

Portux

Technical Reference

taskit GmbH

Seelenbinderstr. 33

D-12555 Berlin

Germany

Tel. +49 (30) 611295-0

Fax +49 (30) 611295-10

<http://www.taskit.de>

© *taskit* GmbH, Berlin

All rights reserved. This document and the products referred to herein are copyrighted works of *taskit* GmbH. Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form, without the express written permission of *taskit* GmbH. If however, your only means of access is electronic, permission to print one copy is hereby granted. Neither *taskit* GmbH nor *taskit*'s distributors assume any liability arising from the use of this manual/reference or any product described herein.

Table of contents

| | |
|---|-----------|
| 1. Introduction..... | 5 |
| 2. Technical properties..... | 6 |
| 2.1. Portux920T..... | 6 |
| 2.1.1. <i>CPU</i> | 6 |
| 2.1.2. <i>Memory</i> | 6 |
| 2.1.3. <i>Interfaces on-board</i> | 6 |
| 2.1.4. <i>Interfaces PXB (portux extension bus)</i> | 6 |
| 2.1.5. <i>Supply voltage</i> | 6 |
| 2.1.6. <i>Firmware</i> | 7 |
| 2.1.7. <i>Starterkit (eurocard version)</i> | 7 |
| 2.2. Portux Panel-PC..... | 8 |
| 2.2.1. <i>CPU</i> | 8 |
| 2.2.2. <i>Memory</i> | 8 |
| 2.2.3. <i>Interfaces</i> | 8 |
| 2.2.4. <i>Firmware</i> | 8 |
| 2.2.5. <i>Starterkit</i> | 8 |
| 3. The interface to the AT91RM9200..... | 9 |
| 3.1. <i>Description</i> | 9 |
| 3.2. <i>Physical address space</i> | 9 |
| 3.3. <i>The parallel input / output controller</i> | 11 |
| 3.4. <i>PIO-controller register addresses</i> | 13 |
| 3.5. <i>PIO-controller multiplexing</i> | 16 |
| 4. Hardware..... | 21 |
| 4.1. Portux920T..... | 21 |
| 4.1.1. <i>Mainboard layout Portux920T EU</i> | 21 |
| 4.1.2. <i>Mainboard layout Portux920T SW</i> | 24 |
| 4.2. Portux Panel-PC hardware..... | 27 |
| 4.2.1. <i>Connexion plug arrangement</i> | 27 |
| 4.2.2. <i>Top panel dimensions</i> | 29 |
| 4.2.3. <i>Mainboard layout</i> | 30 |
| 4.2.4. <i>Mainboard dimensions</i> | 32 |
| 4.2.5. <i>On-board connectors</i> | 32 |
| 5. Hardware extensions..... | 38 |
| 5.1. PXB description..... | 38 |

5.2. PXB pin assignment.....40

6. Appendix.....45

6.1. DBGU-adapter.....45

1. Introduction

The Portux is an industrial computer that is based on the Atmel® AT91RM9200 microcontroller. A 920T arm-core with 180 MHz is working inside it. The AT91RM9200 has a plurality of integrated devices like USB-host and client, SPI, I²C, Ethernet MAC and 4 USARTs.

The Portux is available in 2 different versions: The Portux920T is a half size euroboard single computer are the perfect building block for hardware developers. The Portux Panel-PC comes to you with a robust housing and a display.

The *Portux920T* is available in 2 different versions: the Portux920T EU (Europaboard version) can be integrated into 19-inch racks and the Portux920T SW (Sandwich version) allows sandwich assembly.

The *Portux Panel-PC* is also available in 2 different versions: one with a robust aluminium case and one without that case to integrate it in your own case.

Both versions of the Portux Panel-PC are delivered with a display and a matrix keyboard integrated in the front panel. The jacks for the ethernet, USB, PS/2, RS-232 and the slot for the MMC/SD card are soldered to the circuit board.

By default the Portux is delivered with an adapted Linux and the bootloader U-Boot. If the Portux should only start running the “Portux QuickStartGuide” can be read immediately.

If the bootloader, the contents of the rootfilesystem or the modules of the Linux kernel ought to be modified then the “Portux LinuxGuide” must be read.

The technical reference and the “AT91RM9200 Manual” are used as a reference book for developing own hardware extensions or to write own kernel modules.

To create your own applications it is used rarely.

2. Technical properties

2.1. Portux920T

2.1.1. CPU

- Atmel ® AT91RM9200
- ARM920T core
- 200 MIPS at 180 Mhz
- Memory Management Unit
- 16 kb data cache and 16 kb instructions cache

2.1.2. Memory

- 16 MiB flash (100,000 erase cycles typical per sector)
- 64 MiB SDRAM
- 16 KB internal SRAM
- 128 KB internal EEPROM

2.1.3. Interfaces on-board

- 2 synchronous / asynchronous serial interfaces (TTL compatible). The first serial port is also equipped with a Debug interface (UART with RX / TX).
- 10/100 Base-T ethernet MAC via 8P8C connector (Eurocard version) or two 4-pin rows (Sandwich version)
- JTAG
- MCI (Multimedia card interface, MMC and SD-card compatible)
- Portux Extension Bus via 96-pin connector (Eurocard version) or pin row (Sandwich version)

2.1.4. Interfaces PXB (portux extension bus)

- 2 synchronous / asynchronous serial interfaces (TTL compatible)
- USB 2.0 host
- USB 2.0 client
- SPI (Serial Peripheral Interface) with master / slave mode and 4 chip-selects
- TWI (Two Wire Interface) for Atmel ® EEPROM
- I²C bus (via TWI)
- PIF-bus (8 bit bus with 4 chip-selects)

2.1.5. Supply voltage

- Eurocard version: 6.5 – 24 V (cable clamp)
- Sandwich version: 3.3 V (on PXB)

2.1.6. Firmware

- U-Boot 1.1.2 (2004.12.23) or a newer version
- Linux kernel 2.6.xx or newer version
- GNU toolchain with GCC 3.4.2 and DGB for crosscompiling
- Nano-X Window System 0.9 with NXLIB v0.44
- JamVM 1.2 and GNU-classpath

2.1.7. Starterkit (eurocard version)

- Firmware CD
- Serial data cable
- Crosswired ethernet cable
- DBGU adapter
- Power Pack

2.2. Portux Panel-PC

2.2.1. CPU

- Atmel ® AT91RM9200
- ARM920T core
- 200 MIPS at 180 Mhz
- Memory Management Unit
- 16 kb data cache and 16 kb instructions cache

2.2.2. Memory

- 16 MiB flash (100,000 erase cycles typical per sector)
- 64 MiB SDRAM
- 16 KB internal SRAM
- 128 KB internal EEPROM

2.2.3. Interfaces

- 4 synchronous / asynchronous serial interfaces (TTL compatible)
- Debug interface (UART with RX / TX)
- USB 2.0 host
- USB 2.0 client
- SPI (Serial Peripheral Interface) with master / slave mode and 4 chip-selects
- TWI (Two Wire Interface) for Atmel ® EEPROM
- I²C bus (via TWI)
- 10/100 Base-T ethernet MAC
- JTAG
- MCI (Multimedia card interface, MMC and SD-card compatible)
- PIF-bus (8 bit bus with 4 chip-selects)

2.2.4. Firmware

- U-Boot 1.1.2 (2004.12.23) or a newer version
- Linux kernel 2.6.xx or newer version
- GNU toolchain with GCC 3.4.2 and DGB for crosscompiling
- Nano-X Window System 0.9 with NXLIB v0.44
- JamVM 1.2 and GNU-classpath

2.2.5. Starterkit

- Firmware CD
- Serial data cable
- Crosswired ethernet cable
- DBGU adapter
- Power Pack

3. The interface to the AT91RM9200

3.1. Description

The integrated devices of the AT91RM9200 can be programmed using registers. In most cases these registers are controlled by the operating system (Linux) and are not programmed directly on the Portux. Therefore only a general description of the addresses will be given here.

Excepting the parallel input / output controller (PIO), it will be accessed directly in many cases and is documented completely.

Of cause the Portux can be programmed directly without the use of the operating system. This is done by the boot loader U-Boot for example.

We refer the interested to the Atmel® data sheet for the AT91RM9200 here, which contains a complete documentation of all interfaces.

3.2. Physical address space

After the execution of the **remap** command the 4 GiB physical address space is separated as showed in the following table. Accessing these addresses directly is only possible if the MMU is deactivated. As soon as the MMU is activated the visible address space is changed completely.

If read / write functions onto memory addresses are to be accomplished direct in the application, the corresponding address space has to be mapped to the virtual address space using **mmap** or **ioremap** under Linux.

| Address | Size | Portux | Portux Device | Device |
|------------------------------|---------|------------------------------|---------------------------|----------------------|
| 0x00 000 000 0x0F FFF FFF | 256 MiB | 0x00 000 000 0x00 0FF FFF | Internal SRAM (mapped) | Internal memories |
| | | 0x00 100 000 0x00 1FF FFF | Internal ROM | |
| | | 0x00 200 000 0x00 2FF FFF | Internal SRAM | |
| | | 0x00 300 000 0x00 3FF FFF | USB host port (mapped) | |
| | | 0x00 400 000 0x0F FFF FFF | undefined | |
| 0x10 000 000 0x1F FFF FFF | 256 MiB | 0x10 000 000 0x10 FFF FFF | 16 MiB Flash | EBI Chip select 0 |
| 0x20 000 000 0x2F FFF FFF | 256 MiB | 0x20 000 000 0x23 FFF FFF | 64 MiB SDRAM | EBI Chip select 1 |
| 0x30 000 000 0x3F FFF FFF | 256 MiB | | | EBI Chip select 2 |

| Address | Size | Portux | Portux Device | Device |
|------------------------------|---------------------|--|---|----------------------|
| 0x40 000 000 0x4F FFF FFF | 256 MiB | | | EBI Chip select 3 |
| 0x50 000 000 0x5F FFF FFF | 256 MiB | | | EBI Chip select 4 |
| 0x60 000 000 0x6F FFF FFF | 256 MiB | | | EBI Chip select 5 |
| 0x70 000 000 0x7F FFF FFF | 256 MiB | | | EBI Chip select 6 |
| 0x80 000 000 0x8F FFF FFF | 256 MiB | | | EBI Chip select 7 |
| 0x90 000 000 0x9F FFF FFF | 256 MiB | | | undefined |
| 0xA0 000 000 0xAF FFF FFF | 256 MiB | | | undefined |
| 0xB0 000 000 0xBF FFF FFF | 256 MiB | | | undefined |
| 0xC0 000 000 0xCF FFF FFF | 256 MiB | 0xC0 000 000 0xC0 41F FFF | 4,125 MiB Dataflash | |
| 0xD0 000 000 0xDF FFF FFF | 256 MiB | | | undefined |
| 0xE0 000 000 0xEF FFF FFF | 256 MiB | | | undefined |
| 0xF0 000 000 0xFF FFE FFF | 255 MiB 1020 KiB | 0xFF FA0 000 0xFF FA0 040 0xFF FA0 080 0xFF FA4 000 0xFF FA4 040 0xFF FA4 080 0xFF FB0 000 0xFF FB4 000 0xFF FB8 000 0xFF FBC 000 0xFF FC0 000 0xFF FC4 000 0xFF FC8 000 | Timer counter 0 Timer counter 1 Timer counter 2 Timer counter 3 Timer counter 4 Timer counter 5 USB device port Multimedia card interface Two Wire Interface (I ² C) Ethernet MAC USART0 USART1 USART2 | User peripheral |

| Address | Size | Portux | Portux Device | Device |
|------------------------------|-------|--------------|------------------|-----------------------|
| | | 0xFF FCC 000 | USART3 | |
| | | 0xFF FE0 000 | SPI | |
| 0xFF FFF 000 0xFF FFF FFF | 4 KiB | 0xFF FFF 400 | PIO controller A | System Peripherals |
| | | 0xFF FFF 600 | PIO controller B | |
| | | 0xFF FFF 800 | PIO controller C | |

Table 1: Physical Address Space

3.3. The parallel input / output controller

The AT91RM9200 comes with 3 PIO controllers, each may manage up to 32 programmable I/O ports. Each I/O port is associated with a bit number in the 32 bit register of the user interface. Each I/O port may be configured for general purpose I/O or assigned to a function of an integrated peripheral device. In doing so multiplexing with 2 integrated devices is possible. That means a pin may be used as GPIO, device A or device B.

Survey of the PIO controller properties:

- interrupt on level changing for each I/O port individually configurable
- glitch filter; pulse that are lower than a half clock cycle are ignored
- multi drive ability
- programmable pull up resistor at each I/O port
- pin data status register; the level of the I/O port may be read out at any time
- synchronous output; this enables the possibility to clear and set several I/O ports with one write cycle

Pull-up resistor

Each I/O port has an integrated pull-up resistor. After a reset the pull-ups are activated at each pin. The value of the resistor lies between 100 and 200 k Ω . The pull-up resistor can be activated or deactivated by setting the PIO_PUER (pull-up enable register) or PIO_PUDR (pull-up disable register). Writing these registers results in setting or deleting the corresponding bits in the PIO_PUSR (pull-up status register). Reading a 1 in the the PIO_PUSR means that the pull-up is deactivated, reading a 0 means the the pull-up is activated.

I/O port or integrated peripheral

If a pin is multiplexed with the functions of one or two integrated devices the selection is controlled via the PIO_PER (PIO enable register) or the PIO_PDR (PIO disable register). The result of a write operation into these registers is the PIO_PSR (PIO status register). A value of 0 indicates that the corresponding pin is controlled by an integrated device, a value of 1 indicates that the corresponding pin is controlled by the PIO controller.

Selecting peripheral A or B

The PIO controller provides multiplexing for up to two peripheral functions. The selection is done by writing the register PIO_ASR (A select register) or rather PIO_BSR (B select register). The selection can be read out from the PIO_ABSR (AB select status register), where 0 means peripheral A is selected and 1 means peripheral B is selected. The selection only affects the output, the inputs of the peripheral are always connected to the corresponding pins. Setting the registers PIO_ASR and PIO_BSR has only an effect if the I/O is deactivated thus PIO_PDR is set for the corresponding pin.

Output control

If an I/O port is controlled by the PIO controller (setting of PIO_PER) then one pin can be used as Output. This is achieved by writing the PIO_OER (output enable register) and PIO_ODR (output disable register). The result of this operation can be read out from the PIO_OSR (output status register). If the corresponding bit is 0 the I/O port is only used as input, if it is 1 the I/O port is driven by the PIO controller.

The level of an output can be set by writing the register PIO_SODR (set output data register) and PIO_CODR (clear output data register). These writing operations set or clear the corresponding bit in the PIO_ODSR (output data status register). Writing into PIO_OER and PIO_ODR or rather PIO_SODR and PIO_CODR will set the corresponding bit in PIO_OSR or rather PIO_ODSR independently of whether the corresponding pin is driven by the PIO controller or is assigned to an internal device. This allows to configure the I/O port before it is driven by the PIO controller. Thus the first level of a pin can be assigned.

Synchronous output

In some applications it can be desirable to operate several I/O port in one write / delete cycle. This can be achieved by writing the registers PIO_OWER (output write enable register) and PIO_OWDR (output write disable register). This results in setting / deleting the corresponding bit in the PIO_OWSR (output write status register). PIO_OWSR is used by the PIO Controller as a PIO_ODSR write authorization mask. The level of the pins that are configured in that way can now be set directly by writing the PIO_ODSR and not over the indirection with PIO_SODR or PIO_CODR. Therewith a synchronous output is ensured.

After reset the synchronous output is disabled, that means PIO_OWSR resets at 0x0.

Multi drive control (open drain)

Each I/O port can be independently programmed in Open Drain mode. This permits several drivers to be connected to one I/O line which can be driven low by each driver. An external pull-up resistor (or enabling of the internal one) is generally required to guarantee a high level on the line. An I/O pin can be set into Open Drain Mode by writing the registers PIO_MDER (multi driver enable register) and PIO_MDDR (multi driver disable register).

PIO_MDSR (Multi-driver Status Register) indicates the pins that are configured to work in open drain mode.

Inputs

The level on each I/O line can be read through PIO_PDSR (Peripheral Data Status Register). This register indicates the level of the I/O lines regardless of their configuration. Reading the I/O line levels requires the clock of the PIO controller to be enabled, otherwise PIO_PDSR reads the levels present on the I/O line at the time the clock was disabled.

Input glitch filter

Optional input glitch filters are independently programmable on each I/O line. When the glitch filter is enabled, a glitch with a duration of less than 1/2 master clock (MCK) cycle is automatically rejected, while a pulse with a duration of 1 master clock cycle or more is accepted. For pulse durations between 1/2 master clock cycle and 1 master clock cycle the pulse may or may not be taken into account, depending on the precise timing of its occurrence. Thus for a pulse to be visible it must exceed 1 master clock cycle. The glitch filters are controlled by the register set; PIO_IFER (Input Filter Enable Register), PIO_IFDR (Input Filter Disable Register) and PIO_IFSR (Input Filter Status Register). Writing PIO_IFER and PIO_IFDR respectively sets and clears bits in PIO_IFSR. When the glitch filter is enabled, it does not modify the behaviour of the inputs on the peripherals. It acts only on the value read in PIO_PDSR and on the input change interrupt detection.

The glitch filters require that the PIO Controller clock is enabled.

Input change interrupt

The PIO Controller can be programmed to generate an interrupt when it detects an input change on an I/O line.

As Input change detection is possible only by comparing two successive samplings of the input of the I/O line, the PIO Controller clock must be enabled.

The Input Change Interrupt is controlled by writing PIO_IER (Interrupt Enable Register) and PIO_IDR (Interrupt Disable Register), which respectively enable and disable the input change interrupt by setting and clearing the corresponding bit in PIO_IMR (Interrupt Mask Register).

The input change interrupt is available, regardless of the configuration of the I/O line.

When an input change is detected on an I/O line, the corresponding bit in PIO_ISR (Interrupt Status Register) is set. If the corresponding bit in PIO_IMR is set, the PIO Controller interrupt line is asserted.

If an input change occurs on several I/O lines at the same time a single interrupt signal is generated.

When the software reads PIO_ISR, all the interrupts are automatically cleared. This signifies that all the interrupts that are pending when PIO_ISR is read must be handled.

3.4. PIO-controller register addresses

Each I/O line controlled by the PIO Controller is associated with a bit in each of the PIO Controller User Interface registers. Each register is 32 bits wide.

Bit 0 of a register in the PIO controller A represents the I/O pin PA0, bit 8 represents PA8

and so on.

Thus it is possible to program every single I/O port individually for each feature.

The address of a register is a result from the offset of the below mentioned table in addition to the base address of the corresponding PIO controller (0xFF FFF 400 PIO controller A, 0xFF FFF 600 PIO controller B and 0xFF FFF 800 PIO controller C).

| Offset | Name | Register | Access | Reset Value | Value: 0 | Value: 1 |
|--------|----------|-------------------------------|--------|-------------|-----------------------|-----------------------|
| 0x0000 | PIO_PER | PIO Enable Register | WO | - | No Effect | Enable PIO |
| 0x0004 | PIO_PDR | PIO Disable Register | WO | - | No Effect | Disable PIO |
| 0x0008 | PIO_PSR | PIO Status Register | RO | 0x00000000 | PIO is inactive | PIO is active |
| 0x000C | | Reserved | | | | |
| 0x0010 | PIO_OER | Output Enable Register | WO | - | No Effect | Output Enabled |
| 0x0014 | PIO_ODR | Output Disable Register | WO | - | No Effect | Output Disabled |
| 0x0018 | PIO_OSR | Output Status Register | RO | 0x00000000 | Only Input | Output Enabled |
| 0x001C | | Reserved | | | | |
| 0x0020 | PIO_IFER | Input Filter Enable Register | WO | - | No Effect | Input Filter Enabled |
| 0x0024 | PIO_IFDR | Input Filter Disable Register | WO | - | No Effect | Input Filter Disabled |
| 0x0028 | PIO_IFSR | Input Filter Status Register | RO | 0x00000000 | Input Filter Disabled | Input Filter Enabled |
| 0x002C | | Reserved | | | | |
| 0x0030 | PIO_SODR | Set Output Data Register | WO | - | No Effect | High Level |
| 0x0034 | PIO_CODR | Clear Output Data Register | WO | - | No Effect | Low Level |
| 0x0038 | PIO_ODSR | Output Data Status Register | WR | 0x00000000 | Low Level | High Level |
| 0x003C | PIO_PDSR | Pin Data Status Register | RO | | Low Level | High Level |
| 0x0040 | PIO_IER | Interrupt Enable Register | WO | - | No Effect | Interrupt Enable |

| Offset | Name | Register | Access | Reset Value | Value: 0 | Value: 1 |
|-----------------------|----------|-------------------------------|--------|-------------|--------------------------|---------------------------|
| 0x0044 | PIO_IDR | Interrupt Disable Register | WO | - | No Effect | Interrupt Disable |
| 0x0048 | PIO_IMR | Interrupt Mask Register | RO | 0x00000000 | Interrupt Disabled | Interrupt Enabled |
| 0x004C | PIO_ISR | Interrupt Status Register | RO | 0x00000000 | No Input Change Detected | Input Change Detected |
| 0x0050 | PIO_MDER | Multi Driver Enable Register | WO | - | No Effect | Multi Driver Enable |
| 0x0054 | PIO_MDDR | Multi Driver Disable Register | WO | | No Effect | Multi Driver Disable |
| 0x0058 | PIO_MDSR | Multi Driver Status Register | RO | 0x00000000 | Multi Driver Disabled | Multi Driver Enabled |
| 0x005C | | Reserved | | | | |
| 0x0060 | PIO_PUDR | Pull-up Disable Register | WO | - | No Effect | Pull-up Disable |
| 0x0064 | PIO_PUER | Pull-up Enable Register | WO | - | No Effect | Pull-up Enable |
| 0x0068 | PIO_PUSR | Pull-up Status Register | RO | 0x00000000 | Pull-up Enabled | Pull-up Disabled |
| 0x006C | | Reserved | | | | |
| 0x0070 | PIO_ASR | Peripheral A Select Register | WO | - | No Effect | Select Peripheral A |
| 0x0074 | PIO_BSR | Peripheral B Select Register | WO | - | No Effect | Select Peripheral B |
| 0x0078 | PIO_ABSR | AB Status Register | RO | 0x00000000 | Peripheral A Selected | Peripheral B Selected |
| 0x007C - 0x009C | | Reserved | | | | |
| 0x00A0 | PIO_OWER | Output Write Enable Register | WO | - | No Effect | Enables Writing PIO_ODSR |
| 0x00A4 | PIO_OWDR | Output Write Disable Register | WO | - | No Effect | Disables Writing PIO_ODSR |
| 0x00A8 | PIO_OWSR | Output Write | RO | 0x00000000 | Writing | Writing |

| Offset | Name | Register | Access | Reset Value | Value: 0 | Value: 1 |
|--------|------|-----------------|--------|-------------|-------------------|------------------|
| | | Status Register | | | PIO_ODSR Disabled | PIO_ODSR Enabled |
| 0x00AC | | Reserved | | | | |

Table 2: PIO controller register addresses

3.5. PIO-controller multiplexing

When programming the PIO controller you have to consider if a I/O pin is assigned to a function of an internal or external device. Only I/O pins that do not perform functions or whose functions are not used or that are not driven by the operating system can be controlled by the user manually without side effects.

Here it is again expressly pointed out that the input of a pin is always connected with the input of an internal device, independently of the programming of the PIO controller. That is, a possibly running driver reacts to the activities of the pin and thus with unexpected behaviour also can produce system crashes. Below mentioned table shows the multiplexing of the I/O pins on the three PIO controllers A, B and C.

PIO Controller A Multiplexing

| PIO Controller A | | | | | | | Portux | | |
|------------------|----------|-------------------|--------|--------------|-----------|-------------|----------|---------------|--------|
| CPU line | I/O Line | Peripheral A | | Peripheral B | | Reset State | Function | IC/X | PXB in |
| | | Function | Device | Function | Device | | | | |
| 42 | PA0 | MISO | SPI | PCK3 | PIO Clock | I/O | MISO | X1-A IC7-A | A19 |
| 43 | PA1 | MOSI | SPI | PCK0 | PIO Clock | I/O | MOSI | X1-C IC7-A | C19 |
| 44 | PA2 | SPCK | SPI | IRQ4 | - | I/O | SPCK | X1-A IC7-A | A20 |
| 45 | PA3 | NPCS0 | SPI | IRQ5 | - | I/O | NPCS0 | IC7-A | - |
| 46 | PA4 | NPCS1 | SPI | PCK1 | PIO Clock | I/O | PXB | X1-C | C20 |
| 47 | PA5 | NPCS2 | SPI | TXD3 | USART3 | I/O | PXB | X1-A | A30 |
| 48 | PA6 | NPCS3 | SPI | RXD3 | USART3 | I/O | PXB | X1-C | C29 |
| 49 | PA7 | ETXCK / EREFCK | ETH | PCK2 | PIO Clock | I/O | EREFCK | IC18-A | - |
| 50 | PA8 | ETXEN | ETH | MCCDB | MCI | I/O | ETXEN | IC18-A | - |
| 51 | PA9 | ETX0 | ETH | MCDB0 | MCI | I/O | ETX0 | IC18-A | - |
| 52 | PA10 | ETX1 | ETH | MCDB1 | MCI | I/O | ETX1 | IC18-A | - |
| 53 | PA11 | ECRS / ECRS DV | ETH | MCDB2 | MCI | I/O | ECRS DV | IC18-A | - |

| PIO Controller A | | | | | | | Portux | | |
|------------------|----------|--------------|------------------------|--------------|-----------|-------------|----------|-----------------------|--------|
| CPU line | I/O Line | Peripheral A | | Peripheral B | | Reset State | Function | IC/X | PXB in |
| | | Function | Device | Function | Device | | | | |
| 54 | PA12 | ERX0 | ETH | MCDB3 | MCI | I/O | ERX0 | IC18-A | - |
| 55 | PA13 | ERX1 | ETH | TCLK0 | TC0 | I/O | ERX1 | IC18-A | - |
| 58 | PA14 | ERXER | ETH | TCLK1 | TC1 | I/O | ERXER | IC18-A | - |
| 59 | PA15 | ENDC | ETH | TCLK2 | TC2 | I/O | EMDC | IC18-A | - |
| 60 | PA16 | EMDIO | ETH | IRQ6 | - | I/O | EMDIO | IC18-A | - |
| 61 | PA17 | TXD0 | USART0 | TIOA0 | TC0 | I/O | TXD0 | X6 | - |
| 64 | PA18 | RXD0 | USART0 | TIOB0 | TC0 | I/O | RXD0 | X6 | - |
| 65 | PA19 | SCK0 | USART0 | TIOA1 | TC1 | I/O | RI0 | X6 | - |
| 66 | PA20 | CTS0 | USART0 | TIOB1 | TC1 | I/O | CTS0 | X6 | - |
| 67 | PA21 | RTS0 | USART0 | TIOA2 | TC2 | I/O | RTS0 | X6 | - |
| 68 | PA22 | RXD2 | USART2 | TIOB2 | TC2 | I/O | PXB | X1-C | C24 |
| 69 | PA23 | TXD2 | USART2 | IRQ3 | - | I/O | PXB | X1-A | A25 |
| 70 | PA24 | SCK2 | USART2 | PCK1 | PIO Clock | I/O | PXB | X1-C | C23 |
| 71 | PA25 | TWD | TWI / I ² C | IRQ2 | - | I/O | TWD | X1-C IC8 IC21-A | A17 |
| 72 | PA26 | TWCK | TWI / I ² C | IRQ1 | - | I/O | TWCK | X1-C IC8 IC21-A | C17 |
| 73 | PA27 | MCCK | MCI | TCLK3 | TC3 | I/O | MCCK | X4 | - |
| 74 | PA28 | MCCDA | MCI | TCLK4 | TC4 | I/O | MCCDA | X4 | - |
| 77 | PA29 | MCDA0 | MCI | TCLK5 | TC5 | I/O | MCDA0 | X4 | - |
| 78 | PA30 | DRXD | DBGU | CTS2 | USART2 | I/O | DRXD | X6 | - |
| 79 | PA31 | DTXD | DBGU | RTS2 | USART2 | I/O | DTRD | X6 | - |

Table 3: PIO controller A multiplexing

PIO Controller B Multiplexing

| PIO Controller B | | | | | | | Portux | | |
|------------------|----------|--------------|-----------|--------------|--------|-------------|----------|--------|--------|
| CPU line | I/O Line | Peripheral A | | Peripheral B | | Reset State | Function | IC/X | PXB in |
| | | Function | Device | Function | Device | | | | |
| 80 | PB0 | TF0 | SSC0 | RTS3 | USART3 | I/O | PXB | X1-A | A31 |
| 81 | PB1 | TK0 | SSC0 | CTS3 | USART3 | I/O | PXB | X1-C | C31 |
| 82 | PB2 | TD0 | SSC0 | SCK3 | USART3 | I/O | PXB | X1-C | C25 |
| 83 | PB3 | RD0 | SSC0 | MCDA1 | MCI | I/O | MCI | X4 | - |
| 84 | PB4 | RK0 | SSC0 | MCDA2 | MCI | I/O | MCI | X4 | - |
| 85 | PB5 | RF0 | SSC0 | MCDA3 | MCI | I/O | MCI | X4 | - |
| 86 | PB6 | TF1 | SSC1 | TIOA3 | TC3 | I/O | PXB | X1-C | C28 |
| 87 | PB7 | TK1 | SSC1 | TIOB3 | TC3 | I/O | PXB | X1-C | C30 |
| 88 | PB8 | TD1 | SSC1 | TIOA4 | TC4 | I/O | PXB | X1-A | A24 |
| 89 | PB9 | RD1 | SSC1 | TIOB4 | TC4 | I/O | PXB | X1-A | A27 |
| 90 | PB10 | RK1 | SSC1 | TIOA5 | TC5 | I/O | PXB | X1-A | A29 |
| 91 | PB11 | RF1 | SSC1 | TIOB5 | TC5 | I/O | PXB | X1-A | A32 |
| 92 | PB12 | TF2 | SSC2 | ETX2 | ETH | I/O | I/O | SW1 | - |
| 95 | PB13 | TK2 | SSC2 | ETX3 | ETH | I/O | I/O | SW1 | - |
| 96 | PB14 | TD2 | SSC2 | ETXER | ETH | I/O | I/O | SW1 | - |
| 97 | PB15 | RD2 | SSC2 | ERX2 | ETH | I/O | I/O | SW1 | - |
| 98 | PB16 | RK2 | SSC2 | ERX3 | ETH | I/O | I/O | IC10 | - |
| 99 | PB17 | RF2 | SSC2 | ERXDV | ETH | I/O | PXB | X1-C | C22 |
| 100 | PB18 | RI1 | USART1 | ECOL | ETH | I/O | RI1 | X3 | - |
| 101 | PB19 | DTR1 | USART1 | ERXCK | ETH | I/O | DTR1 | X3 | - |
| 102 | PB20 | TXD1 | USART1 | - | - | I/O | TXD1 | X3 | - |
| 103 | PB21 | RXD1 | USART1 | - | - | I/O | RXD1 | X3 | - |
| 104 | PB22 | SCK1 | USART1 | - | - | I/O | DCD0 | X6 | - |
| 118 | PB23 | DCD1 | USART1 | - | - | I/O | DCD1 | X3 | - |
| 119 | PB24 | CTS1 | USART1 | - | - | I/O | CTS1 | X3 | - |
| 120 | PB25 | DSR1 | USART1 | EF100 | ETH | I/O | DSR1 | X3 | - |
| 121 | PB26 | RTS1 | USART1 | - | - | I/O | RTS1 | X3 | - |
| 122 | PB27 | PCK1 | PIO Clock | - | - | I/O | PXB | X1-A | A21 |
| 123 | PB28 | FI1 | - | - | - | I/O | PXB | X1-A | A13 |
| 124 | PB29 | IRQ1 | - | - | - | I/O | IRQ0 | IC21-A | - |

Table 4: PIO controller B multiplexing

PIO Controller C Multiplexing

| PIO Controller C | | | | | | | Portux | | |
|------------------|----------|---------------|-------------|--------------|--------|-------------|----------|-----------|--------|
| CPU line | I/O Line | Peripheral A | | Peripheral B | | Reset State | Function | IC/X | PXB in |
| | | Function | Device | Function | Device | | | | |
| 17 | PC0 | BFCCK | Burst Flash | - | - | I/O | I/O | X1-C LED1 | C14 |
| 18 | PC1 | BFRDY / SMOE | Burst Flash | - | - | I/O | I/O | X1-C LED2 | C15 |
| 21 | PC2 | BFAVD | Burst Flash | - | - | I/O | IRQ | IC18-A | - |
| 22 | PC3 | BFBA A / SMWE | Burst Flash | - | - | I/O | -SDRS232 | X3 X6 | - |
| 23 | PC4 | BFOE | Burst Flash | - | - | I/O | SDEETHER | IC18-A | - |
| 24 | PC5 | BFWE | Burst Flash | - | - | I/O | PXB | X1-A | A22 |
| 25 | PC6 | NWAIT | - | - | - | I/O | PXB | X1-A | A10 |
| 161 | PC7 | A23 | - | - | - | A23 | A23 | IC14-A | - |
| 162 | PC8 | A24 | - | - | - | A24 | A24 | IC14-A | - |
| 163 | PC9 | A25 / CFRNW | - | - | - | A25 | A25 | IC14-A | - |
| 11 | PC10 | NCS4 / CFCS | - | - | - | NCS4 | PXB | X1-C | C11 |
| 12 | PC11 | NCS5 / CFCE1 | - | - | - | NCS5 | PXB | X1-A | A12 |
| 13 | PC12 | NCS6 / CFCE2 | - | - | - | NCS6 | PXB | X1-C | C12 |
| 14 | PC13 | NCS7 | - | - | - | NCS7 | PXB | X1-C | C21 |
| 15 | PC14 | - | - | - | - | I/O | PXB | X1-A | A26 |
| 16 | PC15 | - | - | - | - | I/O | PXB | X1-C | C26 |
| 201 | PC16 | D16 | - | - | - | I/O | D16 | IC5-A | - |
| 202 | PC17 | D17 | - | - | - | I/O | D17 | IC5-A | - |
| 203 | PC18 | D18 | - | - | - | I/O | D18 | IC5-A | - |
| 204 | PC19 | D19 | - | - | - | I/O | D19 | IC5-A | - |
| 205 | PC20 | D20 | - | - | - | I/O | D20 | IC5-A | - |
| 026 | PC21 | D21 | - | - | - | I/O | D21 | IC5-A | - |
| 207 | PC22 | D22 | - | - | - | I/O | D22 | IC5-A | - |
| 028 | PC23 | D23 | - | - | - | I/O | D23 | IC5-A | - |

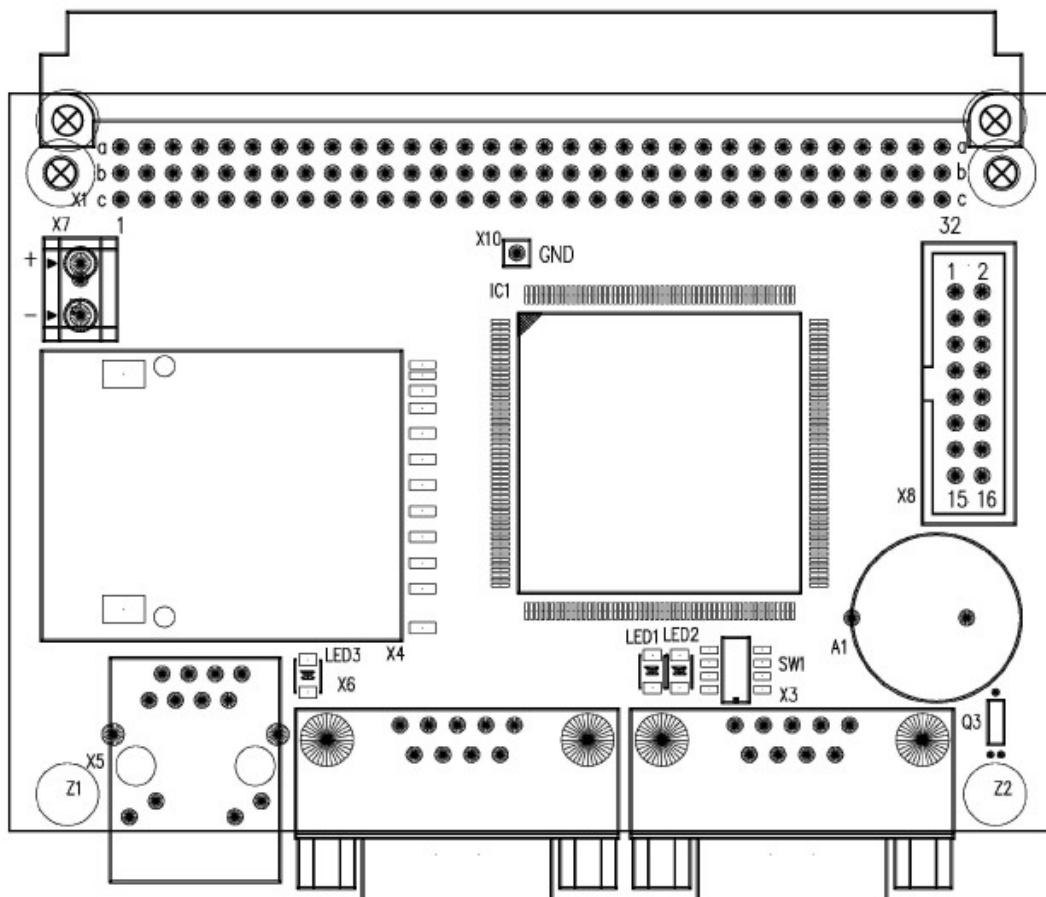
| PIO Controller C | | | | | | | Portux | | |
|------------------|----------|--------------|--------|--------------|--------|-------------|----------|-------|--------|
| CPU line | I/O Line | Peripheral A | | Peripheral B | | Reset State | Function | IC/X | PXB in |
| | | Function | Device | Function | Device | | | | |
| 1 | PC24 | D24 | - | - | - | I/O | D24 | IC5-A | - |
| 2 | PC25 | D25 | - | - | - | I/O | D25 | IC5-A | - |
| 3 | PC26 | D26 | - | - | - | I/O | D26 | IC5-A | - |
| 4 | PC27 | D27 | - | - | - | I/O | D27 | IC5-A | - |
| 5 | PC28 | D28 | - | - | - | I/O | D28 | IC5-A | - |
| 6 | PC29 | D29 | - | - | - | I/O | D29 | IC5-A | - |
| 9 | PC30 | D30 | - | - | - | I/O | D30 | IC5-A | - |
| 10 | PC31 | D31 | - | - | - | I/O | D31 | IC5-A | - |

Table 5: PIO controller C multiplexing

4. Hardware

4.1. Portux920T

4.1.1. Mainboard layout Portux920T EU



Portux920T EU

A1 Battery slot

X1 PXB – Portux Extension Bus

See chapter 5: “PXB pin assignment”

X3 RS232-2 D-Sub jack

| Pin | Assignment |
|-----|------------|
| 1 | -DCD |
| 2 | RXD |

| Pin | Assignment |
|------------|-------------------|
| 3 | TXD |
| 4 | -DTR |
| 5 | GND |
| 6 | -DSR |
| 7 | -RTS |
| 8 | -CTS |
| 9 | -RI |
| () | CGND |

X4 MMC/SD Card slot

X5 Ethernet 8P/8C (RJ-45) jack

X6 RS232-1 D-Sub jack

The first serial port (USART0) of the AT91RM9200 as well as the debug unit (DBGU) are connected to this D-Sub jack.

The debug unit is a simple UART that is only operating the RXD and TXD lines. These signals are connected to the DTR and DSR lines of the USART0, that is not operating these lines. These signals are called DTXD and DRXD.

A cabling description to create an adapter for common RS232 null modem cables can be found in the appendix.

| Pin | Assignment |
|------------|-------------------|
| 1 | -DCD |
| 2 | RXD |
| 3 | TXD |
| 4 | DTXD |
| 5 | GND |
| 6 | DRXD |
| 7 | -RTS |
| 8 | -CTS |
| 9 | -RI |
| () | CGND |

X7 Vin

Press down the orange cable holder, then insert the cable and release the orange cable holder.

X8 JTAG

The signal NRST (pin 15) is connected by the resistor J2. J2 is populated with a 0 ohm resistor.

| Pin | Assignment | Pin | Assignment |
|------------|-------------------|------------|-------------------|
| 1 | VCC | 2 | VCC |
| 3 | NTRST | 4 | GND |
| 5 | TDI | 6 | GND |
| 7 | TMS | 8 | GND |
| 9 | TCK | 10 | GND |
| 11 | TCK | 12 | GND |
| 13 | TDO | 14 | GND |
| 15 | NRST | 16 | GND |

X20 Bootmode

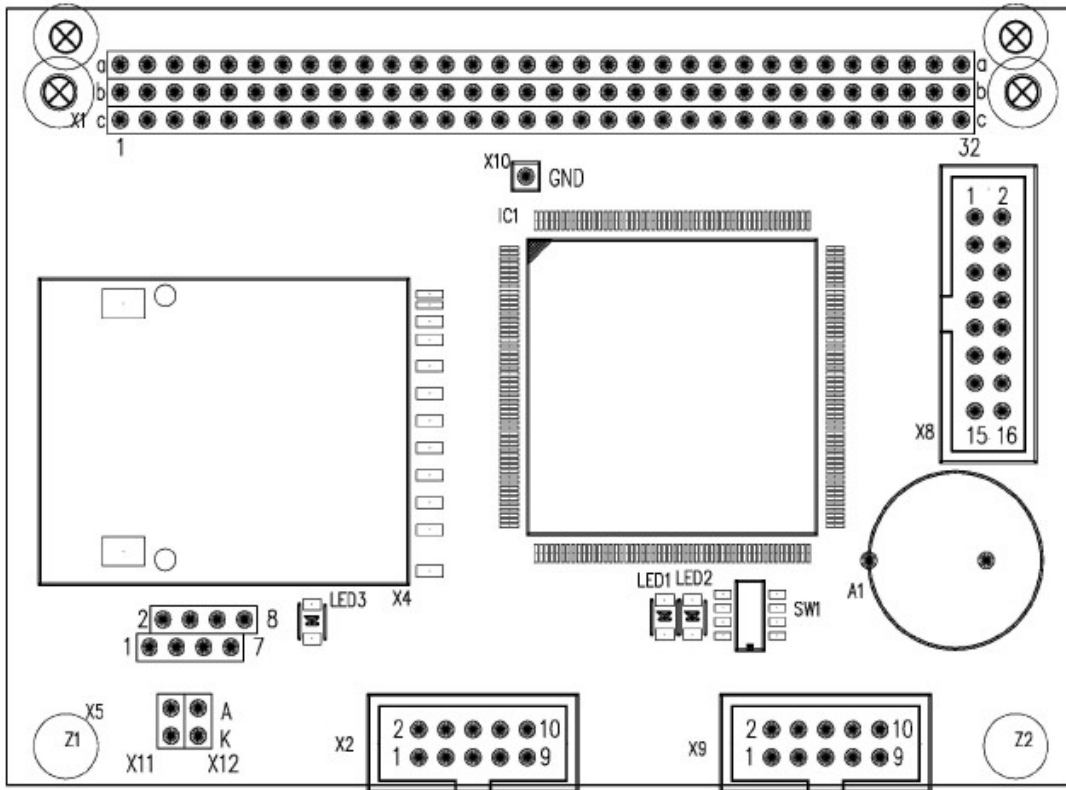
Atmel ® boot mode selector for debug unit on X6 (see Atmel ® Manual)

SW1

4 row DIP switch. Position ON pulls the corresponding I / O line to LOW level.
Before the DIP switch is uses it necessary to activate the integrated pull-up resistors (see chapter 3.3).

| Switch | I/O line |
|---------------|-----------------|
| 1 | PB12 |
| 2 | PB13 |
| 3 | PB14 |
| 4 | PB15 |

4.1.2. Mainboard layout Portux920T SW



A1 Battery slot

X1 PXB

See chapter 5: "PXB pin assignment"

X2 RS232-1 connector

The first serial port (USART0) of the AT91RM9200 as well as the debug unit (DBGU) are connected to this connector.

The debug unit is a simple UART that is only operating the RXD and TXD lines. These signals are connected to the DTR and DSR lines of the USART0, that is not operating these lines. These signals are called DTXD and DRXD.

A cabling description to create an adapter for common RS232 null modem cables can be found in the appendix.

| Pin | Assignment | Pin | Assignment |
|-----|------------|-----|------------|
| 1 | -DCD | 2 | DRXD |
| 3 | RXD | 4 | -RTS |
| 5 | TXD | 6 | -CTS |
| 7 | DTXD | 8 | -RI |

| Pin | Assignment | Pin | Assignment |
|------------|-------------------|------------|-------------------|
| 9 | GND | 10 | VCC |

X4 MMC

Slot for MMC/SD cards.

X5 Ethernet (two 4-pin rows)

| Pin | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------------|-----|-----|----|------|------|----|-----|----|
| Assignment | TD+ | TD- | CT | CGND | CGND | CT | RD+ | RD |

X8 JTAG

The signal NRST (pin 15) is connected by the resistor J2. J2 is populated with a 0 ohm resistor.

| Pin | Assignment | Pin | Assignment |
|------------|-------------------|------------|-------------------|
| 1 | VCC | 2 | VCC |
| 3 | NTRST | 4 | GND |
| 5 | TDI | 6 | GND |
| 7 | TMS | 8 | GND |
| 9 | TCK | 10 | GND |
| 11 | TCK | 12 | GND |
| 13 | TDO | 14 | GND |
| 15 | NRST | 16 | GND |

X9 RS232-2 connector

| Pin | Assignment | Pin | Assignment |
|------------|-------------------|------------|-------------------|
| 1 | -DCD | 2 | DSR |
| 3 | RXD | 4 | -RTS |
| 5 | TXD | 6 | -CTS |
| 7 | DTR | 8 | -RI |
| 9 | GND | 10 | VCC |

X11 Ethernet LED green

X12 Ethernet LED yellow

X20 Bootmode

Atmel ® boot mode selector for debug unit on X2 (see Atmel ® Manual)

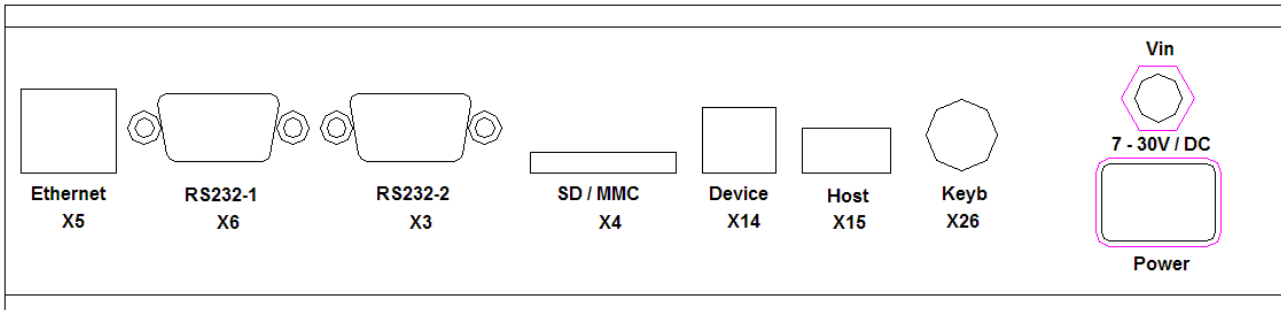
SW1

4 row DIP switch. Position ON pulls the corresponding I / O line to LOW level.
Before the DIP switch is uses it necessary to activate the integrated pull-up resistors (see chapter 3.3).

| Switch | I/O line |
|---------------|-----------------|
| 1 | PB12 |
| 2 | PB13 |
| 3 | PB14 |
| 4 | PB15 |

4.2. Portux Panel-PC hardware

4.2.1. Connexion plug arrangement



X5 Ethernet jack

The pin assignment is only valid for the 8pin onboard strip.

| Pin | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | () |
|------------|-----|-----|----|------|------|----|-----|----|------|
| Assignment | TD+ | TD- | CT | CGND | CGND | CT | RD+ | RD | CGND |

X6 RS232-1 D-Sub jack

The first serial port (USART0) of the AT91RM9200 as well as the debug unit (DBGU) are connected to this D-Sub jack.

The debug unit is a simple UART that is only operating the RXD and TXD lines. These signals are connected to the DTR and DSR lines of the USART0, that is not operating these lines. These signals are called DTXD and DRXD.

A cabling description to create an adapter for common RS232 null modem cables can be found in the appendix.

| Pin | Assignment |
|-----|------------|
| 1 | -DCD |
| 2 | RXD |
| 3 | TXD |
| 4 | DTXD |
| 5 | GND |
| 6 | DRXD |
| 7 | -RTS |
| 8 | -CTS |
| 9 | -RI |
| () | CGND |

X3 RS232-2 D-Sub jack

| Pin | Assignment |
|------------|-------------------|
| 1 | -DCD |
| 2 | RXD |
| 3 | TXD |
| 4 | -DTR |
| 5 | GND |
| 6 | -DSR |
| 7 | -RTS |
| 8 | -CTS |
| 9 | -RI |
| () | CGND |

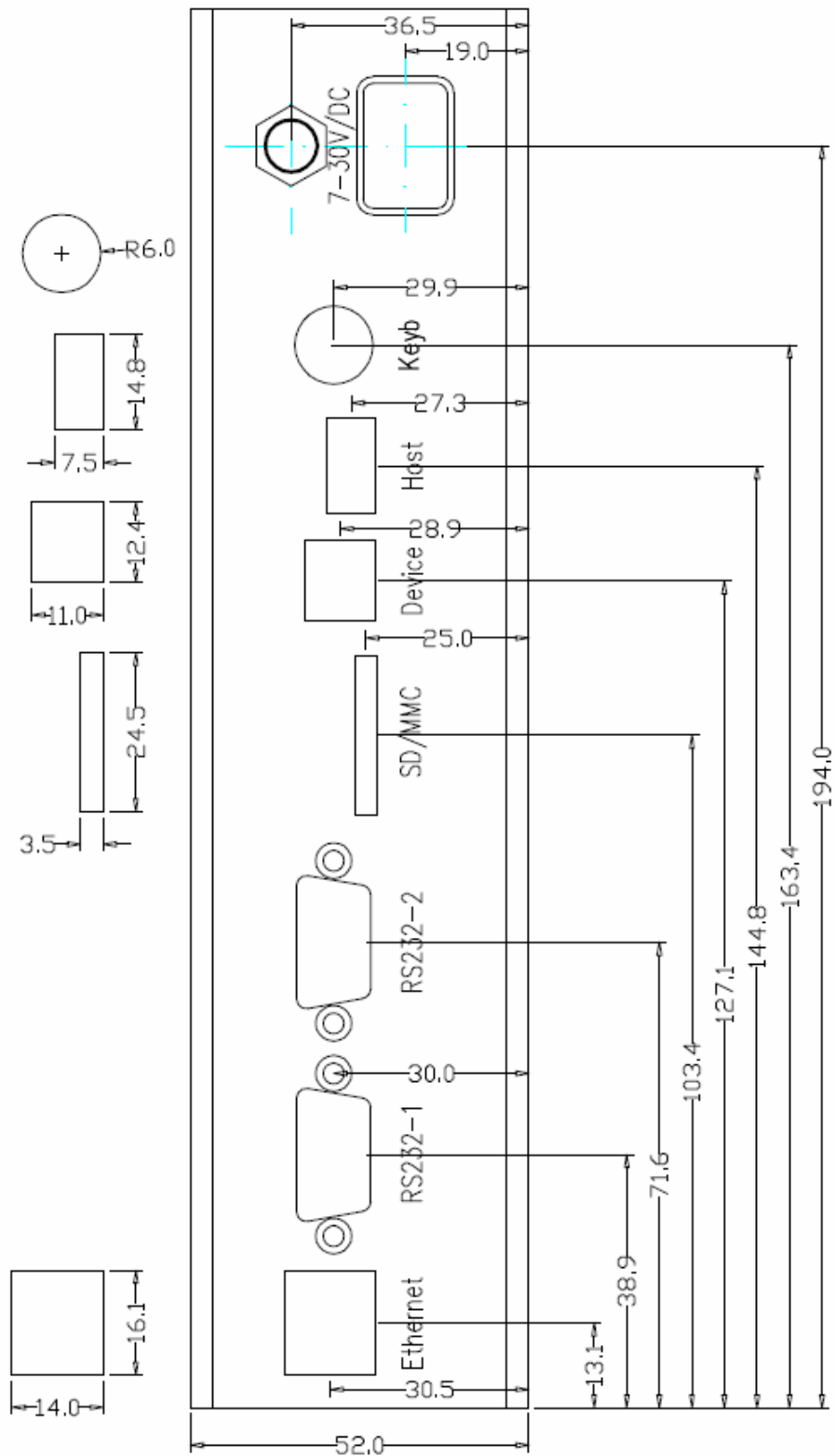
*X4 SD / MMC card slot**X14 USB-device jack**X15 USB-host jack**X26 PS/2 jack**Power*

This is the power button to switch the Portux Panel-PC on or off.

Vin jack

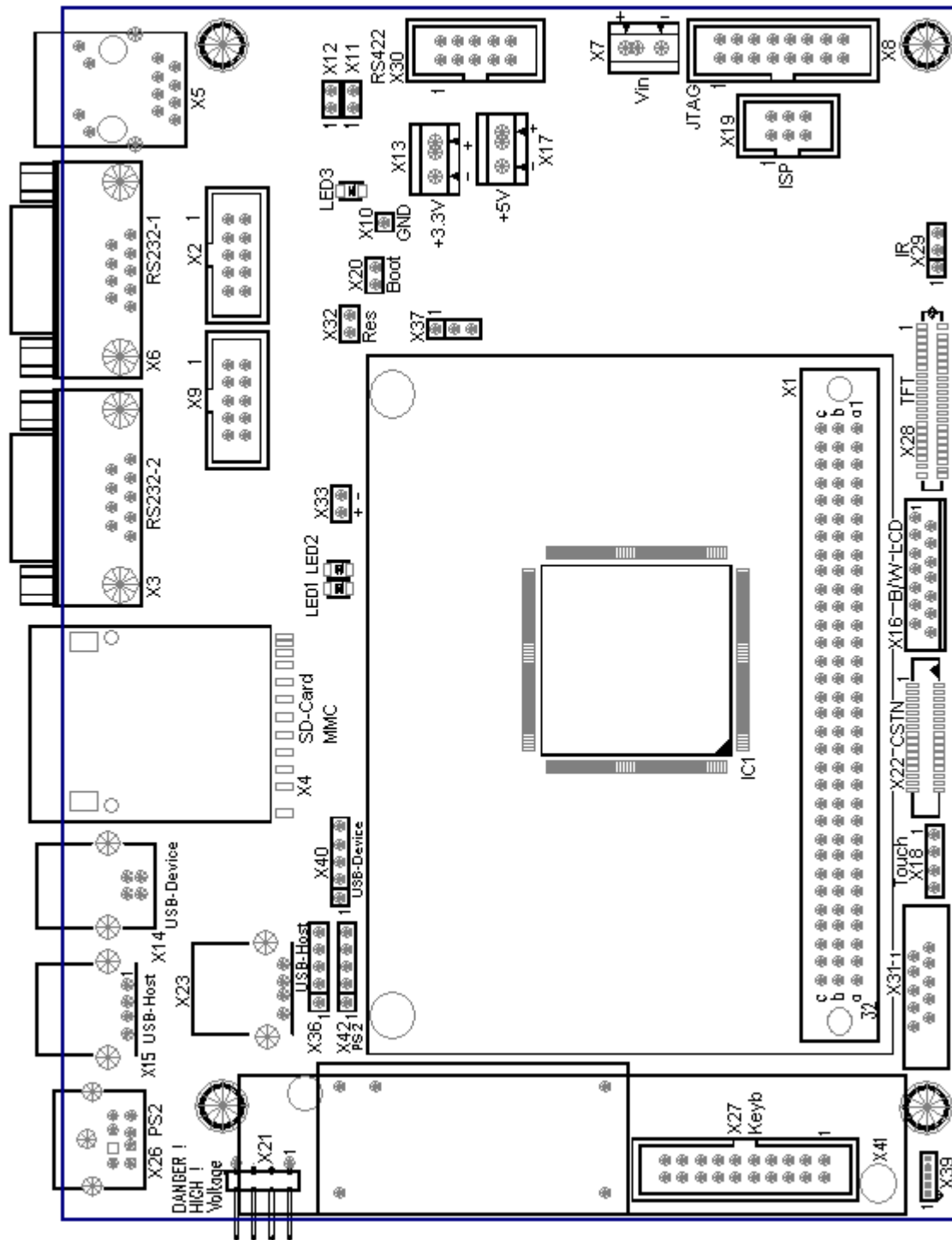
Connect a taskit or any other power supply with 7 to 30 Volt direct current to this jack. Where the positive pole is the pin in the middle and the negative pole is the circle around that pin.

4.2.2. Top panel dimensions

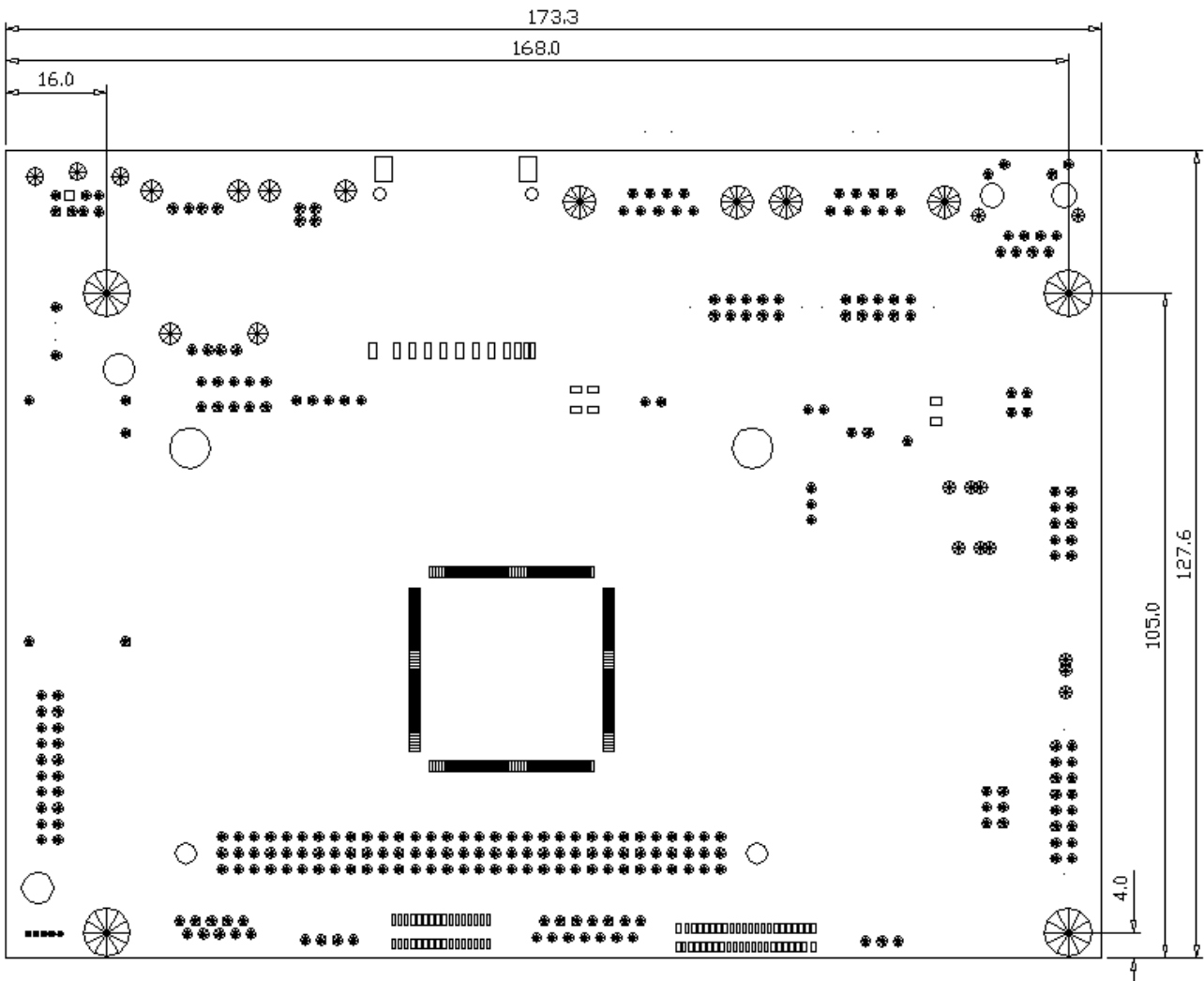


4.2.3. Mainboard layout

| | |
|-----|--|
| A1 | RTC battery slot |
| X1 | Portux Extension Bus |
| X2 | RS232-1 / DBGU strip |
| X3 | RS232-2 jack |
| X4 | MMC/SD card slot |
| X5 | Ethernet jack |
| X6 | RS232-1 / DBGU jack |
| X7 | Power supply (Vin) |
| X8 | JTAG strip |
| X9 | RS232-2 strip |
| X10 | Ground pin |
| X11 | Connector for Ethernet ACT LED |
| X12 | Connector for Ethernet Receive LED |
| X13 | 3.3V output |
| X14 | USB-device jack |
| X15 | USB-host jack |
| X16 | B/W LCD strip |
| X17 | 5V output |
| X18 | Touch screen connector |
| X20 | Jumper for Atmel bootmode selection |
| X21 | Backlight power supply for LCD/CSTN displays |
| X22 | CSTN strip |
| X26 | PS/2 jack |
| X27 | I/O lines / Matrix keyboard connector |
| X28 | TFT strip |
| X29 | IR sensor connection |
| X30 | RS422/485 strip |
| X32 | Reset jumper |
| X33 | Buzzer connector |
| X36 | USB host strip |
| X37 | One Wire Bus strip |
| X39 | TFT Power Supply connector |
| X40 | USB device strip |
| X42 | PS/2 strip |
| SW1 | 4 bit user switch (PIO controller B) |



4.2.4. Mainboard dimensions



4.2.5. On-board connectors

A1 Battery

Fastener for an 3V lithium battery to provide power for the internal real time clock.

X1 PXB

The Portux can be equipped with a 96-pin strip to connect hardware extension to the Portux Extension Bus (PXB). Pin assignment see chapter 5.2.

X2 RS232-1

The first serial port (USART0) of the AT91RM9200 as well as the debug unit (DBGU) are connected to this D-Sub jack.

The debug unit is a simple UART that is only operating the RXD and TXD lines. These

signals are connected to the DTR and DSR lines of the USART0, that is not operating these lines. These signals are called DTXD and DRXD.

A cabling description to create an adapter for common RS232 null modem cables can be found in the appendix.

| Pin | Assignment | Pin | Assignment |
|-----|------------|-----|------------|
| 1 | -DCD | 2 | DRXD |
| 3 | RXD | 4 | -RTS |
| 5 | TXD | 6 | -CTS |
| 7 | DTXD | 8 | -RI |
| 9 | GND | 10 | VCC |

X7 Vin - fastener for power supply

Connect to a regulated power supply (7V – 30V DC) via a 2-pole cable.

Usage: Push down the orange cable carrier and insert the corresponding cable and release the orange cable holder.

X8 JTAG

The signal NRST (pin 15) is connected by the resistor J2. J2 is populated with a 0 ohm resistor.

| Pin | Assignment | Pin | Assignment |
|-----|------------|-----|------------|
| 1 | VCC | 2 | VCC |
| 2 | NTRST | 4 | VCC |
| 3 | TDI | 6 | VCC |
| 4 | TMS | 8 | VCC |
| 5 | TCK | 10 | VCC |
| 6 | TCK | 12 | VCC |
| 7 | TDO | 14 | VCC |
| 8 | NRST | 16 | VCC |

X9 RS232-2

| Pin | Assignment | Pin | Assignment |
|-----|------------|-----|------------|
| 1 | -DCD | 2 | -DSR |
| 3 | RXD | 4 | -RTS |
| 5 | TXD | 6 | -CTS |
| 7 | -DTR | 8 | -RI |

| Pin | Assignment | Pin | Assignment |
|-----|------------|-----|------------|
| 9 | GND | 10 | VCC |

X10 GND

Common ground of the circuit board.

X11 Ethernet LED

Strip to connect a led to display the ethernet link (act) state (green led on the RJ-45 jack).

X12 Ethernet LED

Strip to connect a led to display the ethernet FDX (col) state (yellow led on the RJ-45 jack).

X13 3.3V

3.3 Volt power supply for peripheral devices.

X16 B/W LCD

Connector socket to connect a B/W LC display.

X17 5V

5 Volt power supply for peripheral devices.

X18 Touch

Strip to connect a touchscreen.

X20 Boot

Jumper for Atmel boot mode selection.

X21 LCD/CSTN HV supply

Power supply for B/W LCD and CSTN displays.

X22 CSTN

Connector socket to connect a CSTN display.

X27 Keyb

This strip is connected to the Atmel ATmega32 controller. When the matrix keyboard is connected to this strip (pin 9 – 20) there are 6 user programmable I/O lines available (pin 3

– 8). If there is no matrix keyboard connected then there are 8 user programmable I/ O lines (pin 3-10) available.

| Pin | Assignment | Pin | Assignment |
|-----|--------------------|-----|--------------------|
| 1 | +3V3 | 2 | NC |
| 3 | MAT_H (I/O) | 4 | MAT_G (I/O) |
| 5 | MAT_F (I/O) | 6 | MAT_E (I/O) |
| 7 | MAT_D (I/O) | 8 | MAT_C (I/O) |
| 9 | MAT_B (I/O, Keyb) | 10 | MAT_A (I/O, Keyb) |
| 11 | MAT_1 (Keyb) | 12 | MAT_2 (Keyb) |
| 13 | MAT_3 (Keyb) | 14 | MAT_4 (Keyb) |
| 15 | MAT_5 / TP1 (Keyb) | 16 | MAT_6 / TP2 (Keyb) |
| 17 | MAT_7 / TP3 (Keyb) | 18 | MAT_8 / TP4 (Keyb) |
| 19 | Ground (Keyb) | 20 | +3V3 (Keyb) |

X28 TFT

Connector socket to connect a TFT display.

X29 IR

Connector for IR receiver.

| Pin | Assignment |
|-----|------------|
| 1 | GND |
| 2 | VCC |
| 3 | IR_RCV |

X30 RS422/485

| Pin | Assignment | Pin | Assignment |
|-----|------------|-----|------------|
| 1 | 5V VCC | 2 | 12V VCC |
| 3 | GND | 4 | GND |
| 5 | TXD- | 6 | TXD+ |
| 7 | RXD- | 8 | RXD+ |
| 9 | RGND | 10 | 3V3 VCC |

X32 Res

Portux Panel-PC reset switch. Short-circuit to perform a system restart.

X33 Buzzer

Strip to connect an 3.3V inductive buzzer.

X36 USB host

| Pin | 1 | 2 | 3 | 4 | 5 |
|------------|--------|----|----|-----|----|
| Assignment | 5V VCC | D- | D+ | GND | NC |

X37 One Wire Bus

| Pin | Assignment |
|-----|------------|
| 1 | VCC |
| 2 | DATA |
| 3 | GND |

X39 TFT Power Supply

Connector for a high voltage power supply pack to power a TFT display.

X40 USB device

| Pin | 1 | 2 | 3 | 4 | 5 |
|------------|------|----|----|-----|----|
| Assignment | VBUS | D- | D+ | GND | NC |

X42 PS2

| Pin | 1 | 2 | 3 | 4 | 6 |
|------------|------|-------|-----|-----|-----|
| Assignment | DATA | RESET | GND | VCC | CLK |

SW1

4 row DIP switch. Position ON pulls the corresponding I / O line to LOW level. Before the DIP switch is uses it necessary to activate the integrated pull-up resistors (see chapter 3.3).

| Switch | I/O line |
|---------------|-----------------|
| 1 | PB12 |
| 2 | PB13 |
| 3 | PB14 |
| 4 | PB15 |

5. Hardware extensions

5.1. PXB description

The PXB – **Portux Extension Bus** consists of a 96-pin strip.

By the PXB a 16-Bit data line as well as a 26-Bit address line, USB host, USB device and various I/O lines of the Atmel® AT91RM9200 are made available.

The I/O lines are assigned to the PIO controllers A, B and C of the Atmel® AT91RM9200 and may be configured for general purpose I/O or assigned to a function of an integrated peripheral device (see chapter 3.5).

The following table shows the usable interfaces across the PXB bus:

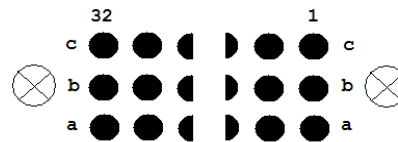
| Peripheral | Signals | Multiplexing | Portux | Device |
|------------|----------------|-------------------------------------|--------|-------------------|
| EBI | A0-A25 | A23 – PC7 A24 – PC8 A25 – PC9 | Yes | SDRAM Flash |
| | D0-D15 | – | Yes | SDRAM Flash |
| | NWR0 | – | Yes | Flash |
| | NWR1 | – | No | – |
| | NWR3 | – | No | – |
| | NCS3 | – | No | – |
| | NCS4 | PC10 | No | – |
| | NCS5 | PC11 | No | – |
| | NCS6 | PC12 | No | – |
| | NCS7 | PC13 | No | – |
| | NRD | – | Yes | Flash |
| | BFCK | PC0 | Yes | LED1 |
| | BFRDY/ SMOE | PC1 | Yes | LED2 |
| | BFAVD | PC2 | No | – |
| | BFBAA/ AMWE | PC3 | No | – |
| | BFOE | PC4 | No | – |
| BFWE | PC5 | No | – | |
| USB Host | HDMA | – | No | – |
| | HDPA | – | No | – |
| USB Device | DDM | – | No | – |
| | DDP | – | No | – |
| SPI | MISO | PA0 | Yes | Dataflash (NPSC0) |

| Peripheral | Signals | Multiplexing | Portux | Device |
|-------------------------|---------|--------------|--------|----------------------------------|
| | MOSI | PA1 | Yes | Dataflash (NPSC0) |
| | SPCK | PA2 | Yes | Dataflash (NPSC0) |
| | NPSC1 | PA4 | No | - |
| | NPSC2 | PA5 | No | - |
| | NPSC3 | PA6 | No | - |
| TWI/I ² C | TWD | PA25 | Yes | RTC TWI-EEPROM (not equipped) |
| | TWCK | PA26 | Yes | RTC TWI-EEPROM (not equipped) |
| USART2 | RXD2 | PA22 | No | - |
| | TXD2 | PA23 | No | - |
| | SCK2 | PA24 | No | - |
| USART3 | RXD3 | PA5 | No | - |
| | TXD3 | PA6 | No | - |
| | SCK3 | PB2 | No | - |
| | RTS3 | PB0 | No | - |
| | CTS3 | PB1 | No | - |
| Compact-Flash Interface | CFOE | - | No | - |
| | CFWE | - | No | - |
| | CFIOR | - | No | - |
| | CFIOW | - | No | - |
| | CFRNW | PC9 | No | - |
| | CFCS | PC10 | No | - |
| | CFCE1 | PC11 | No | - |
| | CFCE2 | PC12 | No | - |
| | NWAIT | PC6 | No | - |
| Others | FIQ | - | No | - |
| | NRST | - | Yes | - |
| I/O | PB6 | - | No | - |
| | PB7 | - | No | - |
| | PB8 | - | No | - |
| | PB9 | - | No | - |
| | PB10 | - | No | - |
| | PB11 | - | No | - |
| | PB17 | - | No | - |

| Peripheral | Signals | Multiplexing | Portux | Device |
|------------|---------|--------------|--------|--------|
| | PB27 | - | No | - |
| | PC5 | - | No | - |
| | PC14 | - | No | - |

Units that are operated by the EBI can not be used concurrent. For example if you decide to implement a compact flash interface, the chip selects 4-6 are not available for connecting another static memory (e.g. Flash).
 If you decide to use the USART3, the chip selects 2 and 3 of the SPI-bus are not available. So you can only connect one further SPI device in addition to the serial data flash on the circuit board to chip select 1.

5.2. PXB pin assignment



The 3 32-pin stripes are marked with a, b and c on the circuit board, the single pins are numbered serially from 1 to 32. The assignment of the pins is shown in the following table.

PXB pin assignment row a:

| Pin | Assignment | Function | Device |
|-----|------------|--------------|----------|
| 1 | GND | GND | - |
| 2 | VCC | VCC | - |
| 3 | NWR0 | NWR0 | EBI |
| 4 | NCS3 | NCS3 | EBI |
| 5 | A1 | A1 | - |
| 6 | D0 | D0 | - |
| 7 | D2 | D2 | - |
| 8 | D4 | D4 | - |
| 9 | D6 | D6 | - |
| 10 | PC6 | I/O NWAIT | - - |
| 11 | A3 | A3 | - |
| 12 | PC11 | I/O NCS5 | - - |
| 13 | PB28 | I/O FIQ | - - |
| 14 | HDMA | HDMA | USB-Host |
| 15 | HDPa | HDPa | USB-Host |

| Pin | Assignment | Function | Device |
|------------|-------------------|----------------------|--------------------------------|
| 16 | DDM | DDM | USB-Dev |
| 17 | PA25 | I/O TWD IRQ2 | - TWI/I ² C - |
| 18 | VCC | VCC | - |
| 19 | PA00 | I/O MISO PCK3 | - SPI PIO-Clock |
| 20 | PA02 | I/O SPCK IRQ4 | - SPI - |
| 21 | PB27 | I/O PCK0 | - PIO-Clock |
| 22 | PC05 | I/O BFWE | - Burst Flash |
| 23 | VCC | VCC | - |
| 24 | PB08 | I/O TD1 TIOA4 | - SSC1 TC4 |
| 25 | PA23 | I/O TXD2 IRQ3 | - USART2 - |
| 26 | PC14 | I/O | - |
| 27 | PB09 | I/O RD1 TIOB4 | - SSC1 TC4 |
| 28 | VCC | VCC | - |
| 29 | PB10 | I/O RK1 TIOA5 | - SSC1 TC5 |
| 30 | PA05 | I/O NPCS2 TXD3 | - SPI USART3 |
| 31 | PB00 | I/O TF0 RTS3 | - SSC0 USART3 |
| 32 | PB11 | I/O RF1 TIOB5 | - SSC1 TC5 |

PXB pin assignment row b:

| Pin | Assignment | Function | Device |
|-----|------------|----------|--------|
| 1 | D8 | D8 | - |
| 2 | D9 | D9 | - |
| 3 | D10 | D10 | - |
| 4 | D11 | D11 | - |
| 5 | D12 | D12 | - |
| 6 | D13 | D13 | - |
| 7 | D14 | D14 | - |
| 8 | D15 | D15 | - |
| 9 | A4 | A4 | - |
| 10 | A5 | A5 | - |
| 11 | A6 | A6 | - |
| 12 | A7 | A7 | - |
| 13 | A8 | A8 | - |
| 14 | A9 | A9 | - |
| 15 | A10 | A10 | - |
| 16 | A11 | A11 | - |
| 17 | A12 | A12 | - |
| 18 | A13 | A13 | - |
| 19 | A14 | A14 | - |
| 20 | A15 | A15 | - |
| 21 | A16 | A16 | - |
| 22 | A17 | A17 | - |
| 23 | A18 | A18 | - |
| 24 | A19 | A19 | - |
| 25 | A20 | A20 | - |
| 26 | A21 | A21 | - |
| 27 | A22 | A22 | - |
| 28 | A23 | A23 | - |
| 29 | A24 | A24 | - |
| 30 | A25 | A25 | - |
| 31 | NWR1 | NWR1 | EBI |
| 32 | NWR3 | NER3 | EBI |

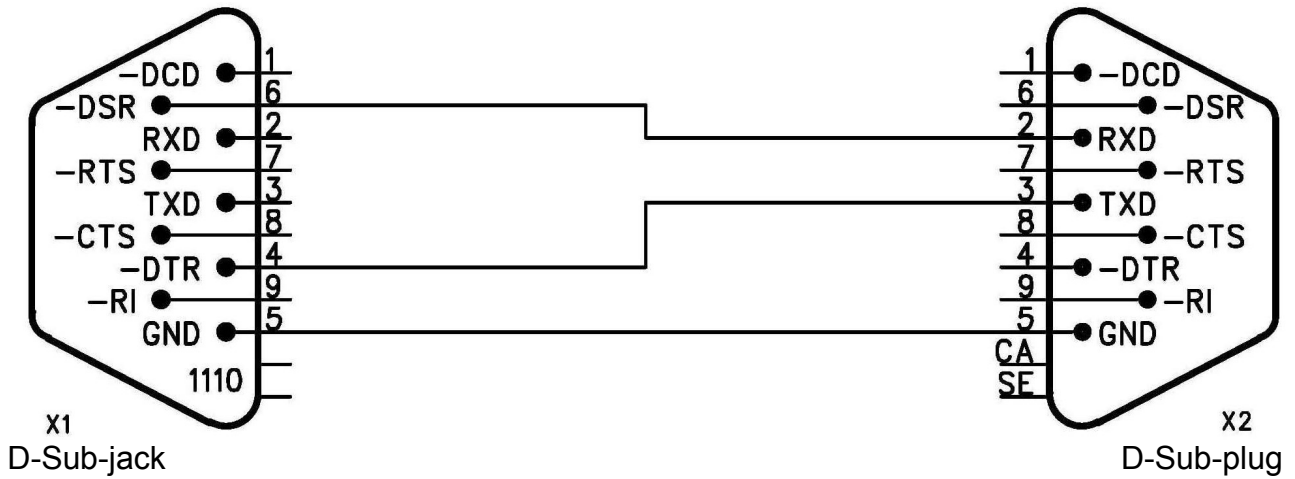
PXB pin assignment row c:

| Pin | Assignment | Function | Device |
|-----|------------|----------------------|--------------------------------|
| 1 | GND | GND | - |
| 2 | VCC | VCC | - |
| 3 | NRD | NRD | EBI |
| 4 | A0 | A0 | - |
| 5 | NRST | NRST | |
| 6 | D1 | D1 | - |
| 7 | D3 | D3 | - |
| 8 | D5 | D5 | - |
| 9 | D7 | D7 | - |
| 10 | A2 | A2 | - |
| 11 | PC10 | NCS4/ CFCS | - |
| 12 | PC12 | NCS6/ CFCE1 | - |
| 13 | GND | GND | - |
| 14 | PC00 | I/O BFCK | - Burst Flash |
| 15 | PC01 | I/O BFRDY/SMOE | - Burst Flash |
| 16 | DDP | DDP | USB-Dev |
| 17 | PA26 | I/O TWCK IRQ1 | - TWI/I ² C - |
| 18 | GND | GND | - |
| 19 | PA01 | I/O MOSI PCK3 | - SPI PIO Clock |
| 20 | PA04 | I/O NPCS1 PCK1 | - SPI PIO Clock |
| 21 | PC13 | I/O NSC7 | - - |
| 22 | PB17 | I/O RF2 ETX2 | - SSC2 ETH |
| 23 | PA24 | I/O SCK2 TIOA1 | - USART2 TC1 |
| 24 | PA22 | I/O RXD2 TIOB2 | - USART2 TC2 |

| Pin | Assignment | Function | Device |
|-----|------------|----------------------|---------------------|
| 25 | PB02 | I/O TD0 SCK3 | - SSC0 USART3 |
| 26 | PC15 | I/O | - |
| 27 | GND | GND | - |
| 28 | PB06 | I/O TF1 TIOA3 | - SSC1 TC3 |
| 29 | PA06 | I/O NPCS3 RXD3 | - SPI USART3 |
| 30 | PB07 | I/O TK1 TIOB3 | - SSC1 TC3 |
| 31 | PB01 | I/O TK0 CTS3 | - SSC0 USART3 |
| 32 | GND | GND | - |

6. Appendix

6.1. DBGU-adapter



Jack and plug will be installed into an empty adapter case (RS232), the connections can be soldered with flexible wire.