Register 1"*). This is to be noted upon installation of the software copy of the register contents.

The following figure illustrates the default memory model:

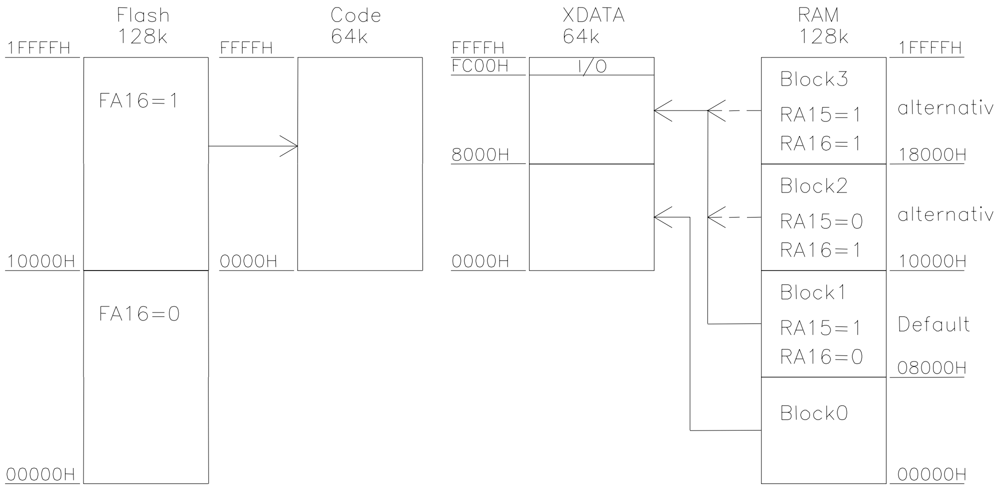


Figure 7: Default Memory Model following a Hardware Reset

As the phyCORE-P8xC591 is populated with a 128 kByte SRAM device, blocks of 32 kByte each of the upper 96 kByte can be accessed and switched via bank latching. The corresponding I/O area is mapped to the XDATA memory space. Within this I/O area; there is no access to any available RAM.

The following sections describe the address decoder’s registers for configuration of the memory model.

4.1 Control Register 1

Control Register 1 (Address 7C00H / FC00H)							
Bit 7							Bit 0
PRG-EN	IO-SW	RAM-SW	VN-EN	FA18	FA17	FA16[1]	FA15
Default Values:							
Reset Value:				0000 0000 b			
Runtime Model:				0000 0010 b			

Table 14: Control Register 1 of the Address Decoder

Bit invalid in programming model (refer to PRG-EN)

Bit valid only in programming model (refer to PRG-EN)

PRG-EN: Can be used to activate the special Flash programming memory model (PRG-EN = 1). This model is used within FlashTools[2] for Flash programming purposes and is of limited use within user applications because of its special restrictions.

In this model, 32 kByte of RAM located within the address area 0000H - 7FFFH is accessible, as well as 32 kByte of Flash memory within the address area 8000H - FFFFH. The Flash memory can only be written in the XDATA memory space and can only be read from the CODE memory space. The RAM can be read and written in the XDATA memory space. RAM can also be read from the CODE memory space.

The address line A15 of the Flash is derived from the Control Register 1 (Bit 0, FA15) only in the programming model. In the runtime configuration (PRG-EN = 0), the address line A15 of the controller leads directly to the Flash device.

[1]: If using FlashTools - a firmware allowing convenient on-board Flash programming - it should be noted that the bit FA16 will be preset at the start of user code. This is to be noted upon installation of the software copy of the register contents.

[2]: The FlashTools firmware is pre-installed in the external Flash device upon delivery of the module.

The bit IO-SW is also relevant to the programming model, whereas the bit VN-EN is not relevant. The following figure illustrates the programming model (the I/O area is not represented):

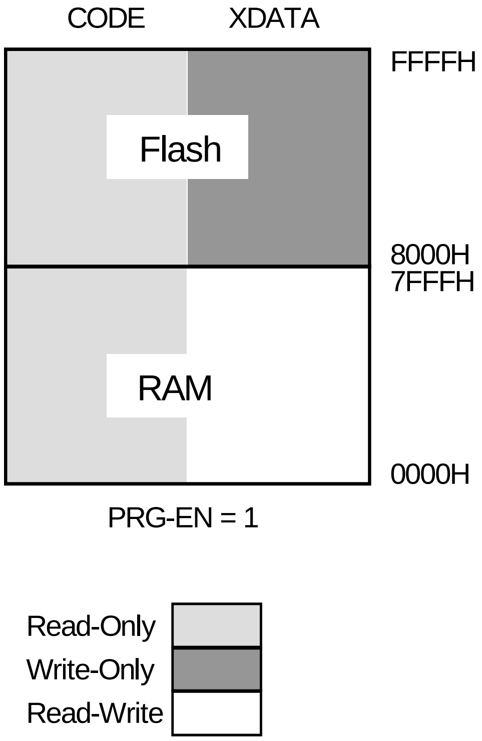


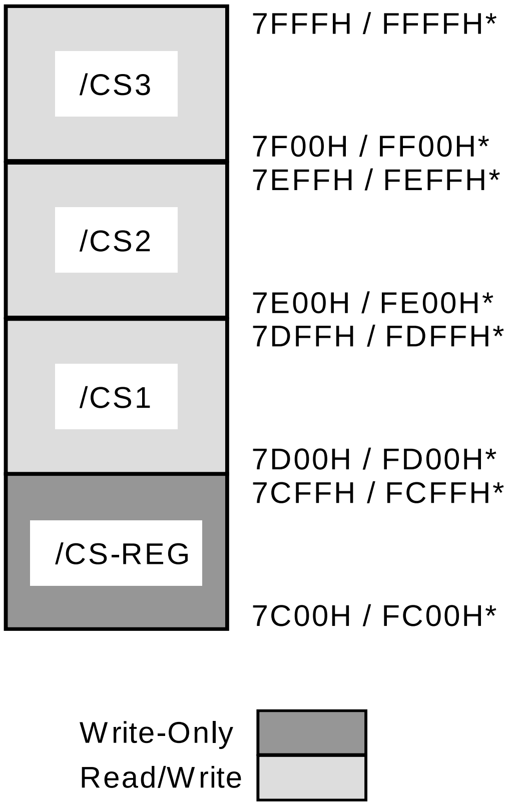
Figure 8: Memory Model for Flash Programming

IO-SW: By means of this bit, the I/O area of the module can be selectively mapped either to the upper or to the lower 32 kByte of the address space. With IO-SW = 0 following a hardware reset, the I/O area is accessible in the range between FC00H - FFFFH. Setting bit IO-SW = 1 maps the I/O area to 7C00H - 7FFFH.

This I/O area generally consists of 4 blocks of 256 bytes each. In three of these blocks, the address decoder provides a pre-decoded Chip Select signal that simplifies the connection of peripheral hardware to the module.

These Chip Select signals will be activated on read/write access to the XDATA memory space within the appropriate address range. The fourth block is reserved for internal access to the decoder's internal register (write-only access). This block is **not** available for use of connecting external devices.

The I/O area configuration is shown in the picture below:



* = Default Setting

Figure 9: Configuration of the I/O Area

The areas referred to as /CS1 to /CS3 are the freely available Chip Select signals; while the signal /CS-REG is the decoder's internal signal. This latter signal is required to access the internal registers. This signal is not available to the user and no external circuitry should be connected within the address area valid for the /CS-REG signal. In order to ensure proper functioning of FlashTools[1] firmware, enabling on-board programming of the Flash memory, it is essential that the /CS-REG signal be used as described herein. These internal registers are located at address 7C00H - 7C03H (IO-SW = 1) or FC00H - FC03H (IO-SW = 0). The rest of the /CS-REG block remains unused and is reserved for future expansion.

RAM-SW: This bit enables exchange of a 32 kByte memory portion of RAM block 0 and RAM block 1 - 3. Following a hardware reset (RAM-SW = 0) the RAM block 0 is mapped in the XDATA address area from 0000H - 7FFFh and one of RAM blocks 1 - 3 is addressable from 8000H to FFFH. Setting bit RAM-SW = 1 enables access to RAM block 0 in the address area 8000H - FFFFH. Likewise, access to RAM block 1 is possible in the address area 0000H - 7FFFH. In the corresponding I/O areas, there is no access to any memory device.

[1]: Firmware portion of the utility program for on-board Flash programming and is pre-installed in the Flash at time of delivery.

VN-EN: This bit enables free selection of von Neumann memory[1] within the address space of the controller. Following a hardware reset, the Harvard[2] architecture is configured as default. The von Neumann memory is especially useful when programming code is to be downloaded and subsequently run during runtime, as is the case with a Monitor program. The location of the optional von Neumann memory areas is defined by the Address and Mask Registers (*see below*).

Following a hardware reset (VN-EN = 0), the settings in the Address and Mask Registers are not released. The von Neumann memory is not available at this time. Setting bit VN-EN = 1 activates the Address and Mask Registers and incorporates their settings into access control for von Neumann memory areas. This bit is only relevant in the runtime model (PRG-EN = 0). In the programming model (PRG-EN = 1) bit VN-EN is unimportant and will be ignored.

[1]: Memory space in which no difference is made between CODE and XDATA access. This means that both accesses use the same physical memory device, usually a RAM.

[2]: Memory space in which CODE and XDATA accesses use physical different memory devices. CODE access typically uses a ROM or Flash device, whereas XDATA access uses a RAM.

FA[18..15]: The phyCORE-P8xC591 can be optionally populated with a Flash device of 512 kByte capacity. Because of the limited 64 kByte address space of the P8xC591 microcontroller, the remainder of the Flash memory can only be accessed by bank switching.

In the runtime model (PRG-EN = 0), 64 kByte banks can be switched by controlling the upper address lines A[18..16] for the Flash through software. For this purpose, register bits FA[18..16] of the address decoder provide a latch to which the desired upper addresses can be written.

Of particular note is the bit FA15, which is solely relevant in the programming model (PRG-EN = 1). As in this model only 32 kByte of Flash can be accessed, it serves as address line A15 for the Flash memory. In the runtime model (PRG-EN = 0) with a 64 kByte Flash memory area, to contrast, the address line A15 of the controller is attached directly to the Flash.

The function of the bits FA[18..16] depends on the hardware configuration of the module. As described above, these bits are only relevant if the phyCORE-P8xC591 is populated with a Flash device of 512 kByte capacity.

4.2 Control Register 2

Control Register 2 (Address 7C01H / FC01H)							
Bit 7							Bit 0
N/A[1]	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	RA16	RA15
Default Values:							
Reset Value:				0000 0001 b			
Runtime Model:				0000 0001 b			

Table 15: Control Register 2 of the Address Decoder

RA16: The module accommodates a 128 kByte RAM device. As the address space is limited to 64 kByte in the XDATA memory space of the controller, the remainder of the RAM can only be accessed by bank switching.

Three memory banks of 32 kByte banks can be switched by setting the high address lines A[16..15] through software. For this purpose, register bits RA[16..15] of the address decoder provides a latch to which the desired upper addresses can be written.

The purpose of these bits depends on the hardware configuration of the module. As described above, these bits are only relevant if the phyCORE-P8xC591 is populated with a RAM device of 128 kByte capacity.

The configuration RA16 = 0 and RA15 = 0 is reserved for the connection of further memory devices, but is not supported on this module.

[1]: N/A: Not Accessible

4.3 Address Register

The Address Register 7C02H / FC02H functions in conjunction with the Mask Register (*see section 4.4*) to define the von Neumann[1] and Harvard[2] memory area in the controller's memory space. By setting the bit VN-EN in Control Register 1, the values of the Address and the Mask Register become valid for the definition of von Neumann and Harvard memory areas and will be incorporated in address decoding. (*refer to section 4.1, "Control Register 1"*)

The location of one or more Harvard memory areas can be configured with both registers. The remaining areas of the memory space are configured as von Neumann memory in which RAM is accessible in both XDATA and CODE memory space.

The mechanism for the memory space distinction is based on a comparison of the current address with a pre-defined address pattern of variable width. If the relevant bit positions of the address matches the pre-defined address pattern, memory access occurs according to the Harvard architecture. If the current address is different to the pre-defined address pattern, memory access occurs according to the von Neumann architecture.

Address Register (Addresses 7C02H / FC02H)							
Bit 7							Bit 0
HA15	HA14	HA13	HA12	*Res.*[3]	*Res.*	*Res.*	*Res.*
Reset Value:	0000 0000 b						

Table 16: Address Register of the Address Decoder

[1]: Memory space in which no difference exists between CODE and XDATA access. This means that both accesses use the same physical memory device, usually a RAM.

[2]: Memory space in which CODE and XDATA accesses use different physical memory devices, usually CODE access uses a ROM or Flash device, whereas XDATA access uses a RAM.

[3]: Reserved bits are not to be changed, the default value (0) must remain.

The Address Register holds the address pattern mentioned above. Each bit of the pattern is compared with the corresponding address line of the controller (HA15 with A15, ..., HA12 with A12). As address lines A15 .. A12 are used to define Harvard memory space, only Harvard areas of at least 4 kByte can be configured. Memory areas smaller than 4 kByte can not be configured.

4.4 Mask Register

The Mask Register (7C03H / FC03H) can be used to mask single bits in the Address Register (see above). Following a hardware reset, all bits within the Address Register are relevant. By setting the individual bits in the Mask Register, all corresponding bits in the Address Register will no longer be regarded in the address comparison.

Mask Register (Address 7C03H / FC03H)							
Bit 7							Bit 0
MA15	MA14	MA13	MA12	*Res.*[1]	*Res.*	*Res.*	*Res.*
Reset Value:				0000 0000 b			

Table 17: Mask Register of the Address Decoder

[1] : Reserved bits are not to be changed, the default value (0) must remain.

The following examples of different combinations of the Address and Mask Registers illustrate these functions (only A15 to A8 are shown):

Address Reg.	Mask Reg.	Comments (only for VN-EN = 1)	
1XXX0000 B	01110000 b	Harvard von Neumann	8000H - FFFFH 0000H - 7FFFH
0XXX0000 B	01110000 b	Harvard von Neumann	0000H - 7FFFH 8000H - FFFFH
11110000 B	00000000 b	Harvard von Neumann	F000H - FFFFH 0000H - EFFFH
01X00000 B	00100000 b	Harvard von Neumann	4000H - 4FFFH 6000H - 6FFFH 0000H - 3FFFH 5000H - 5FFFH 7000H - FFFFH
10000000 B	00000000 b	Harvard von Neumann	8000H - 8FFFH 0000H - 7FFFH 9000H - FFFFH
101X0000 B	00010000 b	Harvard von Neumann	A000H - BFFFH 0000H - 9FFFH C000H - FFFFH

Table 18: Example of Address Decoder Functions

Reserved bits without function for address decoding (*refer to description of the register*).

X = irrelevant (on account of a bit set in the Mask Register)

The last example from the above table is further illustrated by the following figure:

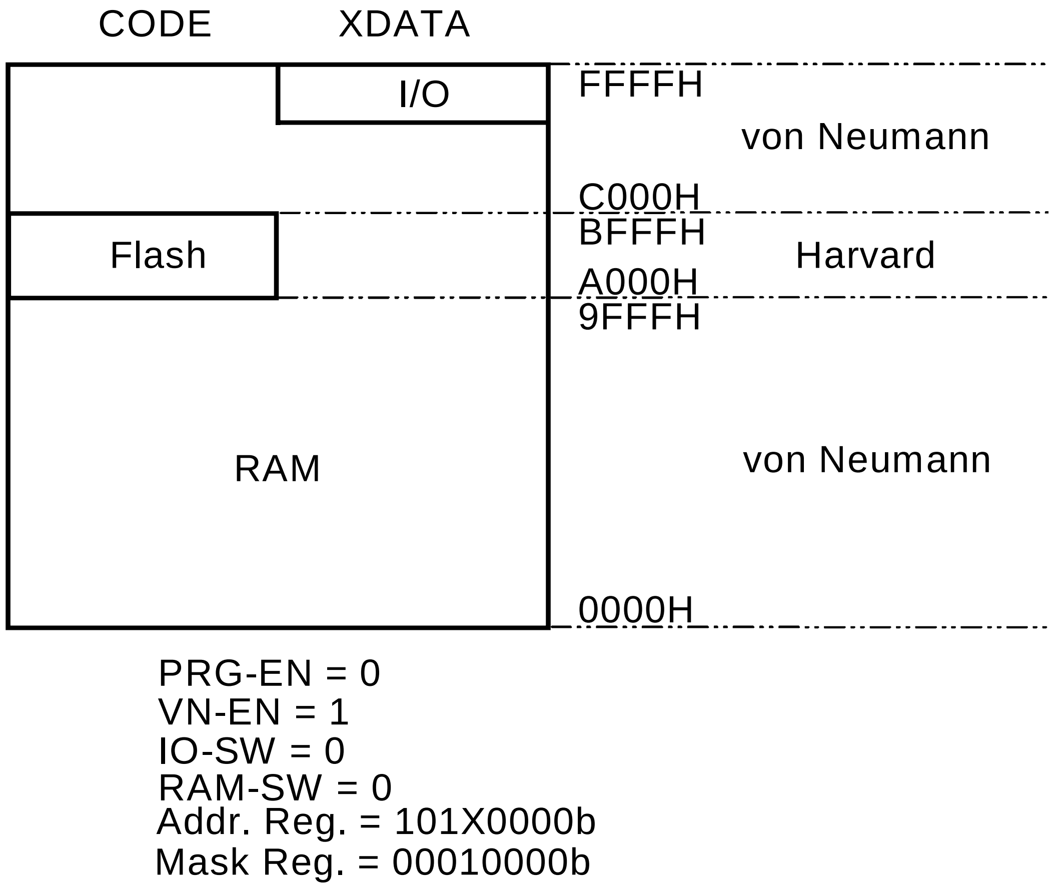


Figure 10: Example of a Memory Model

Following a hardware reset, the memory space is configured as Harvard memory. Only after setting the bit (VN-EN = 1), the settings in the Address and Mask Registers are valid and regarded in the address decoding.

5 Serial Interfaces

5.1 RS-232 Interface

An RS-232 transceiver is located on the phyCORE-P8xC591 at U3. This device adjusts the signal levels of the P3.0/RxD0 and P3.1/TxD0 lines. The RS-232 interface enables connection of the module to a COM port on a host-PC. In this instance, the RxD0 line (X1F15) of the transceiver is connected to the TxD line of the COM port; while the TxD0 line (X1F14) is connected to the RxD line of the COM port. The ground circuitry of the phyCORE-P8xC591 is also connected to the applicable ground pin on the COM port.

The microcontroller's on-chip UART does not support handshake signal communication. However, depending on user needs, handshake communication can be replicated 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.

5.2 RS-485 Interface

As an alternative to the RS-232 interface, an RS-485 interface can be configured on the phyCORE-P8xC591 using the lines P3.0/RxD and P3.1/TxD. Jumpers J3 and J14 enable selection between RS-232 and RS-485 interfaces (*see sections 3.3 and 3.11*).

The RS-485 transceiver (U4) supports up to 32 nodes in one bus system. Data transmission occurs via differential signal levels according to RS-485 interface standards.

Note:
To utilize the RS-485 interface, Jumper J7 must be bridged with a 100 Ω / 0805 resistor. This enables control of the transmit function on the RS-485 transceiver IC via port 1.5 (*refer to section 3.6*).

5.3 CAN Interface

A CAN transceiver is populated at U5 on the phyCORE-P8xC591 module. This transceiver enables transmission and receipt of CAN signals via CANTx and CANRx, respectively. 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 terminating resistor must be connected to each end of the CAN bus between the pins delivering the CANH and CANL signals.

For larger CAN networks, an external optocoupler should be implemented to galvanically separate the CAN bus signals and the phyCORE-P8xC591 circuitry. This requires that the CANTx and CANRx lines be separated from the on-board CAN transceiver by opening Jumpers J11 und J12 (*refer to section 3.9*). The Hewlett Packard HCPL06xx or Toshiba TLP113 HCPL06xx optocoupler is recommended. Parameters for configuring a proper CAN bus system are found in the DS102 norms from the CiA[1] (CAN in Automation) User and Manufacturer's Interest Group.

[1]: CiA CAN in Automation – International User‘s and Manufacturer's Union, founded in March 1992. CiA offers technical, product- and market-related information on the topic of Controller Area Network, with the goal of increasing general knowledge about CAN and furthering future development of the CAN protocol.

6 Flash Memory (U8)

Flash, as non-volatile memory on the phyCORE-P8xC591, provides an easily reprogrammable means of code storage to the user. The phyCORE-P8xC591 can be populated at U8 by a single Flash device of type 29F010 with two banks of 64 kByte each or device type 29F040 with 8 banks of 64 kByte each.

Flash memory devices offer up to 100,000 reprogramming cycles, and enable on-board programming of user code. These Flash devices are programmable with 5 V. No dedicated programming voltage is required. All standard versions of the phyCORE-P8xC591 feature a programming utility firmware – FlashTools (*refer to applicable QuickStart Instruction for more details*) – resident in the Flash device.

This firmware enables on-board download, as well as subsequent erasure and reprogramming, of user code into the Flash with the help of an intuitive PC-side software. The FlashTools firmware portion resides in the initial 32 kByte of Flash memory, which is not available for storage of user code. The total memory available for user programs is 64 kByte (29F010) or 448 kByte (29F040) *(refer to Figure 11)*.

Note:
Should the FlashTools firmware portion be erased from the Flash device without having a back-up or an equivalent replacement, reprogramming is no longer possible!

Please note that this firmware protects itself against any intentional or accidental erasure or overwriting. As the Flash device’s hardware protection mechanism is not utilized, protection is limited to the software level. In the event that a user wishes to download his or her own programming algorithms or tools into the Flash, the user must ensure that a programming tool remains in the Flash memory. *Refer to the “QuickStart Instructions" for a detailed description of the on-board programming procedure.*

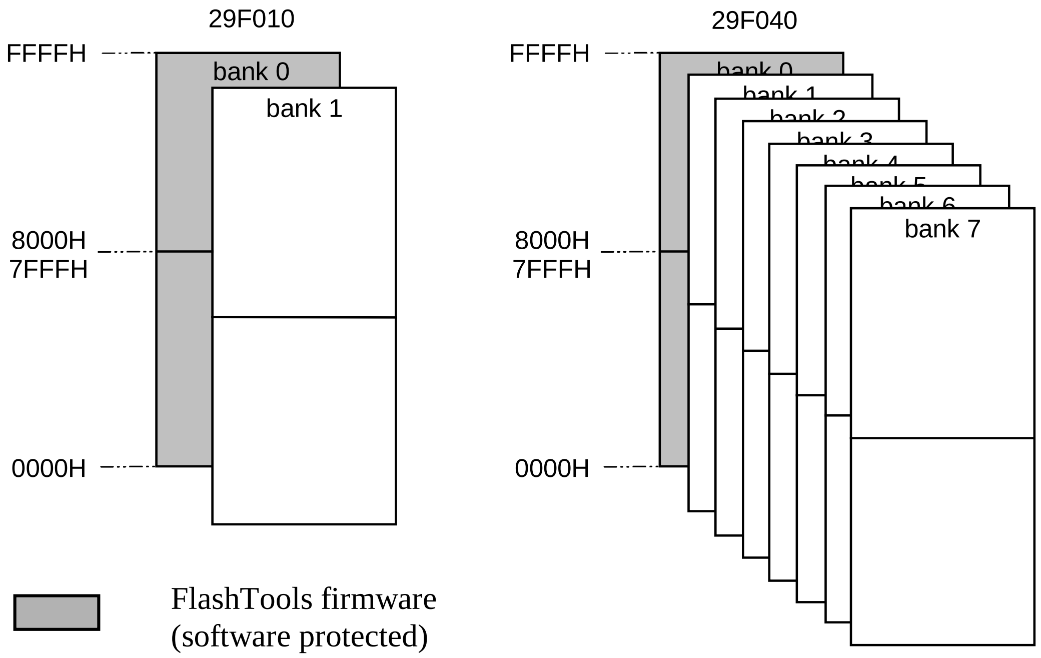


Figure 11: Memory Areas of the Flash Device

Use of a Flash device as the only code memory results in limited usability of the Flash as non-volatile memory for data. This is due to the internal structure of the Flash device as, during the Flash’s internal programming process, the reading of data from Flash is not possible. 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.

7 Serial EEPROM (U10)

A non-volatile memory with a serial (I²C bus) interface populates space U10 on the phyCORE-P8xC591. This device is intended to store configuration parameters and user data. This memory device can be in the form of an EEPROM device or an FRAM device. The I²C bus is generated using port pins P1.6 (SCL) and P1.7 (SDA). By opening the two solder Jumpers J5 and J6, the I²C bus can be disconnected from the controller pins. In this case, these pins are available as standard I/O's.

In the default configuration, an EEPROM is populated on U10. With approximately 10^6 write and erase cycles, an EEPROM memory device is a reliable solution for most requirements. For applications that require frequent and fast storage of a large amount of data, other memory devices can populate U10. Modern I²C FRAMs with approximately 10^{10} write and erase cycles can be used for this purpose. These ferro-electrical memory devices can store data even if no power is supplied to the board. If populated with either an EEPROM or FRAM, the solder Jumper J10 should remain in postion 1+2 (connection to VCC).

As an option, use of serial SRAMs (if available) is also possible. Jumper J10 must be set to position 2+3 for this purpose. This enables buffering of the memory device with a battery, which must be connected to the module's VBAT input (*refer also to section 11, "Battery Buffer"*).

Memory Device Addressing Scheme:

The address lines A0 (IC pin 1) and A1 (IC pin 2) are connected to GND. Address line A2 (IC pin 3) is connected to VCC. The address configuration for the memory devices is shown in the table below:

Device Type	**Capacity**	**Type**	**Address**
EEPROM	4 kByte	Catalyst 24WC32	1010100
EEPROM	8 kByte	Catalyst 24WC64	1010100
FRAM	512 Byte	Ramtron FM24C04	101010x
FRAM	8 kByte	Ramtron FM24C64	1010100

Table 19: Memory Device Options at U10

8 Real-Time Clock RTC-8563 (U11)

For real-time or time-driven applications the phyCORE-P8xC591 is equipped with an RTC-8563 Real-Time Clock (RTC) at U11. 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-P8xC591 is equipped with a battery, the Real-Time Clock runs independently of the module's power supply.

Programming of the Real-Time Clock is done via the I^2C bus (I^2C address 1010001), connected to port P1.6 (SCL) and port P1.7 (SDA) on the controller. The Real-Time Clock also provides an interrupt output which extends to port P3.2 or P3.3 via Jumper J4. An interrupt occurs in case of a clock alarm, timer alarm, timer overflow and event counter alarm. All interrupts must then 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-8563, refer to the corresponding Data Sheet located on the Spectrum CD.*

Note:
Following attachment of a power supply to the board, the RTC generates **no** interrupts, as the RTC is not yet initialized.

9 Reset Controller (U2)

The Reset controller at U2 is used to generate a definite release of a Reset signal if the supply voltage VCC drops below 4.65 V. This ensures the proper start-up of the microcontroller. Furthermore, the Reset controller can switch the voltage of a back-up battery as VPD to several IC's in case the main power supply becomes interrupted. The basic characteristics of this controller are described in the appropriate Data Sheet, which is available on the Spectrum CD.

All pins of the Reset controller are routed to the phyCORE-connector. The VPD voltage is available on the OUT pin of the Reset controller. In normal operation mode, this pin is supplied by VCC (via a diode). Additionally, VBAT is routed via the voltage divider R3/R4 to pin PFI. If VBAT = 3.3 V, a voltage of 1.65 V is available at PFI. If the voltage at PFI drops below 1.25 V, the signal /PFO is released. The signals WDI and /PFO are available at the phyCORE-connector pins X1D5 and X1F5.

10 Remote Supervisory Chip (U12)

Space U12 is intended to be populated by an RSC1308 Remote Supervisory Chip. This IC can initiate a boot sequence via a serial interface, such as RS-232, RS-485 or I²C. The RSC can start PHYTEC FlashTools without requiring a manual reset of the phyCORE module via a Boot jumper or button. This enables a remote software update of the on-board Flash device.

The Remote Supervisory Chip is under development and not available yet. This feature will be available on future boards.

11 Battery Buffer

The battery that buffers the memory is not essential to the functioning of the phyCORE-P8xC591. However, this battery buffer embodies an economical and practical means of storing nonvolatile data in SRAM and is necessary for data storage in the Real-Time Clock in case of a power failure.

The VBAT input (pin X1D4) is provided for connecting the external battery. The negative polarity pin on the battery must connect to GND on the phyCORE-P8xC591. 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 SRAM (U9), the RTC (U11) and an optional serial SRAM at U10 (Jumper J10 closed at 2+3) will be buffered by a battery connected to VBAT.

Power consumption depends on the components used and memory area. Refer to corresponding Data Sheets for the RAM devices mounted on the board (*refer also to section 12,"Technical Specifications").*

Note:
Be advised that despite the battery buffer, changes in the data content within the RAM can occur. The battery buffer does not completely remove the danger of data destruction.

12 Technical Specifications

The physical dimensions of the phyCORE-P8xC591 are represented in *Figure 12*. The module's profile is approximately 11 mm thick, with a maximum component height of 3.5 mm on the back-side of the PCB and approximately 6 mm on the front-side. The PCB itself is approximately 1.5 mm thick.

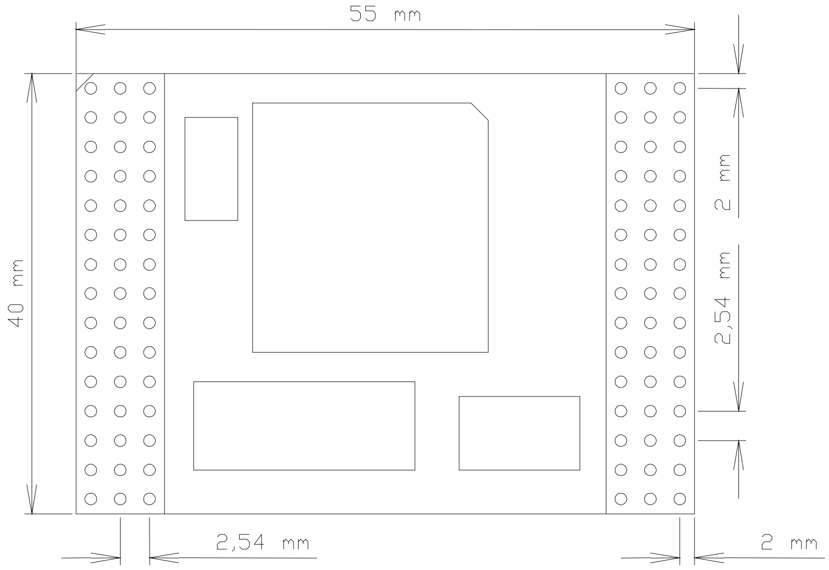


Figure 12: Physical Dimensions (not Shown at Scale)

Additional specifications:

Dimensions:	55 x 40 mm, ± 0.01 mm
Weight:	approximately 25 g
Storage temperature:	-40°C to +90°C
Operating temperature:	standard 0°C to + 70°C, extended -40°C to + 85°C
Humidity:	maximum 95 % r.F. not condensed
Operating voltage:	5 V ±5 %, VBAT 3 V ±20 %
Power consumption:	maximum 220 mA, typically 110 mA at 12 MHz oscillator frequency and 128 kByte RAM at +20°C
Power consumption with battery buffer:	maximum 100 µA, typically 1 µA for RAM supply and 1 µA for Real-Time Clock supply at +20°C
Delay Time /CS1- /CS3:	10 ns

These specifications describe the standard configuration of the phyCORE-P8xC591 as of the printing of this manual.

Please note that the module storage temperature is only 0°C to +70°C if a battery buffer is used for the RAM devices.

13 Hints for Handling the Module

If changing controllers, please ensure that appropriate PLCC extraction tools are used and that the socket and all components remain free from intrusive damage.

Removal of the standard quartz or oscillator 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.

14 The phyCORE-P8xC591 on the phyCORE Development Board LD 5V

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.

14.1 Concept of the phyCORE Development Board LD 5V

The phyCORE Development Board LD 5V provides a flexible development platform enabling quick and easy start-up and subsequent programming of the phyCORE-P8xC591 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, PHYTEC is 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.

The following figure illustrates the modular development platform concept:

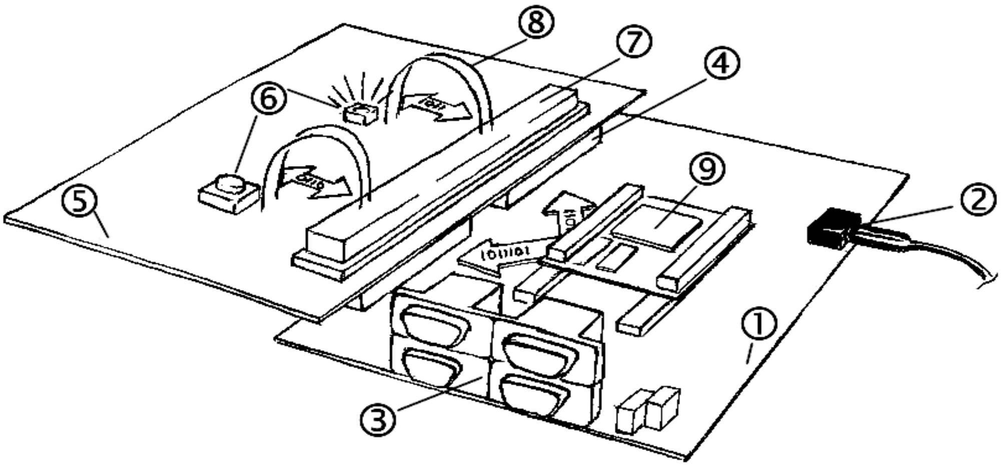


Figure 13: Modular Development and Expansion Board Concept with the phyCORE-P8xC591

The following sections contain specific information relevant to the operation of the phyCORE-P8xC591 mounted on the phyCORE Development Board LD 5V. For a general description of the Development Board, please refer to the corresponding Development Board Hardware Manual.

14.2 Development Board LD 5V Connectors and Jumpers

14.2.1 Connectors

As shown in *Figure 14*, the following connectors are available on the phyCORE Development Board LD 5V:

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
BAT1- receptacle for an optional battery

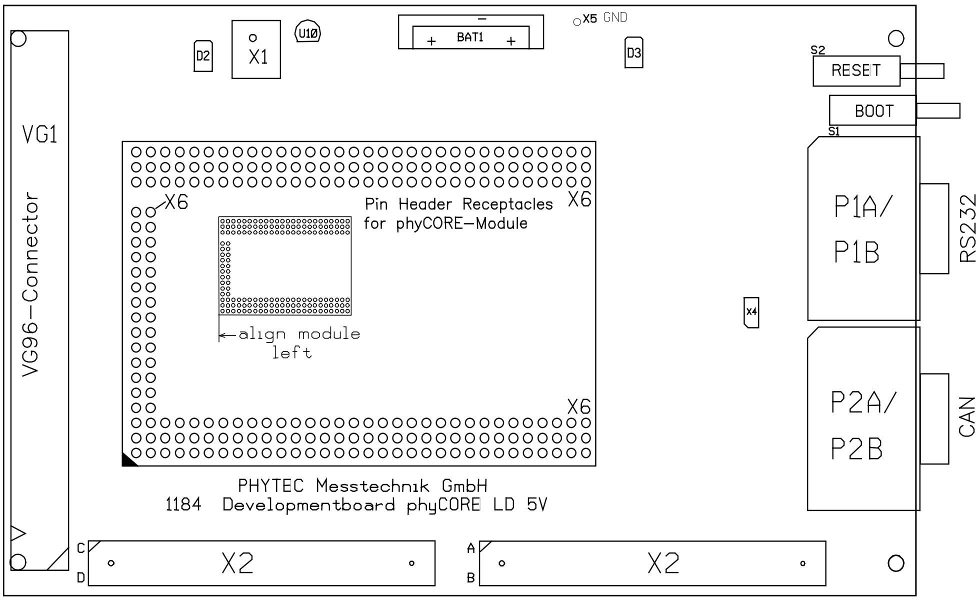


Figure 14: Location of Connectors on the phyCORE Development Board LD 5V

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.

14.2.2 Jumpers on the phyCORE Development Board LD 5V

Peripheral components of the phyCORE Development Board LD 5V can be connected to the signals of the phyCORE-P8xC591 by setting the applicable jumpers.

The Development Board's peripheral components are configured for use with the phyCORE-P8xC591 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-P8xC591 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.

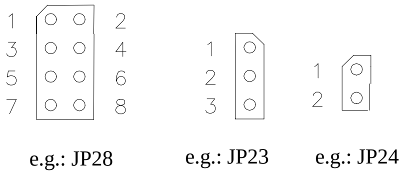


Figure 15: Numbering of Jumper Pads

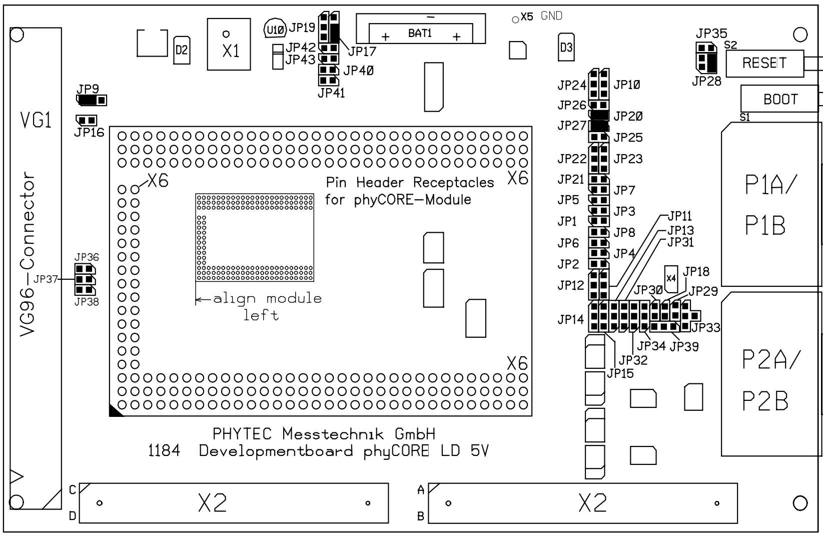


Figure 16: Location of the Jumpers (View of the Component Side)

Figure 17 shows the factory default jumper settings for operation of the phyCORE Development Board LD 5V with the standard phyCORE-P8xC591 (standard = P8xC591 controller, use of the RS-232 interface, the optional RS-485 interface, the first CAN interface, LED D3, the Boot button on the Development Board). Jumper settings for other functional configurations of the phyCORE-P8xC591 module mounted on the Development Board are described in *section 14.3*.

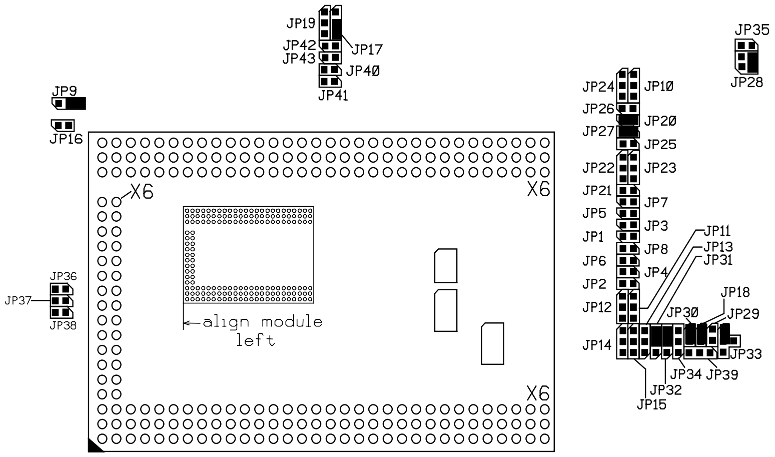


Figure 17: Default Jumper Settings of the phyCORE Development Board LD 5V with phyCORE-P8xC591

14.2.3 Unsupported Features and Improper Jumper Settings

The following table contains improper jumper settings for operation of the phyCORE-P8xC591 on a phyCORE Development Board LD 5V. Functions configured by these settings are not supported by the phyCORE module.

Supply Voltage:

The phyCORE Development Board LD 5V supports two main supply voltages for the start-up of various phyCORE modules. When using the phyCORE-P8xC591, only one main supply voltage is required, VCC1 with 5V. The connector pins for a second supply voltage on the phyCORE-P8xC591 are not defined.

Sockets G and H on the phyCORE Development Board LD 5V support connection of supply voltages for analog components. The phyCORE-P8xC591 is not populated with these sockets, and therefore Jumpers JP36 to JP38 must remain open.

Jumper	Setting	Description
JP16	closed	VCC2 routed to phyCORE-P8xC591
JP36	closed	AVDD routed to phyCORE-P8xC591
JP37	closed	REF+ routed to phyCORE-P8xC591
JP38	closed	REF- routed to phyCORE-P8xC591

Table 20: Improper Jumper Settings for the Development Board

14.3 Functional Components on the phyCORE Development Board LD 5V

This section describes the functional components of the phyCORE Development Board LD 5V supported by the phyCORE-P8xC591 and appropriate jumper settings to activate these components. Depending on the specific configuration of the phyCORE-P8xC591 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 LD 5V depending on user needs.

14.3.1 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 ±5 % regulated.

The required current load capacity of the power supply depends on the specific configuration of the phyCORE-P8xC591 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[1].

Jumper	Setting	Description
JP9	2 + 3	5 V main supply voltage to the phyCORE-P8xC591

Table 21: JP9 Configuration of the Main Supply Voltage VCCI

[1] : If the phyCORE-P8xC591 is purchased in a Rapid Development Kit, an appropriate 5V power adapter is included.

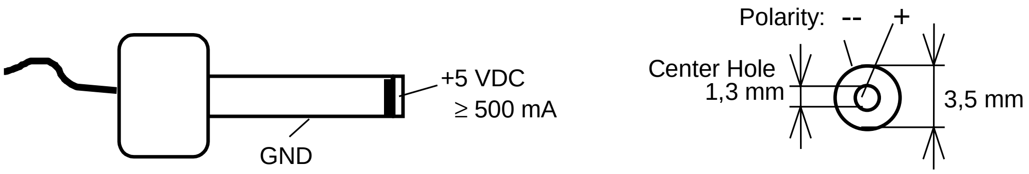


Figure 18: Connecting the Supply Voltage at X1

Caution:
When using the 5V supply, the following jumper settings are not allowed:

Jumper	Setting	Description
JP9	1 + 2	3.3 V as main supply voltage for the phyCORE-P8xC591
	open	phyCORE-P8xC591 not connected to main supply voltage

Table 22: JP9 Improper Jumper Settings for the Main Supply Voltage

Setting Jumper JP9 to position 1+2 configures a main power supply to the phyCORE-P8xC591 of 3.3 V which could destroy the module. If Jumper JP9 is open, no main power supply is connected to the phyCORE-P8xC591. This jumper setting should therefore never be used.

14.3.2 Starting FlashTools

The Flash memory of the phyCORE-P8xC591 contains the FlashTools firmware. The combination of this firmware and the corresponding software installed on the PC allows for on-board Flash programming with application programs via an RS-232 interface.

In order to start FlashTools on the phyCORE-P8xC591, the Boot pin (X1D6) of the phyCORE module must be connected to a high-level signal at the time the Reset signal changes from its active to the inactive state.

The phyCORE Development Board LD 5V provides three different options to enable the Flash programming mode:

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 FlashTools firmware if the Boot button is pressed during a hardware reset or power-on.

Jumper	Setting	Description
JP28	3 + 4	Boot button (in conjunction with Reset button or connection of the power supply) starts the FlashTools firmware on the phyCORE-P8xC591

Table 23: JP28 Configuration of the Boot Button

2. The Boot input of the phyCORE-P8xC591 can also be permanently connected to VCC. This spares pushing the Boot button during a hardware reset or power-on.

Jumper	Setting	Description
JP28	2 + 4	Boot input connected permanently with VCC. FlashTools are always started with Reset button or with connection of the power supply

Table 24: JP28 Configuration of a Permanent FlashTools Start Condition

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

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 (/RESIN) via pin 7 while a static high-level is applied to pin 4 for the Boot signal.

Jumper	Setting	Description
JP22	1 + 2	Pin 7 (CTS) of the DB-9 socket P1A as Reset signal for the phyCORE-P8xC591
JP23	1 + 2	Pin 4 (DSR) of the DB-9 socket P1A as Boot signal for the phyCORE-P8xC591
JP10	2 + 3	High-level Boot signal connected with the Boot input of the phyCORE-P8xC591

Table 25: 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-P8xC591

Table 26: Improper Jumper Settings for Boot via RS-232

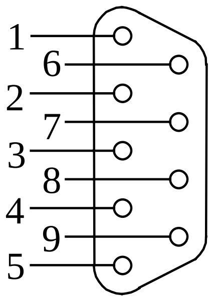
14.3.3 First Serial Interface at Socket P1A

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-P8xC591. When connected to a host-PC, the phyCORE-P8xC591 can be rendered in FlashTools mode via signals applied to the socket P1A (*refer to section 14.3.2*).

Jumper	Setting	Description
JP20	closed [1]	Pin 2 of DB-9 socket P1A connected with RS-232 interface signal TxD0 of the phyCORE-P8xC591
	open	Pin 2 of DB-9 socket P1A not connected
JP21	open	Pin 9 of DB-9 socket P1A not connected
JP22	open	Pin 7 of DB-9 socket P1A not connected
	1 + 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
	1 + 2	Boot input of the module can be controlled via DTR signal from a host-PC
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-P8xC591
	open	Pin 3 of DB-9 socket P1A not connected

[1] = required for communication with FlashTools

Table 27: Jumper Configuration for the RS-232 Interface



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

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

14.3.4 Power Supply to External Devices via Socket P1A

The phyCORE Development Board LD 5V 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-P8xC591 over the Internet or a direct dial connection.

On all PHYTEC Rapid Development Kits mounted with a phyCORE module featuring a Philips microcontroller, these optional components are already populated at U11 and U12 at time of delivery.

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

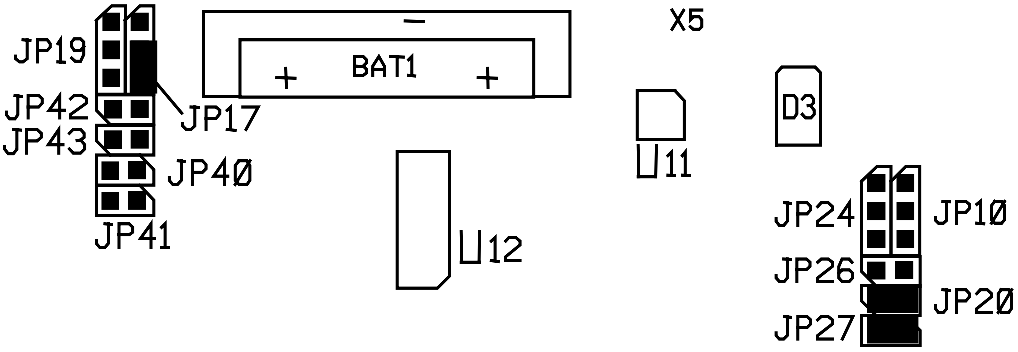


Figure 20: Location of Components at U11 and U12 for Power Supply to External Subassemblies

The components at U11 and U12 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 LD 5V.

- Overvoltage Protection:

  If the voltage at pin 6 exceeds the limiting value that can be provided by the phyCORE Development Board LD 5V, the voltage at pin 6 will be switched off immediately. This prevents damage to the phyCORE Development Board LD 5V 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 LD 5V 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 28: JP24 Power Supply to External Devices Connected to P1A on the Development Board

14.3.5 Socket P1B

Socket P1B is the upper socket of the double DB-9 connector at P1. The phyCORE-P8xC591 does not support a second RS-232 interface. Socket P1B remains unused.

Jumper	Setting	Description
JP1	open	Pin 2 of the DB-9 socket P1B not connected
JP2	open	Pin 9 of the DB-9 socket P1B not connected
JP3	open	Pin 7 of the DB-9 socket P1B not connected
JP4	open	Pin 4 of the DB-9 socket P1B not connected
JP5	open	Pin 6 of the DB-9 socket P1B not connected
JP6	open	Pin 8 of the DB-9 socket P1B not connected
JP7	open	Pin 1 of the DB-9 socket P1B not connected
JP8	open	Pin 3 of the DB-9 socket P1B not connected
JP40	open	Pin 2 of the DB-9 socket P1B not connected
JP41	open	Pin 3 of the DB-9 socket P1B not connected

Table 29: Jumper Configuration of the DB-9 Socket P1B

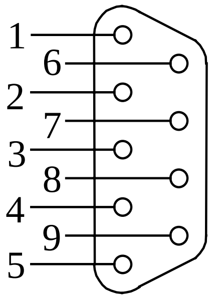


Figure 21: Pin Assignment of the DB-9 Socket P1B (Front View)

Caution:
When using the phyCORE-P8xC591 mounted on a phyCORE Development Board LD 5V the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP1	closed	Pin 2 of the DB-9 socket P1B is connected to B (RS-485) of the phyCORE-P8xC591
JP8	closed	Pin 3 of the DB-9 socket P1B is connected to A (RS-485) of the phyCORE-P8xC591
JP40	closed	Pin 2 of the DB-9 socket P1B is connected to pin VAGND of the phyCORE-P8xC591
JP41	closed	Pin 3 of the DB-9 socket P1B is connected to pin VAREF of the phyCORE-P8xC591

Table 30: Improper Jumper Settings for Configuration of P1B

If an RS-232 cable is connected to P1B by mistake, the voltage level on the RS-232 lines could destroy the phyCORE-P8xC591.

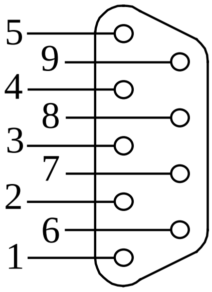
14.3.6 CAN Interface at Plug P2A

Plug P2A is the lower plug of the double DB-9 connector at P2. P2A is connected to the CAN interface (CAN0) of the phyCORE-P8xC591 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-P8xC591 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-P8xC591
JP32	2 + 3	Pin 7 of the DB-9 plug P2A is connected to CAN-H0 from on-board transceiver on the phyCORE-P8xC591
JP11	open	Input at optocoupler U4 on the phyCORE Development Board LD 5V open
JP12	open	Output at optocoupler U5 on the phyCORE Development Board LD 5V open
JP13	open	No supply voltage to CAN transceiver and optocoupler on the phyCORE Development Board LD 5V
JP18	open	No GND potential at CAN transceiver and optocoupler on the phyCORE Development Board LD 5V
JP29	open	No power supply via CAN bus
JP42	open	Input at optocoupler U4 on the phyCORE Development Board LD 5V open
JP43	open	Output at optocoupler U5 on the phyCORE Development Board LD 5V open

Table 31: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the phyCORE-P8xC591



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-P8xC591,Front View)

Caution:
When using the DB-9 plug P2A as CAN interface and the CAN transceiver on the phyCORE-P8xC591 the following jumper settings are not functional and could damage the module:

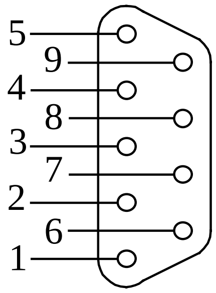
Jumper	Setting	Description
JP31	1 + 2	Pin 2 of DB-9 plug P2A connected with CAN-L0 from CAN transceiver on the Development Board
JP32	1 + 2	Pin 7 of DB-9 plug P2A connected with CAN-H0 from CAN transceiver on the Development Board
JP11	1 + 2	Input at optocoupler U4 on the Development Board connected with CAN-L0 from phyCORE-P8xC591
JP11	2 + 3	PWM0 from P8xC591 is connected to CAN transceiver U2 via optocoupler U4
JP12	1 + 2	Output at optocoupler U5 on the Development Board connected with CAN-H0 on phyCORE-P8xC591
JP12	2 + 3	PWM1 from P8xC591 is connected to CAN transceiver U2 via optocoupler U5
JP13	1 + 2	Supply voltage for CAN transceivers and optocouplers derived from external source (CAN bus) via on-board voltage regulator
JP13	2 + 3	Supply voltage for CAN transceivers and optocouplers derived from local supply circuitry on the Development Board
JP18	closed	CAN transceiver and optocoupler on the Development Board connected with local GND potential
JP29	closed	Supply voltage for on-board voltage regulator from pin 9 of DB-9 plug P2A or P2B
JP42	closed	Input on optocoupler U4 on the Development Board is connected with the CAN_Tx (P1.1) of the phyCORE-P8xC591
JP43	closed	Output at optocoupler U5 on the Development Board connected with CAN_Rx (P1.0) of the phyCORE-P8xC591

Table 32: Improper Jumper Settings for the CAN Plug P2A (CAN Transceiver on phyCORE-P8xC591)

2. The CAN transceiver populating the phyCORE-P8xC591 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	open	Input at optocoupler U4 on the phyCORE Development Board LD 5V open
JP12	open	Output on optocoupler U5 on the phyCORE Development Board LD 5V open
JP13	2 + 3	Supply voltage for CAN transceiver and optocoupler derived from local supply circuitry on the phyCORE Development Board LD 5V
JP18	closed	CAN transceiver and optocoupler on the Development Board connected with local GND potential
JP29	open	No power supply via CAN bus
JP42	closed	Input at optocoupler U4 on the Development Board connected to CAN_Tx (P1.1) of the phyCORE-P8xC591
JP43	closed	Output at optocoupler U5 on the Development Board connected to CAN_Rx (P1.0) of the phyCORE-P8xC591

Table 33: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the Development Board



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:

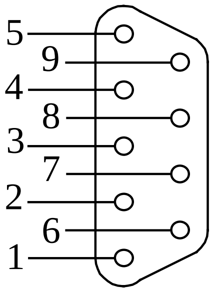
Jumper	Setting	Description
JP31	2 + 3	Pin 2 of DB-9 plug P2A connected with CAN-L0 from on-board transceiver on the phyCORE-P8xC591
JP32	2 + 3	Pin 7 of DB-9 plug P2A connected with CAN-H0 from on-board transceiver on the phyCORE-P8xC591
JP11	1 + 2	Input at optocoupler U4 on the Development Board connected with CAN-L0 from phyCORE-P8xC591
JP11	2 + 3	PWM0 from P8xC591 is connected with CAN transceiver U2 via optocoupler U4
JP12	1 + 2	Output at optocoupler U5 on the Development Board connected with CAN-H0 on phyCORE-P8xC591
JP12	2 + 3	PWM1 from P8xC591 is connected to CAN transceiver U2 via optocoupler U5
JP13	1 + 2	Supply voltage for CAN transceiver and optocoupler 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 or P2B

Table 34: Improper Jumper Settings for the CAN Plug P2A (CAN Transceiver on the Development Board)

3. The CAN transceiver populating the phyCORE-P8xC591 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 13 V, 14 to 20 V or 21 to 27 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	open	Input at optocoupler U4 on the phyCORE Development Board LD 5V open
JP12	open	Output at optocoupler U5 on the phyCORE Development Board LD 5V open
JP13	1 + 2	Supply voltage for CAN transceiver and optocoupler on the Development Board derived from external source (CAN bus) via on-board voltage regulator
JP18	open	CAN transceiver and optocoupler 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 (or P2B)
JP39	1 + 2	external CAN supply of 7 to 13 V
	2 + 3	external CAN supply of 14 to 20 V
	open	external CAN supply of 21 to 27 V
JP42	closed	Input at optocoupler U4 on the Development Board connected to CAN_Tx (P1.1) of the phyCORE-P8xC591
JP43	closed	Output at optocoupler U5 on the Development Board connected to CAN_Rx (P1.0) of the phyCORE-P8xC591

Table 35: Jumper Configuration for CAN Plug P2A using the CAN Transceiver on the Development Board with Galvanic Separation



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-P8xC591
JP32	2 + 3	Pin 7 of DB-9 plug P2A connected with CAN-H0 from on-board transceiver on the phyCORE-P8xC591
JP11	1 + 2	Input at optocoupler U4 on the Development Board connected with CAN-L0 from phyCORE-P8xC591
JP11	2 + 3	PWM0 from P8xC591 is connected to CAN transceiver U2 via optocoupler U4
JP12	1 + 2	Output at optocoupler U5 on the Development Board connected with CAN-H0 on phyCORE-P8xC591
JP12	2 + 3	PWM1 from P8xC591is connected to CAN transceiver U2 via optocoupler U5
JP13	2 + 3	Supply voltage for CAN transceiver and optocoupler derived from local supply circuitry on the phyCORE Development Board LD 5V
JP18	closed	CAN transceiver and optocoupler on the Development Board connected with local GND potential

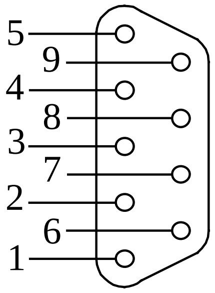
Table 36: Improper Jumper Settings for the CAN Plug P2A (CAN Transceiver on Development Board with Galvanic Separation)

14.3.7 RS-485 Interface at Plug P2B

Plug P2B is the upper plug of the double DB-9 connector at P2. P2B is connected to the RS-485 interface signals of the phyCORE-P8xC591 via jumpers. The RS-485 interface is an alternative function of the serial interface signals on the P8xC591 controller. The default configuration of the phyCORE-P8xC591 activates the RS-232 interface. In order to enable the RS-485 signals, different jumper settings on the phyCORE-P8xC591 are required (*refer to sections 3.3, 3.6 and 3.11 for details*).

Jumper	Setting	Description
JP33	1 + 2	Pin 2 of DB-9 plug P2B connected with RS-485 A signal on the phyCORE-P8xC591
JP34	open	Pin 7 of DB-9 plug P2B disconnected from signals on the Development Board
JP14	open	CAN optocoupler U6 on the Development Board disconnected from module pins
JP15	open	CAN optocoupler U7 on the Development Board disconnected from module pins
JP13	open	CAN transceiver and optocoupler on the Development Board disconnected from supply voltage
JP18	closed	Pin 3 and 6 of DB-9 plug P2B connected with local GND potential on the Development Board
JP29	open	Supply voltage via pin 9 of DB-9 plugs P2A or P2B disabled
JP30	closed	Pin 8 of DB-9 plug P2B connected with RS-485 B signal on the phyCORE-P8xC591

Table 37: Jumper Configuration for DB-9 Plug P2B as RS-485 Interface



Pin 8:	A Signal RS-485
Pin 3:	GND (Development Board Ground)
Pin 2:	B Signal RS-485
Pin 6:	GND (Development Board Ground)

Figure 25: Pin Assignment of the DB-9 Plug P2B as RS-485 Interface

Caution:
When using the DB-9 plug P2B as RS-485 interface the following jumper settings are not functional and could damage the module:

Jumper	Setting	Description
JP33	2 + 3	Pin 2 of DB-9 plug P2B connected with CAN-L1 signal from U3 on the Development Board
	2 + 4	Pin 2 of DB-9 plug P2B connected with PWM0 signal on the phyCORE-P8xC591
JP34	1 + 2	Pin 7 of DB-9 plug P2B connected with CAN-H1 signal from U3 on the Development Board
	2 + 3	Pin 7 of DB-9 plug P2B connected with PWM1 signal on the phyCORE-P8xC591
JP14	1 + 2	CAN optocoupler U6 connected with CAN-L0 of the phyCORE-P8xC591
	2 + 3	CAN optocoupler U6 connected with PWM0 on the phyCORE-P8xC591
JP15	1 + 2	CAN optocoupler U7 connected with CAN-H0 on the phyCORE-P8xC591
	2 + 3	CAN optocoupler U7 connected with PWM1 on the phyCORE-P8xC591
JP13	1 + 2	Supply voltage for CAN transceiver and optocoupler on the Development Board derived from external source (CAN bus) via on-board voltage regulator
JP13	2 + 3	Supply voltage for CAN transceiver and optocoupler derived from local supply circuitry on the phyCORE Development Board LD 5V
JP18	open	Pin 3 and 6 of DB-9 connector P2B disconnected from local GND potential on the Development Board
JP29	closed	Supply voltage for on-board voltage regulator from pin 9 of DB-9 connector P2A or P2B

Table 38: Improper Jumper Settings for the RS-485 Interface at Plug P2B

14.3.8 Programmable LED D3

The phyCORE Development Board LD 5V offers a programmable LED at D3 for user implementations. This LED can be connected to a port pin at GPIO0 (JP17 = 1+2) or the data bus via a latch U14 (JP17 = 2+3). When using the phyCORE-P8xC591, only the data bus can be used to control LED D3 because port pin P1.0 (GPIO0) is reserved for the on-board CAN interface.

Control and illumination of the LED can be enabled via user code toggling data bit D0 at address FFDA0h. A low-level at latch U14 causes the LED to illuminate, LED D3 remains off when writing a high-level to latch U14.

Jumper	Setting	Description
JP17	2 + 3	Data bit D0 from the P8xC591 controller controls LED D3 via latch U14 on the Development Board

Table 39: JP17 Configuration of the Programmable LED D3

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

Jumper	Setting	Description
JP17	1 + 2	LED D3 is connected with port pin P1.0 (GPIO0) on the P8xC591

Table 40: JP17 Improper Jumper Setting LED D3

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

As described in *section 14.1*, all signals from the phyCORE-P8xC591 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:

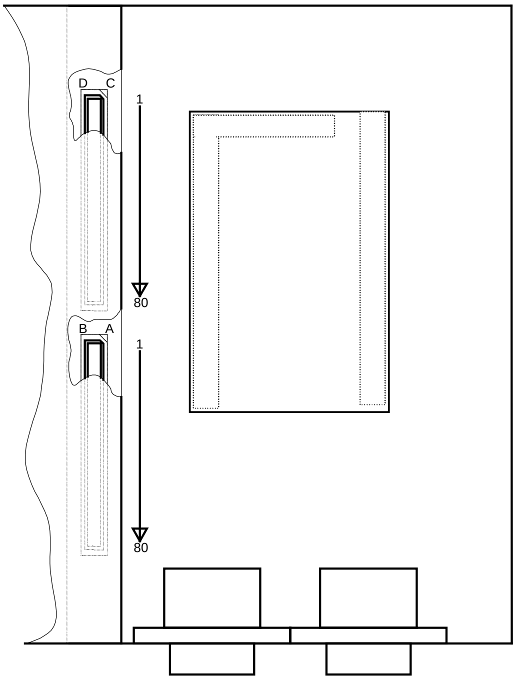


Figure 26: Pin Assignment Scheme of the Expansion Bus

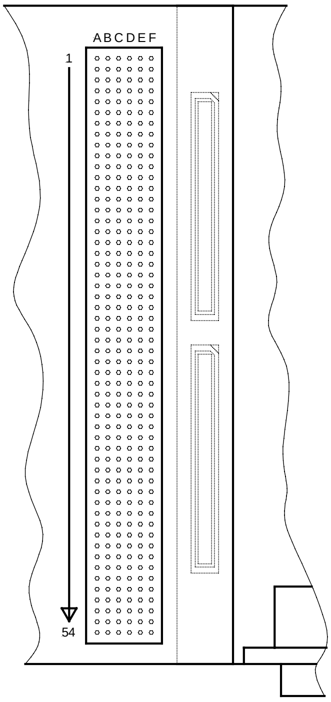


Figure 27: Pin Assignment Scheme of the Patch Field

The pin assignment on the phyCORE-P8xC591, in conjunction with the expansion bus (X2) on the Development Board and the patch field on an expansion board, is as follows:

Signal	**phyCORE-P8xC591**	**Expansion Bus**	**Patch Field**
P0.0/ AD0	12C	18B	33F
P0.1/ AD1	13A	19A	34A
P0.2/ AD2	13C	20A	34E
P0.3/ AD3	14A	20B	34B
P0.4/ AD4	14B	21A	34D
P0.5/ AD5	14C	21B	34F
P0.6/ AD6	15A	22B	35A
P0.7/ AD7	15C	23A	35E
A0	6A	8B	30B
A1	6B	9A	30D
A2	6C	10A	30F
A3	7A	10B	31A
A4	7C	11A	31E
A5	8A	11B	31B
A6	8C	12B	31F
A7	9A	13A	32A
P2.0/ A8	9B	13B	32C
P2.1/ A9	9C	14A	32E
P2.2/ A10	10A	15A	32B
P2.3/ A11	10C	15B	32F
P2.4/ A12	11A	16A	33A
P2.5/ A13	11B	16B	33C
P2.6/ A14	11C	17B	33E
P2.7/ A15	12A	18A	33B

Table 41: Pin Assignment Data/Address Bus for the phyCORE-P8xC591 / Development Board / Expansion Board

Signal	phyCORE-P8xC591	Expansion Bus	Patch Field
ClkIn	1A	1A	28A
P3.2 / INT0	1C	2B	28E
P3.3/ /INT1	2A	3A	28B
P3.4/ T0	2C	3B	28F
P3.5/ T1	3A	4A	29A
/CS1	3C	5A	29E
/CS2	4A	5B	29B
/CS3	4C	6B	29F
ALE	4B	6A	29D
/RD	5A	7B	30A
/WR	5C	8A	30E

Table 42: Pin Assignment Control Signals for the phyCORE-P8xC591 / Development Board / Expansion Board

Signal	phyCORE-P8xC591	Expansion Bus	Patch Field
/RESET	6E	10C	3D
/RESIN	6F	10D	3F
/WDI	4D	8D	3A
BOOT	6D	9C	3B
P1.0 / CANRxD	7D	11D	4A
P1.1 / CANTxD	8D	12D	4B
P1.2 / ADC0	8F	13C	4F
P1.3 / ADC0	9D	13D	5A
P1.4 / ADC0	9E	14C	5C
P1.5 / ADC0	9F	15C	5E
P1.6 / ADC0	10D	15D	5B
P1.7 / ADC0	10F	16C	5F
PWM0	12D	18D	6B
PWM1	13D	20C	7A
CANL	14D	21C	7B
CANH	15D	23C	8A
P3.0/ RxDTTL	11D	16D	6A
P3.1/ TxDTTL	11E	17D	6C
RxD0	15F	23D	8E
TxD0	14F	22D	7F
A	14E	21D	7D
B	13F	20D	7E

Table 43: Pin Assignment Interface Signals for the phyCORE-P8xC591 / Development Board / Expansion Board

Signal	phyCORE-P8xC591	Expansion Bus	Patch Field
PFI	4F	7D	2F
PFO	5F	8C	3E
VCC	1D, 2D	1C, 2C, 1D, 2D	1A, 1C
VPD	4E	6D	2D
VBAT	4D	6C	2B
VAGND	11F	18C	6E
VAREF	12F	19C	6F
GND	2B, 3B, 5B, 7B, 8B, 10B, 12B, 13B, 15B, 1F; 2F, 3F, 5E, 7E, 8E, 12E, 13E, 15E	2A, 7A, 12A, 17A, 22A, 27A, 32A, 37A,42A, 47A, 52A, 57A, 62A, 67A, 72A, 77A, 4B, 9B, 14B, 19B, 24B, 29B, 34B, 39B, 44B, 49B, 54B, 59B, 64B, 69B, 74B, 79B,3C, 7C, 12C, 17C, 22C, 27C, 32C, 37C, 42C, 47C, 52C, 57C, 62C, 67C, 72C, 77C, 3D, 9D, 14D, 19D, 24D, 29D, 34D, 42D, 47D, 52D, 57D, 62D, 67D, 72D, 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, 9D, 20D, 21D, 24D, 25D, 26D, 28D, 31D, 32D, 33D, 36D, 37D, 38D, 41D, 42D, 43D, 46D, 47D, 48D, 51D, 52D, 53D, 1E, 2E, 1F

Table 44: Pin Assignment Power Supply for the phyCORE-P8xC591 / Development Board / Expansion Board

Signal	phyCORE-P8xC591	Expansion Bus	Patch Field
NC	1B, 2D, 3D, 2E, 3E, 7F,	1B, 4C, 5C, 11C, 4D, 5D,	2A, 1B, 2C, 28C; 1D, 4E,
Pins on the Development Board not being used by the phyCORE-P8xC591	16A to 32A 16B to 32B 16C to 32C 16D to 32C 16E to 32E 16F to 32F except GND pins in row B and E	24A to 80A 23B to 80B 24C to 80C 24D to 80D except GND pins in row A, B, C and D	9A to 27A 35A to 54A 8B to 27B 35B to 54B 8C to 27C 35C to 54C 8D to 27D 35D to 54D 9E to 27E 35E to 54E 9F to 27F 35F to 54F except GND pins in row C and D

Table 45: Unused Pins on the phyCORE-P8xC591 / Development Board / Expansion Board

14.3.10 Battery Connector BAT1

The mounting space BAT1 (see PCB stencil) is provided for connection of a battery that buffers volatile memory devices (SRAM) and the RTC on the phyCORE-P8xC591. The Reset controller on the phyCORE-P8xC591 is responsible for switching from a normal power supply to a back-up battery. This optional battery required for this function (*refer to section 11*) is available through PHYTEC (order code BL-003).

14.3.11 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 LD 5V can be connected to a port pin at GPIO1 (JP19 1+2) or the data bus via latch U14 and driver U15 (JP19 = 2+3). When using the phyCORE-P8xC591, the Silicon Serial Number can only be controlled by the data bus because port pin P1.1 (GPIO1) is reserved for the on-board CAN interface. Access to the DS2401 Silicon Serial Number can be enabled via user code by means of data bit D1 at address FDA0h.

Jumper	Setting	Description
JP19	2 + 3	Data bit D1 from the P8xC591 controls Silicon Serial Number via latch U14 / driver U15

Table 46: JP19 JumperConfiguration for Silicon Serial Number Chip

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

Jumper	Setting	Description
JP19	1 + 2	DS2401 Silicon Serial Number connected with port pin P1.1 on the P8xC591 controller

Table 47: JP19 Improper Jumper Setting for Configuration of the Silicon Serial Number Chip

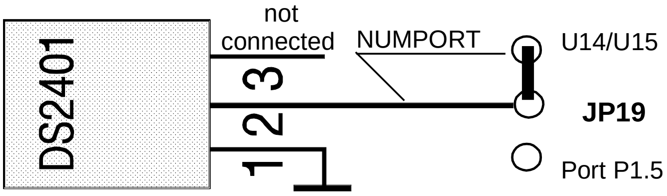


Figure 28: Connecting the DS2401 Silicon Serial Number

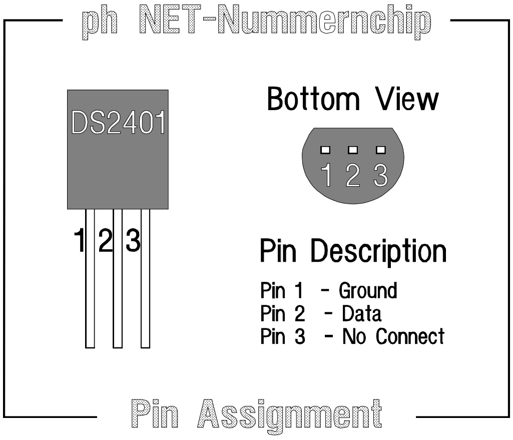


Figure 29: Pin Assignment of the DS2401 Silicon Serial Number

14.3.12 Pin Header Connector X4

The pin header X4 on the Development Board enables connection of an optional modem power supply. Connector X4 supplies 5V = at pin 1 and provides the phyCORE Development Board LD 5V 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.

Index

Document: **phyCORE-P8xC591**
Document number: **L-470e_5, June 2001**

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

Ordering No. L-470e_5
Printed in Germany