# CPC316

**CPU Module**

# User Manual

Rev. 002

September 2021

*The product described in this manual is compliant with all related CE standards.*

Product Title: CPC316
Document name: CPC316 User Manual
User Manual version: 002
Ref. docs:

Copyright © 2021 Fastwel Co. Ltd. All rights reserved.

## Revision Record

| Rev. Index | Brief Description | Product Index | Date |
|---|---|---|---|
| 001 | Preliminary version | CPC316 | April 2020 |
| 002 | Compliance assessment | CPC316 | September 2021 |
| | | | |

## Contact Information


| | **Fastwel Co. Ltd** | | **Fastwel Corporation US** |
|---|---|---|---|
| Address: | 108 Profsoyuznaya St.,<br>Moscow 117437,<br>Russian Federation | | 6108 Avenida Encinas,<br>Suite B, Carlsbad,<br>CA92011, USA |
| Tel.: | +7 (495) 232-1681 | Tel.: | **+1 (858) 488-3663** |
| Fax: | +7 (495) 232-1654 | | |
| E-mail: | info@fastwel.com | E-mail: | info@fastwel.com |
| Web: | http://www.fastwel.com/ | | |

# Table of Contents

# List of Tables

# List of Figures



All information in this document is provided for reference only, with no warranty of its suitability for any specific purpose. This information has been thoroughly checked and is believed to be entirely reliable and consistent with the product that it describes. However, Fastwel accepts no responsibility for inaccuracies, omissions or their consequences, as well as liability arising from the use or application of any product or example described in this document.

Fastwel Co. Ltd. reserves the right to change, modify, and improve this document or the products described in it, at Fastwel's discretion without further notice. Software described in this document is provided on an “as is” basis without warranty. Fastwel assumes no liability for consequential or incidental damages originated by the use of this software.

This document contains information, which is property of Fastwel Co. Ltd. It is not allowed to reproduce it or transmit by any means, to translate the document or to convert it to any electronic form in full or in parts without antecedent written approval of Fastwel Co. Ltd. or one of its officially authorized agents.

Fastwel and Fastwel logo are trademarks owned by Fastwel Co. Ltd., Moscow, Russian Federation. CompactPCI is a trademark of the PCI industrial Computers Manufacturers Group. Ethernet is a registered trademark of Xerox Corporation. IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers Inc. Intel is a trademark of Intel Corporation. Pentium M and Celeron M are trademarks of Intel Corporation. Microsoft is a trademark of the Microsoft Corporation. In addition, this document may include names, company logos and trademarks, which are registered trademarks and, therefore, are property of their respective owners.

Fastwel welcomes suggestions, remarks and proposals regarding the form and the content of this Manual.

This User Manual is designed for providing you with the operating principle and general information required for commissioning, intended use and maintenance of the CPC316 CPU module IMES.467444.114 (hereinafter referred to as the “device” or the “product”) manufactured by Fastwel Group.

The device is made in PC/104-plus format and is designed for use in various embedded systems that require operation in an extended temperature range (from -40 to +85 °C), compatibility of applications with x86 architecture of CPUs, as well as a combination of high performance and low level of generated thermal power and power consumption.

The document is designed for developers of distributed control and data collection systems, process control systems and embedded control systems, for system administrators and engineers of the industrial automation area.

## Ownership rights

This document contains information that is the property of Fastwel Group. It may not be copied or transmitted by any known means, nor may it be stored in information storage and retrieval systems without the prior written consent of Fastwel Group or one of its authorized agents. The information contained in this document, to the best of our knowledge, is free from any errors, however, Fastwel Group accepts no responsibility for any inaccuracies and their consequences, as well as liability arising from the use or application of any diagram, product or example given in this document. Fastwel Group reserves the right to change and improve both this document and the product presented inside of it at its discretion without any further notice.

## Safety requirements

This Fastwel Group's product is developed and tested for the purpose of ensuring compliance to the electric safety requirements. Its design provides long-term trouble-free operation. The service life of the product can be significantly reduced due to the improper handling during unpacking and installation. Therefore, in the interests of your safety and in order to ensure proper operation of the product, you should follow the recommendations below.

# Conventions

**Caution, High Voltage!**

This sign and text warn of the dangers associated with electrical discharges (> 60 V) when touching the device or any part of it. Failure to follow the precautions mentioned or prescribed in the regulations may endanger your life or health, and may result in damages to the equipment. Please also read the below subparagraph dedicated to the rules for working with high voltage.

**Attention! Static-sensitive device!**

This sign and text indicate that electronic boards and their components are sensitive to static electricity, so care should be taken when handling this device and performing inspections to ensure integrity and functionality of the device.

**Attention! Hot surface!**

This sign and text warn of the danger associated with touching hot surfaces of the device.

**Attention!**

This sign is aimed at drawing your attention to aspects of this User Manual that, if not fully understood or ignored, may endanger your health or cause damages to the equipment.

**Note**

This sign is used to text fragments that should be read carefully.

## Safety requirements

This Fastwel Group product has been developed and tested to ensure compliance with electrical safety requirements. Its design provides long-term fail-safe operation. The product's life cycle may be significantly shortened due to mishandling during unpacking and installation. Therefore, for your own safety and for ensuring proper operation of the device, you should follow the recommendations given below.

## Rules for safe handling with high voltage

**Attention!**

All operations with this device should only be performed by personnel with sufficient qualifications.

**Caution, High Voltage!**

Before installing the board in the system, make sure that the mains power supply is off. The same also applies to the installation of expansion boards.

There is a serious risk of electric shock during installation, repairs, and maintenance of the device, so always unplug the power supply cord while carrying out of works. The same also applies to the other power supply cables.

## Instructions for board handling

**Static-sensitive device!**

Electronic boards and their components are sensitive to static electricity. Therefore, special attention should be given when handling these devices to ensure their safety and operability.

✓ Do not leave the board in the non-operating position without protective packaging.

✓ If possible, always work with the board in workplaces protected against static electricity. Should this not be possible, the user should remove the static charge before touching the product with their hands or tools. The best way to do so is by touching any metal part of the system enclosure.

✓ It is especially important to observe precautions when replacing expansion boards, memory modules, jumpers, etc. If the product is equipped with batteries to power the memory or real-time clock, avoid placing the board on conducting surfaces such as anti-static mats or sponges. They can cause short circuits and damage the battery and the board's conducting circuits.

## General rules of usage

✓ In order to keep the warranty, the product must not be altered or changed in any way. Any changes and improvements unauthorized by Fastwel Group other than those contained in this User Manual or received from the technical support service of Fastwel Group in the form of a set of instructions for their implementation will void the warranty.

✓ This device should be installed and connected only to systems that meet all necessary technical and climatic requirements. This also applies to the operating temperature range of a particular version of the board. You should also consider the temperature limits of the batteries installed on the board.

✓ Follow the instructions in this User Manual only when performing all necessary installation and configuration operations.

✓ Retain the original packaging for storing products in the future or to transport in case of a warranty claim. If it is necessary to transport or store the board, pack it the same way as it was packed at the time of receipt.

✓ Proceed with extra caution when handling and unpacking the device. Follow the instructions given below.

## MANUFACTURER'S WARRANTIES

### Warranty liabilities

The manufacturer guarantees that CPC316 meets the requirements of technical specifications of the "CPU MODULE IN PC104 FORMAT" TU 4013-004-52415667-05 provided that the Consumer complies with the operating conditions, transportation, storage, installation and mounting, set by the operational documents.
The manufacturer guarantees that the products supplied by it will not show any manufacturing defects and materials used in compliance with the rules of operation and maintenance during the warranty period established at the moment. The Manufacturer's obligation under this warranty is to repair or replace, free of charge, any defective electronic component included in the returned product.
Products that failed through the Manufacturer's fault during the warranty period will be repaired free of charge. In other cases, the Consumer will be billed based on current remuneration rates and the cost of consumables.

### Right of limitation liability

The manufacturer is not responsible for any damages caused to the Consumer's property due to the failure of the product in the process of its use.

### Warranty period

The warranty period for the manufacturer's products is 36 months from the date of sale (unless otherwise provided by the delivery agreement).
For customized products, the warranty period is 60 months from the date of sale (unless otherwise provided by the delivery agreement).

**The warranty set forth above does not extend to and shall not apply to:**

1. Products, including software, which have been repaired or altered by other than Fastwel personnel, unless Buyer has properly altered or repaired the products in accordance with procedures previously approved in writing by Fastwel.
2. Products, which have been subject to power, supply reversal, misuse, neglect, accident, or improper installation.

**Returning a product for repair**

1. Apply to Fastwel Company or to any of the Fastwel's official representatives for the Product Return Authorization.
2. Attach a failure inspection report with a product to be returned in the form, accepted by customer, with a description of the failure circumstances and symptoms.
3. Carefully package the product in the antistatic bag, in which the product had been supplied. Failure to package in antistatic material will VOID all warranties. Then package the product in a safe container for shipping.
4. The customer pays for shipping the product to Fastwel or to an official Fastwel representative or dealer.

# Transportation, Unpacking and Storage

## Transportation

The modules should be transported in the separate manufacturer's packaging (container), consisting of an individual antistatic packaging and a cardboard box, in closed transport (road, rail, air in heated and sealed compartments) under storage conditions 5 according to the GOST standard 15150-69 or under storage conditions 3 for sea transportation.

It is allowed to transport the modules packed in individual antistatic bags in manufacturer's group packaging (container).

Transportation of packaged modules should be carried out in accordance with the rules of transportation of goods currently valid for this type of transport.

During handling operations and transportation, the packed modules should not be subjected to sudden shocks, drops, impacts and precipitation. The packed modules should be placed on the vehicle in such a way as to preclude their further movements.

## Unpacking

Before unpacking, after transportation at negative ambient temperatures, the modules must be kept for 6 hours under storage conditions 1, in accordance with the GOST standard 15150-69.

Do not place the packed modules near any heat sources before unpacking.

When unpacking the modules, it is necessary to observe all precautions to ensure their safety, as well as marketable condition of manufacturer's consumer packaging.

When unpacking, it is necessary to check the modules for any external mechanical damages after transportation.

## Storage

Storage conditions of the modules 1 comply with the GOST standard 15150-69.

# 1 Description and operation

## 1.1 Purpose of the device

The device is a highly integrated solution based on the x86 platform for the use in real time systems of management, production control, data acquisition and processing and can operate in the standalone and slave mode. Connection of the main I/O means (VGA monitor, LVDS panel, keyboard, printers, USB) enables the device to be used in systems with operator participation. For data storage, both an integrated storage device or a Compact Flash card, and external USB devices can be used.

The product is made in PC/104-plus format and enables you to solve most of the tasks of input-output of digital signals. In addition, the device makes it possible to promptly replace the operating software of the built-in FPGA, which gives additional opportunities for its adaptation to specific tasks.

Expanding device's functionality is also possible by connecting additional expansion modules of the PC/104 and PC/104-plus format.

The product can be connected to RS-232, RS-485 and Ethernet networks, which makes it possible to use it in distributed input-output and information processing systems.

## 1.2 Technical specifications

**Vortex86DX3 CPU (x86-compatible instruction set):**

- Clock rate: 800 MHz;
- Number of physical cores: 2;
- 32 x bit x86 core;
- 32 bit memory bus;
- L1 cache (64 KB);
- L2 cache (512 KB);

**RAM:**

- DDR3 SDRAM 2 GB (soldered onboard);
- Memory bus width: 32 bit;
- Memory bus operating frequency: 667 MHz;

**Video subsystem:**

- video controller with 2D accelerator;
- port for connection of VGA monitor with resolution up to 1920x1080, color 32 bit;
- port for connection of LVDS panel with resolution up to 1920x1080, color 32 bit;

**PCI104 bus (PCI 33 MHz 32 bit);**

**PC104 bus (ISA 8/16 bit, 8/16 MHz);**

- **Non-volatile RAM:**

- volume: 32 KB;
- implemented using the FRAM technology, SPI interface;
- battery power supply is not required;

- **Flash-drive:**

- connected to SATA interface;
- 8 GB NAND Flash (pSLC);
- read/write rate of 100/80 MB/s;
- integrated system of wear control and ECC;

- **LAN ports:**

- 1 x Fast Ethernet 10/100 Mb/s port;
- 1 x Gigabit Ethernet 10/100/1000 Mb/s port;
- system isolation of 500V;

- **USB ports (host):**

- support of USB 1.1, USB 2.0 (HS, FS, LS);
- connection of up to 4 devices;

- **Connector for Compact Flash (Type I/II):**

- support of UltraDMA-5 mode;

- **Serial ports:**

- COM1: RS-232 (9-wire);
- COM2: RS-232 (9-wire);
- exchange rate over RS-232: up to 115.2 Kb/s;
- protection against ESD 15 kV (IEC1000-4-2);
- COM3: isolated RS-422/485 (individual system isolation 500V);
- COM4: isolated RS-422/485 (individual system isolation 500V);
- automatic (hardware) control of transmission direction for RS-485 ports;
- exchange rate over RS-422/485: up to 115.2 Kb/s [1]

- **GPIO port**

- 8 x I/O lines;
- compatibility with the level +5V (TTL);

- **Universal digital I/O port[2]:**

- 48 CMOS/TTL lines;
- output load up to 24 mA;
- possibility of changing functions (scheme) of port within the system;
- power supply voltage output +5 V @ 0.75 A for connected devices;

- **Watchdog timers:**

- 2 watchdog timers, built into the processor, with a programmable event and an actuation interval of 30.5 μs…512 s;

- **Real Time Clock:**

- consumption current when power is off: 2 μA[3]

- **Integrated lithium battery 3V:**

- CR2032, standard capacity 200 mA*h;

- **Buzzer**

- **Isolated port of reset / interrupt source**

[1] Exchange rate over serial ports is defined by the frequency divider register.
[2] The electrical parameters of the digital input/output channels are determined by the parameters of the installed buffer microcircuit SN74LVC1T45DCKTE4 (Texas Instruments, www.ti.com).
[3] Standard value under normal conditions.

- **Digital temperature sensor**

Measuring the temperature of the CPU board from - 55 to +125 °C, typical absolute error of temperature measurement[4]:
± 0.5 °C (within the range from +10 to +50 °C);
± 1.0 °C (within the range from - 10 to +85 °C);
± 2.0 °C (within the range from -40 to +85 °C);
resolution and price per unit for the least significant bit: 12 bit + character / 0.0625 °C
conversion time is up to 1000 ms.

- **Measuring secondary supply voltages of the module by an integrated ADC**
- **Compatibility with operating systems:**

- FreeDOS, Microsoft™ MS-DOS® 6.22
- Linux 2.6, 3.2
- Windows Embedded Standard 7

- **Console serial ports:** COM1 / COM2 / COM3 / COM4;
- **Power supply voltage:** from 4.25 to 5.25 V
- **operating temperature range:** from -40 to +85 °C

**Storage conditions for the modules:** 1 according to the GOST standard 15150-69

- **Humidity:** from 5 to 95%, at +25 °C, non-condensing;
- **Resistance to multiple/single shocks:** 50/150 g;
- **Vibration resistance:** 10 g for frequencies from 50 to 2000 Hz;
- **MTBF [5]:** min. 100,000 hours;
- **Dimensions, max:** 116 x 101.56 x 24.13 mm;
- **Module's weight, max.:** 0.2 kg;
- **Packed weight, max.:** 0.4 kg;

## 1.3 Connection to the device

Below is the typical list of interface boards and devices, which may be connected to the device:

- Devices with Ethernet 10 / 100 / 1000 Mb/sec interface;
- Devices compatible with RS-232;
- RS-485 multiuser networks;
- Compact Flash memory cards;
- USB devices, type 1.1 and 2.0 (Full-speed, High-speed), including devices of the USB Mass Storage Device type;
- Keyboard, mouse (USB ports);
- Monitors and TFT panels;

[4] The measurement accuracy of integrated sensors is not standardized and is determined by the data in the chip manufacturer's documentation.
[5] The MTBF value is calculated using the Telcordia Issue 1 calculation model (Method I Case 3) for continuous operation when located on the ground under conditions corresponding to the climatic category Moderately Cold Climate 4 in accordance with GOST 15150-69 standard, at an ambient temperature of +30°C.

- PC-compatible printer (USB port);
- RTU modules with galvanic isolation (via the MPB-24 or TBI-24/16L and TBI-24LC boards);
- Terminal boards for the digital I/O port for passing from IDC-xx and IDC2-xx connectors to WAGO terminal strips: TIB972.

### 1.3.1 Terminal Board TIB972

TIB972-01 – is a 26 pin terminal board: WAGO terminal strip of the 736 series, IDC2-26 connectors (pitch of 2 mm) and IDC-26 (pitch of 2.54 mm).
Pin numbers of the WAGO terminal strip, series 736, are shown on the PCB.

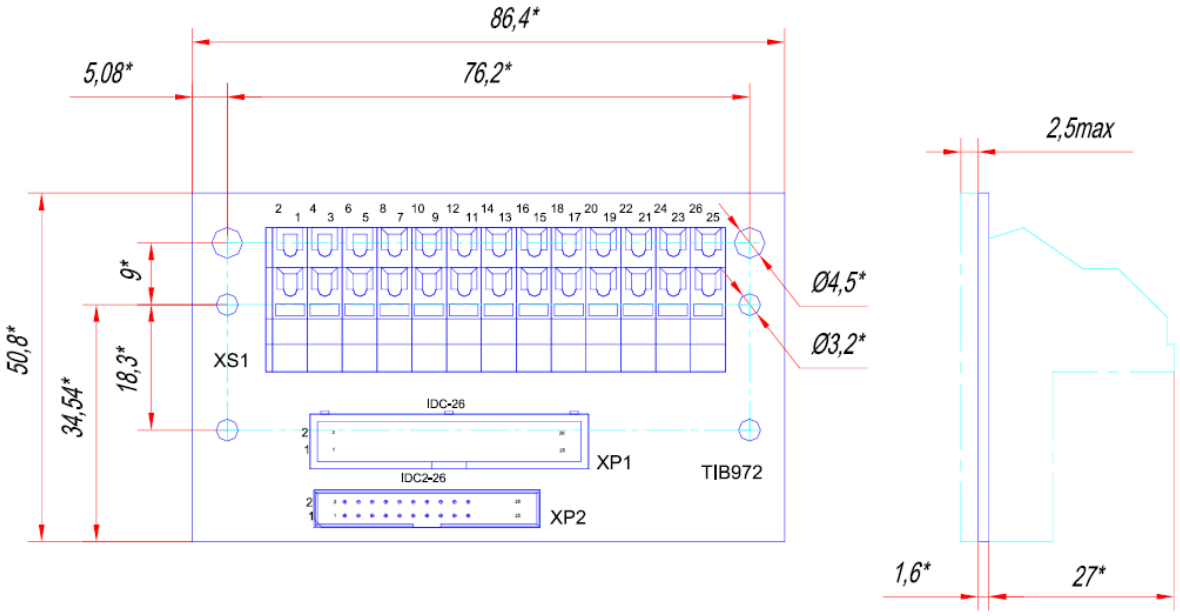



**Fig 1-1 – Terminal board TIB972-01**

**Table 1-1 – Terminal board TIB972-01 (Table of pins)**

| WAGO 736-413 | IDC2-26 | IDC-26 |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| … | … | … |
| 26 | 26 | 26 |

## 1.4 Power Supply

The device's power supply should meet the requirements specified in Table 1-2.

The device is powered through the PC/104 and PC/104-plus connectors (if installed). If you need to supply power from an external source, you can use the additional XP11 power supply connector (4-pin AMP 4-171826-4 connector).

The power supply should provide the starting current specified in Table 1-2. It is also permitted to use a power supply with a current limiting mode for at least 3.0 A (excluding any peripherals connected). When choosing a power supply, the starting current and the current consumption of expansion modules and other devices connected to the device ports, should be considered.

The device is equipped with an active protection circuit against short circuit and against supplying reverse voltage. There is also a surge protection against overvoltage above 30 V (direct or reverse polarity) by parallel connection of the TVS protection diode SM6T33CA. In case of long-term supply of more than 30 V supply voltage (direct or reverse polarity), the protective diode may fail. There are following consumers of the +5 V input power supply:

1) Device (+5V @ 1.5A);

2) External equipment connected to USB ports (4x ports, maximum consumption of each port is +5V @ 0.5A);

3) External equipment, connection to UNIO ports (2 x ports, maximum consumption of each port is +5V @ 0.75A);

4) LVDS panel (1 x port, maximum consumption of +3.3V @ 0.75A);

5) External equipment, connection to GPIO port (maximum consumption of +5V @ 0.75A);

6) External equipment, connected to RS-232 ports (2 x ports, maximum consumption of +5V @ 0.75A).

The maximum possible load-carrying capacity for the "+5 V" power supply voltage is 6 A. If the value is exceeded, the active protection of the device will be triggered and the input power will be disconnected, which will be automatically restored when the load current falls below the limit value, in which case the device will restart.

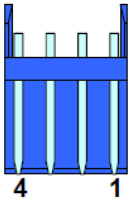
**Table 1-2 - Requirements for the external power supply parameters**

| Version | Power supply voltage, V | Voltage limiting values, V | Load current, A | Start-up current, A |
|---|---|---|---|---|
| CPC316-01<br>CPC316-02<br>without peripherals | +5V | from +4.75 to +5.25 | 2.0 | 3.0 |

Recommended mating part for additional XP11 power connector (AMP 4-171826-4): AMP 4-171822-4 and AMP 170263-1 sets of pins (available for order as ACS00038 kit (socket and pins) or as ACS00038-01 kit (socket, pins and 1 m long wires).

**Table 1-3 – Purpose of the power supply connector pins**

| Power supply connector: AMP 4-171822-4 | |
|---|---|
| **Pin#** | **Function** |
| **1** | **+5 V** |
| **2** | **GND** |
| **3** | **GND** |
| **4** | **-** |



## 1.5 Block diagram

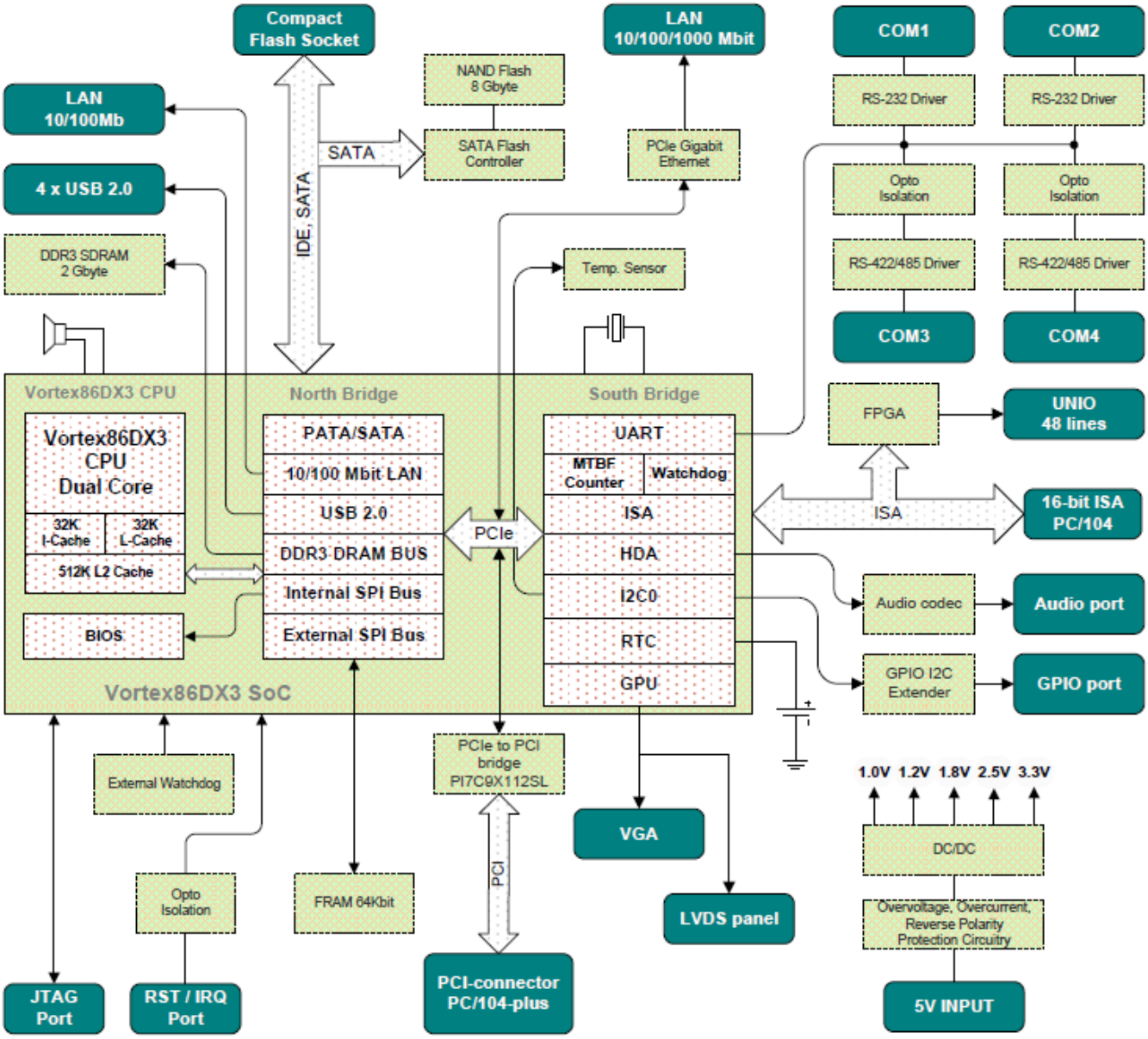



**Fig. 1-2 – Block diagram of the device**

## 1.6 Location of main elements

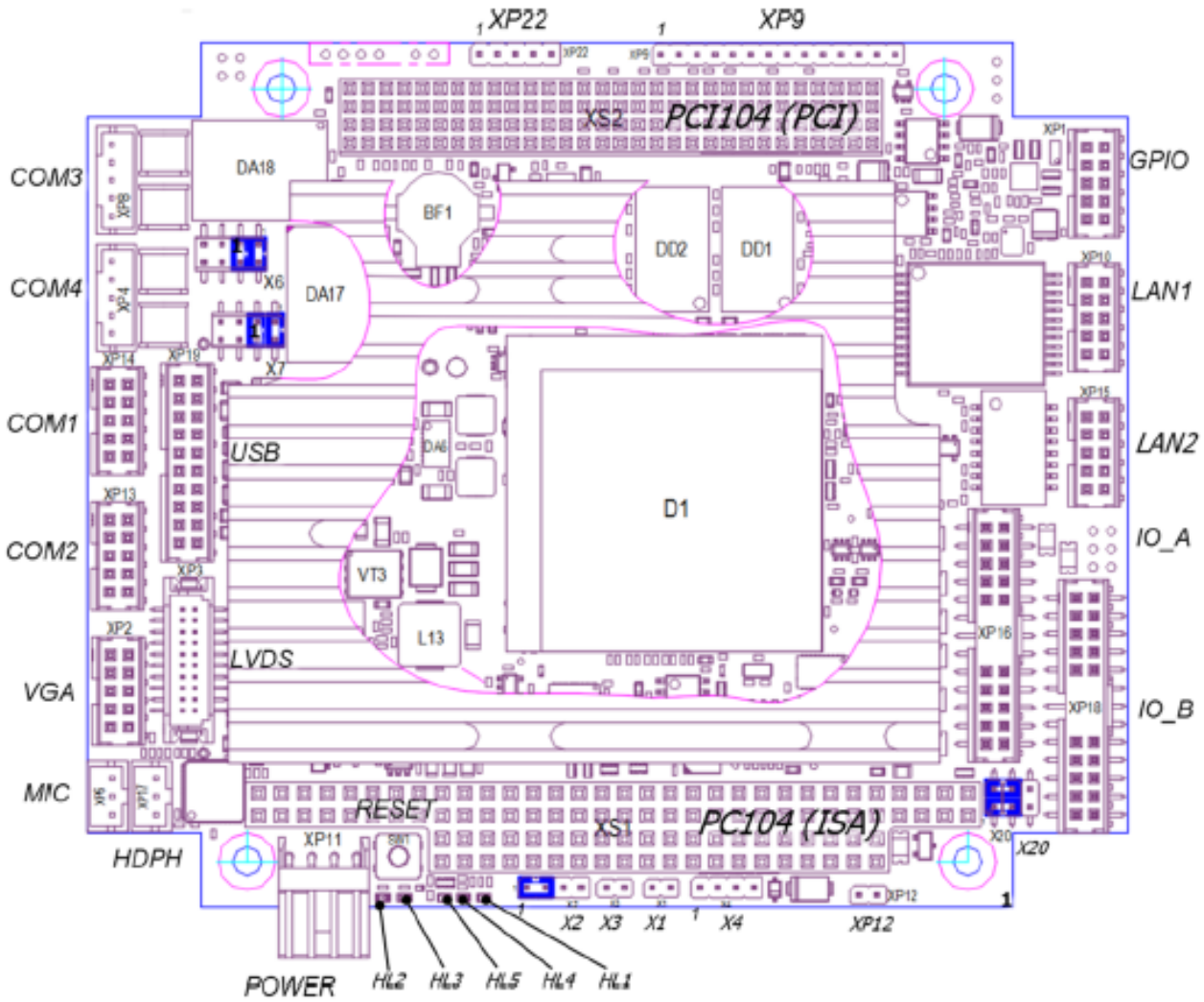



**Fig. 1-3 – Location of main elements of the device on the top side of the board**

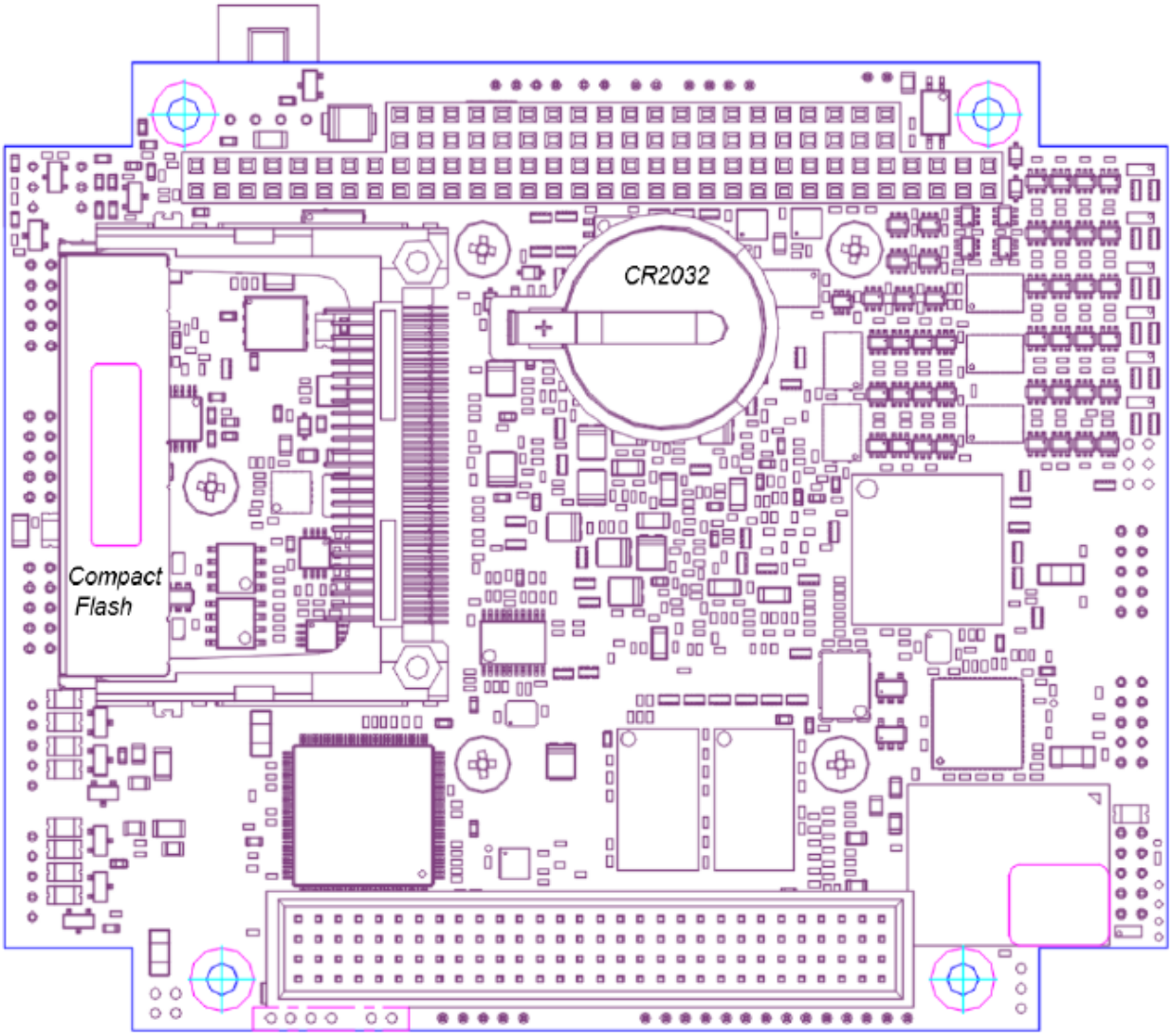



**Fig. 1-4 Location of the main elements of the device on the bottom side of the board**

The assignment of connectors and switches is given in **paragraph 1.9 Structure and functioning** and **paragraph 3.1 Setting the switches**.
The device is made in accordance with the *PC/104-plus* v.2.0 specification.
On the top side, there is a deviation from the specification for the maximum height of components (8.76 mm) in the area of the processor's heatsink: 10.5 mm.

On the bottom side, there is a deviation from the specification for the maximum height of the components (4.83 mm). These components include:

- XP7 connector for Compact Flash (8.3 mm);

- socket X7 for CR2032 battery (4.95 mm).

## 1.7 Versions

| Item No. | Version | Description |
|---|---|---|
| 1 | CPC316-01<br>IMES.467444.114 | Vortex86DX3 CPU (800 MHz Dual Core); 2 GB RAM; 8 GB flash drive, Compact Flash; VGA; LVDS; Audio port; 2 x LAN, 2 x RS-232, 2 x RS-422/485, 4 x USB2.0; 48 x DIO; PC/104-plus format. |
| 2 | CPC316-02<br>IMES.467444.114-01 | Vortex86DX3 CPU (800 MHz Dual Core); 2 GB RAM; 8 GB flash drive, Compact Flash; VGA; 2 x LAN, 2 x RS-232, 2 x RS-422/485, 4 x USB2.0; 48 x DIO; PC/104 format. |

**Note**

The modules with conformal coating obtain the “\COATED” inscription when ordered.

## 1.8 Delivery checklist

The standard delivery checklist includes:

1. CPC316 CPU Module
2. Installation kit – 1 pcs.
3. Packaging.

The installation kit for IMES.467941.053 includes:

1. Jumper – 1 pcs.
2. Rack WE – 4 pcs.
3. DIN7985-M3x6-A2 screw – 4 pcs.
4. DIN985-M3-A2 nut – 4 pcs.
5. DIN125-3,2-A2 washer – 4 pcs.
6. DIN6798A-3,2-A2 washer – 4 pcs.

## 1.9 Structure and functioning

### 1.9.1 Processor

The device is based on a dual-core x86-compatible 32-bit Vortex86DX3 processor with low power consumption, made using 45nm technology. The operating frequency of the processor is 800 MHz. The detailed information on the processor, as well as the current versions of drivers and system software are posted on the manufacturer's website at: http://www.vortex86.com.

### 1.9.2 Supervisor, Watchdog, Reset

The device includes a power supply supervisor (a microchip that monitors the power supply voltage of the device), as well as 3 watchdog timers: 2 watchdog timers integrated into the CPU (WDT0, WDT1). The supervisor generates a hardware reset signal when the "3.3V" power supply voltage drops below 3.08V, as well as an interrupt signal when the “5V” main power supply voltage drops below 4.6V, which makes it possible, if necessary, to save user data in non-volatile RAM (setting interrupts via internal FPGA registers).
The watchdog timer can be used to avoid software freeze-ups. The WDT0 and WDT1 watchdog timers are triggered in the absence of software confirmations for 30.5 µs ... 512 sec. The internal watchdog timer is started in the SYSTEM BIOS SETUP. It is possible to configure the operation of the internal watchdog timer with or without generation of a hardware reset signal.
The device is reset when the power is turned on, by software, as well as by pressing the "RESET" button located on the side panel of the controller.

### 1.9.3 RAM

The device uses dynamic DDR3 SDRAM with a total size of 2 GB and operating at a frequency of 667 MHz as a system memory. The memory module cannot be expanded.

### 1.9.4 NV SRAM

The device has 256 byte integrated non-volatile RAM for storing configuration data (CMOS) written by the SETUP program built into the BIOS.
In addition, it is possible to use the integrated non-volatile memory of 32 KB, made by FRAM technology and does not require battery power. The SPI interface of the processor is used to access the static non-volatile memory.

### 1.9.5 Read Only Memory (FLASH BIOS)

For storing the basic input / output system (hereinafter referred to as BIOS), the device uses flash-memory chip of 2 MB, integrated into the Vortex86DX3 processor chip.

### 1.9.6 Flash drive (SATA NAND Flash)

The device contains a flash memory chip using NAND technology (pSLC). It can be used as a boot drive. To arrange access to the NAND Flash, a controller with an integrated error correction and wear leveling system is used, connected to the SATA interface of the processor.
The integrated drive has a volume of 8GB (real available volume is 7.8GB).
Compact Flash cards and external USB drives can be used as additional drive memory.

### 1.9.7 Slot for Compact Flash cards

The Compact Flash (CF type I / II) can be connected to the device via XP7 (N7E50-M516RB-50, 3M) and set as Master on the Primary IDE channel.
The Compact Flash can be used as the boot drive. Supported mode up to UltraDMA-5.

**Caution!**

WHEN USING THE DEVICE UNDER SEVERE OPERATING CONDITIONS, IT IS NECESSARY TO TAKE ADDITIONAL MEASURES TO FIX THE COMPACT FLASH IN THE CONNECTOR!

External view and overall dimensions of the additional fastener 7E50-C016-00, 3M (not included in the product delivery checklist) are shown in Figure 1-6.

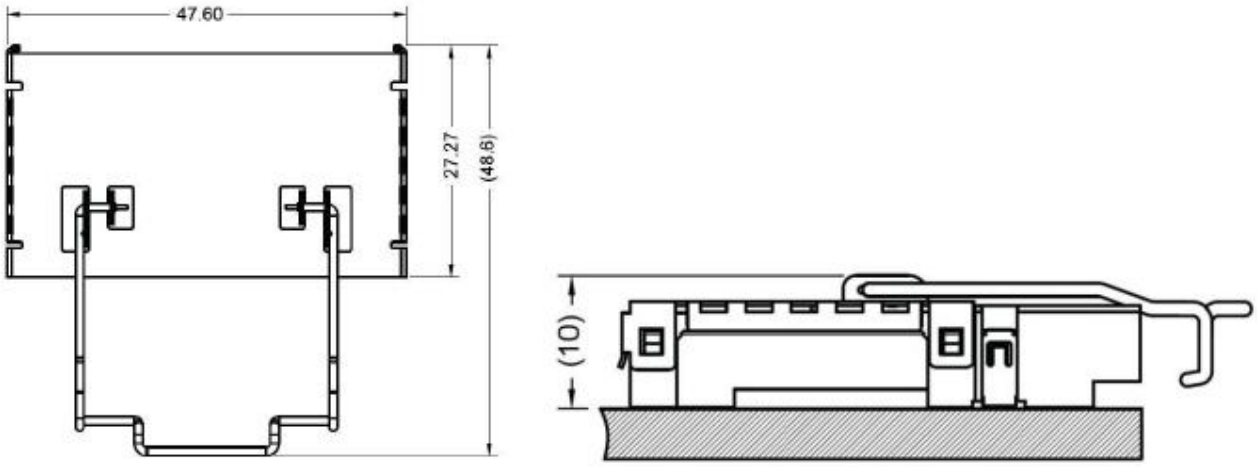



**Fig. 1-5 – Fastener for Compact Flash**

### 1.9.8 COM1-4 serial ports

The device controller has 4 x asynchronous serial ports:

- **COM1, COM2 – RS-232 (5-wire interface, non-insulated);**
- **COM3, COM4 – RS-422/485 (3- or 5-wire connection, insulated 500 V).**

RS-232 COM1 (0x3F8h) and COM2 (0x2F8h) serial ports, as well as RS-422/485 COM3 (0x3E8h) and COM4 (0x2E8h) ports are implemented on the UART controllers integrated into the Vortex86DX3.

The COM1 (RS-232) port of the CP316 module corresponds to the "serial port 5" of the Vortex86DX3 CPU, COM2 (RS-232) – "serial port 6", COM3 (RS-422/485) – "serial port 1", COM4 (RS-422/485) – "serial port 2".

COM3 (XP8) and COM4 (XP4) ports operate in the RS-422/485 mode and provide galvanic isolation up to 500 V (each port has individual isolation from the system). The maximum data transfer rate is 230.4 Kb/sec. The ports are routed to 5-pin B 5B-PH-KL (JST) connectors. For manufacturing the cable, it is recommended to use a PHR-5 socket, JST with SPH-002T-P0.5S pins, JST (available for order as a set ACS00031-01) and a shielded twisted pair.

**Table 1-4 – Purpose oft he pins of RS-422/485 (XP8, XP4) ports**

| COM3-4: B 5B-PH-KL (JST) | | |
|---|---|---|
| **Pin #** | **Function (RS-422)** | **Function (RS-485)** |
| 1 | TX+ | RTxD+ |
| 2 | TX- | RTxD- |
| 3 | RX+ | - |
| 4 | RX- | - |
| 5 | GNDS | GNDS |


The “GNDS” circuit corresponds to the signal isolated “ground” of the interface, the GNDS circuits of the different ports are isolated from each other and are not interconnected.
By setting the X6 (COM3) and X7 (COM4) jumpers, the matching circuits are connected to the signal lines of the RS-422 or RS-485 interfaces and the operating mode is set.

**Attention!**
For proper operation of RS-422/485 transmit/receive unit in a multiuser network, use the appropriate jumpers to install 120 Ohm terminators at the 2 most distant network nodes, as well as the 680 Ohm biasing resistors (at one or two of the most distant network nodes).

**Table 1-5 – Configuration of the COM3 (X6) port**

| **Jumper** | **Description** |
|---|---|
| X6[1-2] [6]<br>X6[3-4] | Biasing resistor is set on the line TX+<br>Biasing resistor is set on the line TX- |
| X6[5-6] | Half-duplex mode is selected [7] |
| X6[7-8] | Half-duplex mode is selected |

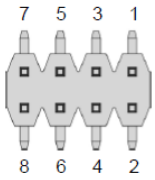
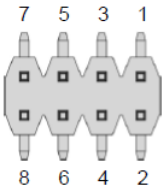



**Table 1-6 – Configuration of the COM4 (X7) port**

| **Jumper** | **Description** |
|---|---|
| X7[1-2] [8] | Biasing resistor is set on the line TX+ |
| X7[3-4] | Biasing resistor is set on the line TX- |
| X7[5-6] | Half-duplex mode is selected [9] |
| X7[7-8] | Half-duplex mode is selected |




[6] The default switch positions are highlighted in bold.
[7] The mode is selected only when the both jumpers are installed.
[8] The default switch positions are highlighted in bold.
[9] The mode is selected only when the both jumpers are installed.

Each port contains lightning protection circuits based on resettable fuses and gas discharge elements. The port also contains TVS diode-based pulse interference protection circuits. The diagram of the output stages of the COM3 port is shown below (the COM4 port has a similar circuit design).
The maximum number of modules connected to the RS-485 network together with the device amounts to 128, provided that the input impedance of the RS-485 drivers is at least 96 kOhm.

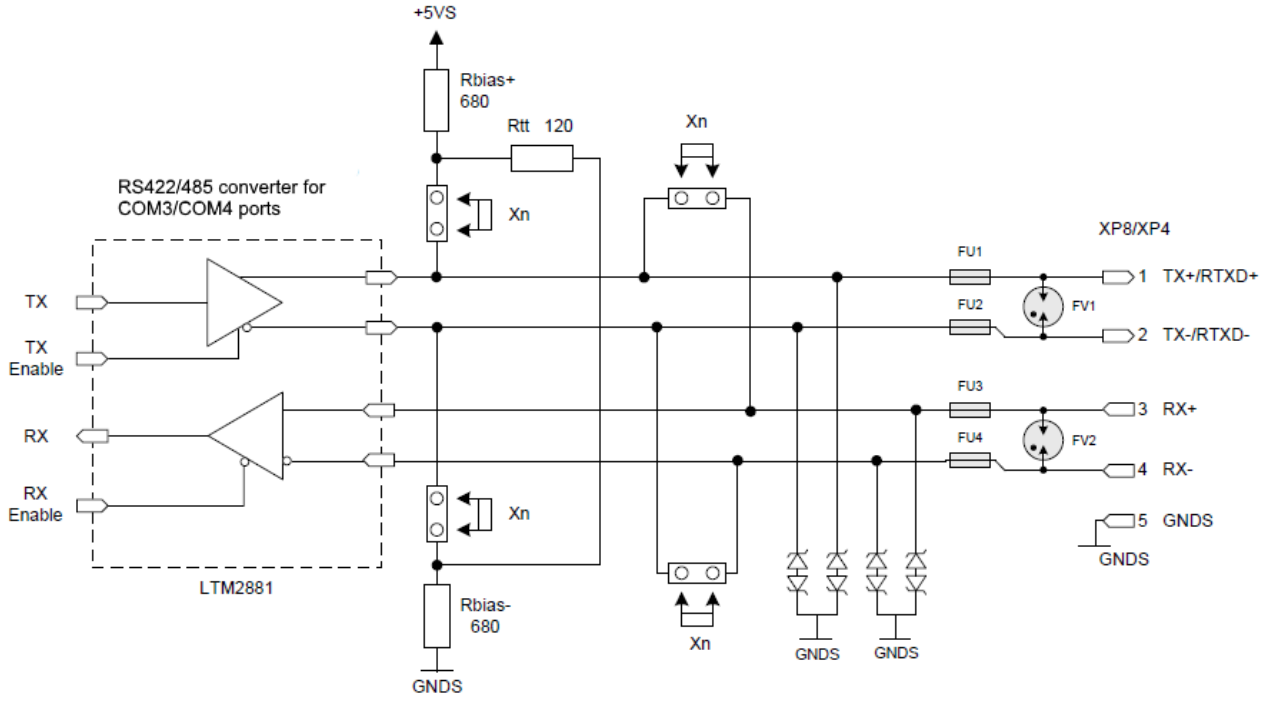



**Fig. 1-6 – Output stages of RS-422/485 ports of the device**

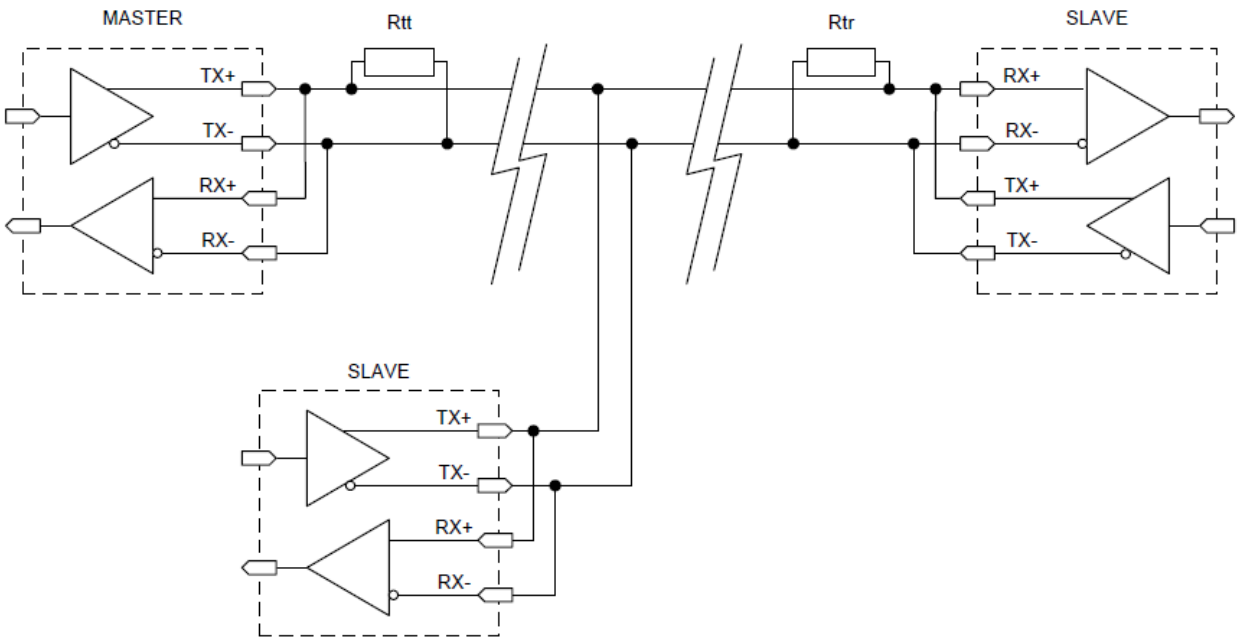



**Fig. 1-7 – Connecting modules via RS-485 interface**

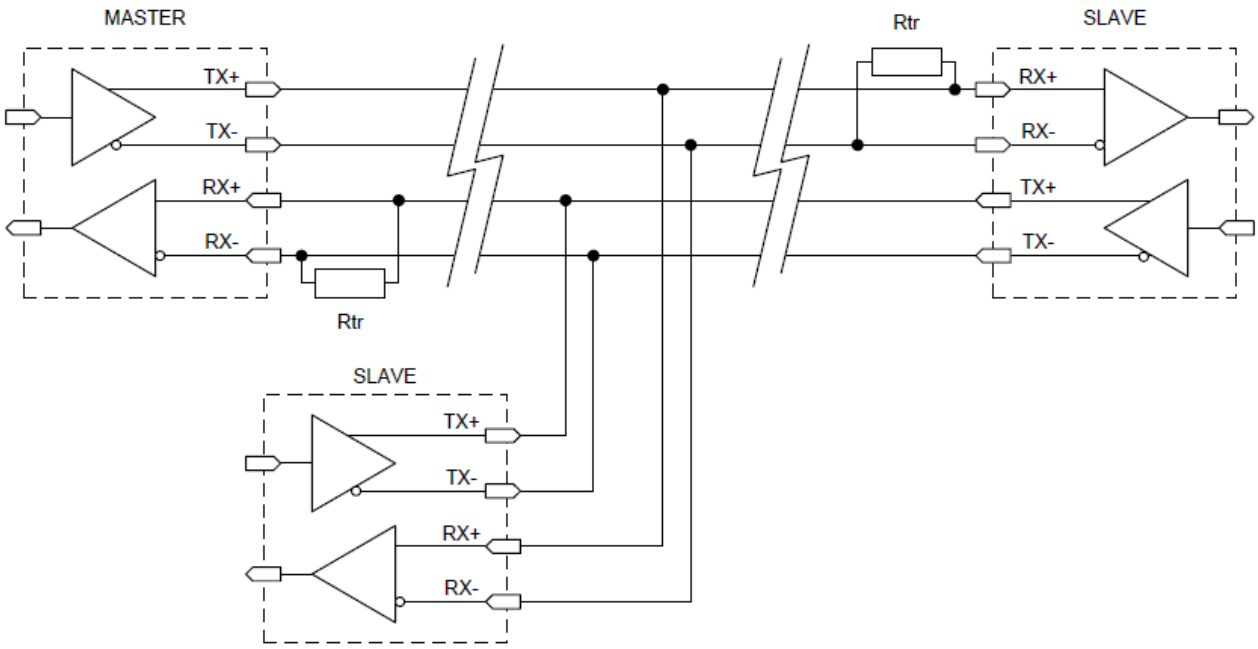



**Fig. 1-8 – Connecting modules via RS-422 interface**

The COM1 (XP13) and COM2 (XP14) ports operate in RS-232 mode. The maximum baud rate for COM1 and COM2 is 115.2 Kb/s. The ports are fully software compatible with the 16550 UART model.

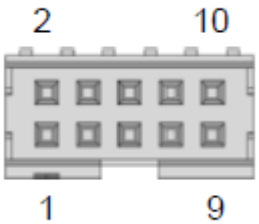
Each port is routed to a vertical two-row 10-pin connector of the IDC2-10 type with a pitch of 2 mm (98414-G06-10LF, FCI). As a mating part, it is necessary to use a 10-pin socket 89947-710LF (FCI) for a ribbon cable with a pitch of 1 mm. A ready-made cable (IDC2-10 - DB9M) ACS00023-04 is available for ordering, cable length: 17 cm.

The both ports can be used for console I/O and file downloads. A null-modem cable is required to communicate with a PC used as a hyperterminal.

By default, the console port is COM1 (port settings in the terminal program PuTTY, Hyperterminal: rate - 115200 bit/s, data bit - 8, stop bit - 1, no parity check).

**Table 1-7 – Purpose of the pins of RS-232 (XP13, XP14) ports**

| COM1, COM2: 98414-G06-10LF (FCI) | | | |
|---|---|---|---|
| **Pin#** | **Function** | **Pin#** | **Function** |
| 1 | DCD | 2 | DSR |
| 3 | RXD# | 4 | RTS |
| 5 | TXD# | 6 | CTS |
| 7 | DTR | 8 | RI |
| 9 | GND | 10 | +5V_EXTR |




The base addresses and interrupts for serial ports are set in BIOS Setup. By default, the following base addresses/interrupts are set for serial ports in BIOS Setup:

COM1 (RS-232): 0x3F8 / IRQ4, [115200 8,n,1] [10]
COM2 (RS-232): 0x2F8 / IRQ3, [115200 8,n,1]
COM3 (RS-485): 0x3E8 / IRQ4, [115200 8,n,1]
COM4 (RS-485): 0x2E8 / IRQ3, [115200 8,n,1]
The baud rate for serial ports can be set in the BIOS Setup. The exchange rate is determined by the value of the CPU frequency divider register. The value of the divider and the data exchange rate is calculated using the following formulas:

**DIV = F / (16 • BR), BR=F / (DIV • ( SM + FD))**

- F – internal generator frequency, MHz (F = 1.8432 / 24 / 48);
- DIV (divider) – divider value (for F = 1.8432, 24 and 48 MHz – minimum value DIV = 1);
- BR (baud rate) - required exchange rate, bit/sec;
- SM (sampling mode) – Base divider value (SM = 16).

**Attention!**
The receiver allows the exchange rate value downward bias by 3.0% and upward bias by 2.5%.

**Table 1-8 – Frequency divider values for serial ports**

| Exchange rate, bit/s | F=1.8432 MHz | | F=24 MHz | | F=48 MHz | |
|---|---|---|---|---|---|---|
| | Divider | Error, % | Divider | Error, % | Divider | Error, % |
| 300 | 384 | – | 5000 | – | 10000 | – |
| 600 | 192 | – | 2500 | – | 5000 | – |
| 1200 | 96 | – | 1250 | – | 2500 | – |
| 2400 | 48 | – | 625 | – | 1250 | – |
| 3600 | 32 | – | 417 | – | 625 | – |
| 4800 | 24 | – | 312 | – | 625 | – |
| 7200 | 16 | – | 208 | – | 417 | – |
| 9600 | 12 | – | 156 | – | 312 | – |
| 19200 | 6 | – | 78 | – | 156 | – |
| 38400 | 3 | – | 39 | – | 78 | – |
| 57600 | 2 | – | 26 | – | 52 | – |
| 115200 | 1 | – | 13 | – | 26 | – |
| 200000 | – | – | – | – | – | – |
| 230400 | – | – | – | – | 13 | – |
| 250000 | – | – | 6 | – | 12 | – |
| 300000 | – | – | 5 | – | 10 | – |
| 460800 | – | – | – | – | – | – |

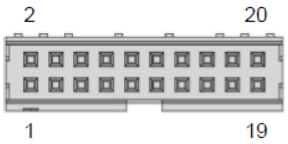
[10] Baud rate 115.2 Kbaud, 8 bits, no parity check, 1 stop bit

### 1.9.9 USB1-4 ports

The Device Controller has 4 x USB Host ports supporting USB 1.1 and USB 2.0 specifications. The operating mode of the interfaces is set in the BIOS Setup menu.
Each of the channels has an independent power supply control circuit and power protection (+5 V, 500 mA).
All 4 x channels are routed to one XP19 connector: IDC2-20, 20-pin, two-row male connector with a pitch of 2 mm, 98414-G06-20LF (FCI).
A 20-pin socket 89947-720LF (FCI) to a flat ribbon cable with a pitch of 1 mm or a 20-pin socket 10073599-020LF (FCI) with pins 77138-101LF (FCI) should be used as a mating part.

**Table 1-9 – Purpose of pins of the USB (XP19) ports**

| USB1-4: 98414-G06-20LF (FCI) | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | +5 V @ 0.5A | 2 | +5 V @ 0.5A |
| 3 | D- | 4 | D- |
| 5 | D+ | 6 | D+ |
| 7 | GND | 8 | GND |
| 9 | - | 10 | - |
| 11 | +5 V @ 0.5A | 12 | +5 V @ 0.5A |
| 13 | D- | 14 | D- |
| 15 | D+ | 16 | D+ |
| 17 | GND | 18 | GND |
| 19 | - | 20 | - |




### 1.9.10 LAN1 and LAN2 ports

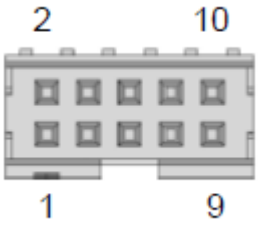
The device contains 2 x LAN ports.
LAN1 port supports 10/100/1000 Mbit operation modes and is based on Intel WGI210IT controller.
The LAN2 port supports 10/100 Mbit operation modes and is implemented on the basis of a controller built into the Vortex86DX3 microprocessor.
The ports provide galvanic isolation up to 500 V (each port is individually isolated from the system).
Each port is routed to a two-row 10-pin connector with a pitch of 2 mm, 98414-G06-10LF (FCI). 10-pin socket 89947-710LF (FCI) for a flat ribbon cable with a pitch of 1 mm or a 10-pin socket 10073599-010LF (FCI) with pins 77138-101LF (FCI).

**Table 1-10 – Purpose of pins of the LAN1 (XP10) port**

| LAN1: 98414-G06-10LF (FCI) | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | MD0+ / TX+ | 2 | MD0- / TX- |
| 3 | MD1+ / RX+ | 4 | MD1- / RX- |
| 5 | MD2+ | 6 | MD2- |
| 7 | MD3+ | 8 | MD3- |
| 9 | - | 10 | - |


**Table 1-11 – Purpose of pins of the LAN2 (XP15) port**

| LAN2: 98414-G06-10LF (FCI) | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | TX+ | 2 | TX- |
| 3 | RX+ | 4 | RX- |
| 5 | - | 6 | - |
| 7 | - | 8 | - |
| 9 | - | 10 | - |




### 1.9.11 Ports of the digital I/O ports UNIO (DIO)

The device includes 2 x universal ports of digital input-output IO_A [23: 0] (XP16) and IO_B [23: 0] (XP18), which are compatible by output pins and control with the UNIO96-5 module. The ports are implemented on the XC6SLX16-2CSG324I field programmable gate array (FPGA) and are designed for input/output of 48 logic signals.

Overvoltage protection for 5.6 V is installed on each line (based on TVS diodes RClamp0524, designed to protect against pulse interference). The transmission direction is controlled by groups of channels or per channel, depending on the loaded configuration. The buffer elements based on the SN74LVC1T45DCKTE4 microchips with control of the transmission direction are used to provide power supply voltage output up to +5 V; to read the status of channels during output operation, groups of buffer elements based on SN74LVC2G125YZPR are used with the arrangement of switching channel groups (each group consists of 8 channels).

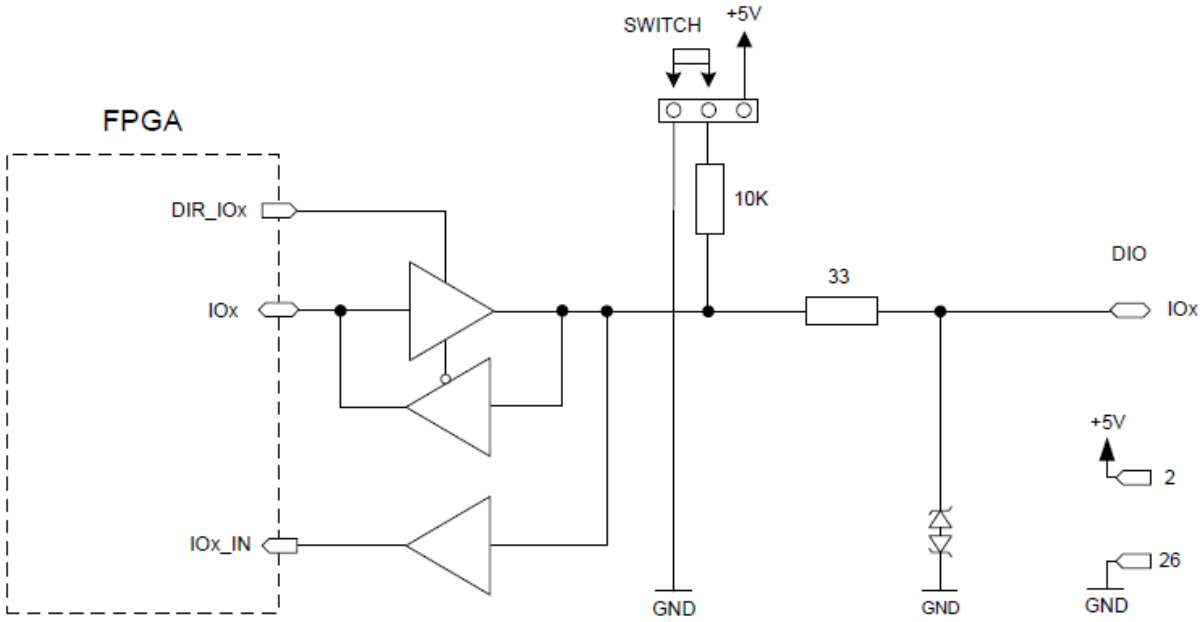



**Fig. 1-9 - Diagram of one channel of the port of digital input-output**

The function (scheme) of the port can be changed directly in the system (ISP technology, in-system-programming) without turning off the power.

The port channels can be used to control RTU modules with galvanic isolation, pulse counting, frequency measurement and generation, time diagrams, etc.

**Attention!**

When connecting external devices to the digital I/O port, using the common wire (pin # 26 of the IDC2-26 connector) is mandatory.

**Attention!**

It is recommended to use an additional connection to the high (+5 V) or low (GND) level of the digital I/O port's lines directly in the devices connected to these ports (recommended resistance rated value is 10 kOhm). In this case, the selected levels (+5 V or GND) on the CPU module and the connected external board should match.

When delivered, the device has the installed firmware n00. The change of the port configuration version is made by software and is carried out by recording to the corresponding control register, see the **Register for selecting the active firmware of the digital I/O port**.

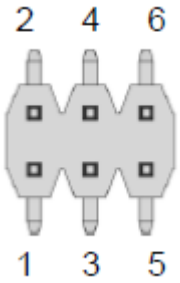
**Attention!**

Firmware of the FPGA chip, which is used as the basis for the digital I/O port, is incompatible with the firmware of other modules. Using the firmware of another module may cause device malfunctions.

Connecting 10 kOhm resistors for binding the port lines to the "+5V" power supply voltage or to the "GND" ground is carried out with the X20 switch.

**Table 1-12 – Binding levels of UNIO ports lines (X20)**

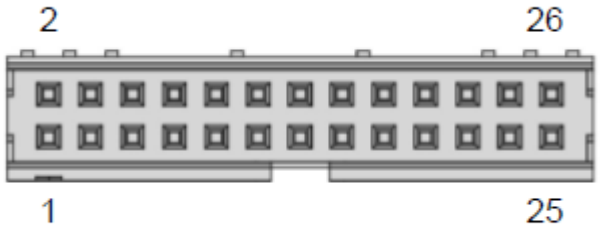
| Jumper | Description |
|---|---|
| **X20[1-3]** [11] | **Binding the lines IO_A[23:0] via resistors 10 kOhm to the power supply voltage +5V** |
| X20[3-5] | Binding the lines IO_A[23:0] via resistors 10 kOm to the ground (GND) |
| **X20[2-4]** | **Binding the lines IO_B[23:0] via resistors 10 kOhm to the power supply voltage +5V** |
| X21[4-6] | Binding the lines IO_B[23:0] via resistors 10 kOhm to the ground (GND) |




Each port is routed to a two-row 26-pin connector with a pitch of 2 mm, 98424-G52-26LF (FCI). A 26-pin socket 89947-726LF (FCI) for a flat ribbon cable with a pitch of 1 mm or a 20-pin socket 10073599-026LF (FCI) with the pins 77138-101LF (FCI) should be used as a mating part.

**Table 1-13 – Purpose of the pins of DIO (UNIO IO_A[23:0], XP16) connector**

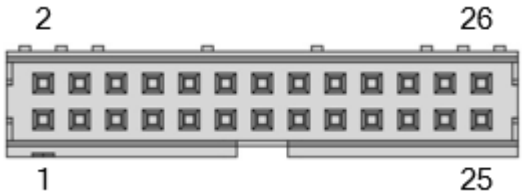
| UNIO IO_A: 98424-G52-26LF (FCI) | | | | |
|---|---|---|---|---|
| **Pin #** | **Function** | | **Pin #** | **Function** |
| 1 | IO_A(12) | | 2 | +5V_EXTA |
| 3 | IO_A(13) | | 4 | IO_A(10) |
| 5 | IO_A(14) | | 6 | IO_A(11) |
| 7 | IO_A(15) | | 8 | IO_A(9) |
| 9 | IO_A(23) | | 10 | IO_A(8) |
| 11 | IO_A(21) | | 12 | IO_A(22) |
| 13 | IO_A(16) | | 14 | IO_A(20) |
| 15 | IO_A(18) | | 16 | IO_A(17) |
| 17 | IO_A(19) | | 18 | IO_A(7) |
| 19 | IO_A(0) | | 20 | IO_A(6) |
| 21 | IO_A(1) | | 22 | IO_A(5) |
| 23 | IO_A(2) | | 24 | IO_A(4) |
| 25 | IO_A(3) | | 26 | GND |




[11] The default switch positions are highlighted in bold

**Table 1-14 – Purpose of pins of the DIO (UNIO IO_B[23:0] , XP18) connector**

| UNIO IO_B: 98424-G52-26LF (FCI) | | | | |
|---|---|---|---|---|
| **Pin #** | **Function** | | **Pin #** | **Function** |
| 1 | IO_B(12) | | 2 | +5V_EXTB |
| 3 | IO_B(13) | | 4 | IO_B(10) |
| 5 | IO_B(14) | | 6 | IO_B(11) |
| 7 | IO_B(15) | | 8 | IO_B(9) |
| 9 | IO_B(23) | | 10 | IO_B(8) |
| 11 | IO_B(21) | | 12 | IO_B(22) |
| 13 | IO_B(16) | | 14 | IO_B(20) |
| 15 | IO_B(18) | | 16 | IO_B(17) |
| 17 | IO_B(19) | | 18 | IO_B(7) |
| 19 | IO_B(0) | | 20 | IO_B(6) |
| 21 | IO_B(1) | | 22 | IO_B(5) |
| 23 | IO_B(2) | | 24 | IO_B(4) |
| 25 | IO_B(3) | | 26 | GND |




### 1.9.12 RTC, SPI FRAM, lithium battery

The device is equipped with an AT-compatible real-time clock with an installed CR2032 lithium battery with a capacity of 200 mAh. The expected / typical battery service life is 7 years[12]. However, the battery service life is highly dependent on operating temperature and how long the system is off.

**Important note!**

If the device is turned off for a long time, it is recommended to synchronize the real-time clock with the exact time readings.

### 1.9.13 VGA and LVDS ports

The video subsystem of the device is based on a video processor integrated into the Vortex86DX3 microprocessor. The video controller with a 2D accelerator function has the following technical specifications and capabilities:

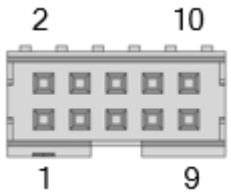
- video memory size – to be allocated from the system memory;
- possibility to connect RGB (VGA) monitors with max. resolution of 1920 x 1080 (60 Hz, 32 bit);
- possibility to connect LVDS panels with max. resolution of 1920 x 1080 (60 Hz, 32 bit)[13].

[12] Under normal operating conditions.
[13] In the event of connecting both VGA monitor and LVDS panel, their maximum resolution is 1280 x 1024.

The VGA (XP2) port is routed to a two-row 10-pin connector with a pitch of 2 mm, 98414-G06-10LF (FCI). As a mating part, you should use a 10-pin 89947-710LF (FCI) socket for a ribbon cable with a pitch of 1 mm or a 10-pin 10073599-010LF (FCI) socket with 77138-101LF (FCI) pins. A ready-made cable (IDC2-10 - DB15F) ACS00027-02 is available for order, with a length of 17 cm.

**Table 1-15 – Purpose of the VGA port pins (XP2)**

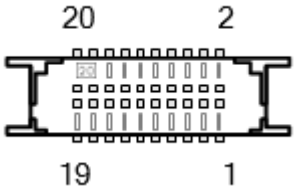
| VGA: 98414-G06-10LF (FCI) | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | RED | 2 | GND |
| 3 | GREEN | 4 | GND |
| 5 | BLUE | 6 | GND |
| 7 | HSYNC | 8 | VSYNC |
| 9 | DDC_SCL | 10 | DDC_SDA |




The LVDS (XP3) port is routed to the two-row 20-pin connector with a pitch of 1.25 mm, DF13EA-20DP-1.25V (Hirose). As a mating part, you should use the 20-pin DF13-20DS-1.25C (Hirose) socket with DF13-2630SCF (Hirose) pins.

**Table 1-16 – Purpose of LVDS (XP3) port pins**

| LVDS: DF13EA-20DP-1.25V (Hirose) | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | VCC (+3.3 V) | 2 | VCC (+3.3 V) |
| 3 | DATA0+ | 4 | DATA0- |
| 5 | GND | 6 | GND |
| 7 | DATA1+ | 8 | DATA1- |
| 9 | GND | 10 | GND |
| 11 | DATA2+ | 12 | DATA2- |
| 13 | GND | 14 | GND |
| 15 | DATA3+ | 16 | DATA3- |
| 17 | GND | 18 | GND |
| 19 | CLK+ | 20 | CLK- |




## 1.9.14 Audio ports

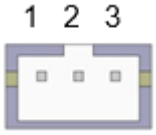
The device's audio ports are based on the HDA audio controller integrated into the Vortex86DX3 microprocessor and the CS4207 24-bit audio codec (Cirrus Logic).
The codec is compatible with the SBPRO™ model.
The set of Audio ports includes: an audio output (STEREO) and a MIC (STEREO).
The DAC1 (Headphone) output of the CS4207 audio codec is used as the audio output.
HDPH audio output (XP17) and MIC input (XP6) are routed to 3-pin B 3B-PH-K-S (JST) connectors.

To manufacture the cable, it is recommended to use a PHR-3, JST socket with SPH-002T-P0.5S pins, JST (available for ordering as a set ACS00031-04).

**Table 1-17 – Purpose of Audio ports pins (XP17, XP6)**

| Audio out: B 3B-PH-K-S (JST) | |
|---|---|
| **Pin #** | **Function** |
| 1 | HDPH Left |
| 2 | GND |
| 3 | HDPH Right |

| Audio MIC: B 3B-PH-K-S (JST) | |
|---|---|
| **Pin #** | **Function** |
| 1 | MIC Left |
| 2 | GND |
| 3 | MIC Right |




## 1.9.15 PC/104 expansion bus (ISA 8/16-bit)

The PC/104 (XS1) connector is designed for installing expansion modules of the PC/104 or PC/104-plus format to the module. It is allowed to install no more than 4x PC/104 expansion modules.

ISA bus operating modes: 8/16 bit, 8.3/16.6 MHz. Master mode is not supported.

The device is equipped with an AMP 1375795-5 connector (pins B10 and C19 are missing) and an AMP 1445251-1 organizer.

**Table 1-18 – Purpose of XS1 connector pins: PC/104 (ISA 8/16-bit) rows A, B**

| ISA: Connector of PC/104 expansion bus (2x32 + 2x20 pins), rows A, B | | | | | |
|---|---|---|---|---|---|
| **Pin #** | **Purpose** | **Configuration** | **Pin #** | **Purpose** | **Configuration** |
| A1 | IOCHK# | Input | B1 | GND | Power supply |
| A2 | SD7 | Input/Output | B2 | RESET | Output |
| A3 | SD6 | Input/Output | B3 | +5V | Input |
| A4 | SD5 | Input/Output | B4 | IRQ9 | Input |
| A5 | SD4 | Input/Output | B5 | - | |
| A6 | SD3 | Input/Output | B6 | DRQ2 | Input |
| A7 | SD2 | Input/Output | B7 | -12V | Power supply |
| A8 | SD1 | Input/Output | B8 | 0WS# | Input |
| A9 | SD0 | Input/Output | B9 | +12V | Power supply |
| A10 | IOCHRDY | Input | B10 | GND | Power supply |
| A11 | AEN | Output | B11 | SMEMW# | Output |
| A12 | SA19 | Output | B12 | SMEMR# | Output |
| A13 | SA18 | Output | B13 | IOW# | Output |
| A14 | SA17 | Output | B14 | IOR# | Output |
| A15 | SA16 | Output | B15 | DACK3# | Output |
| A16 | SA15 | Output | B16 | DRQ3 | Input |
| A17 | SA14 | Output | B17 | DACK1# | Output |
| A18 | SA13 | Output | B18 | DRQ1 | Input |

| | | | | | | |
|---|---|---|---|---|---|---|
| A19 | SA12 | Output | | B19 | DACK0# | Output |
| A20 | SA11 | Output | | B20 | BCLK | Output |
| A21 | SA10 | Output | | B21 | IRQ7 | Input |
| A22 | SA9 | Output | | B22 | IRQ6 | Input |
| A23 | SA8 | Output | | B23 | IRQ5 | Input |
| **ISA: Connector of PC/104 bus expansion (2x32 + 2x20 pins), rows A, B** | | | | | | |
| **Pin #** | **Purpose** | **Configuration** | | **Pin #** | **Purpose** | **Configuration** |
| A24 | SA7 | Output | | B24 | IRQ4 | Input |
| A25 | SA6 | Output | | B25 | IRQ3 | Input |
| A26 | SA5 | Output | | B26 | DACK2# | Output |
| A27 | SA4 | Output | | B27 | TC | Output |
| A28 | SA3 | Output | | B28 | BALE# | Output |
| A29 | SA2 | Output | | B29 | +5V | Power supply |
| A30 | SA1 | Output | | B30 | OSC | Output |
| A31 | SA0 | Output | | B31 | GND | Power supply |
| A32 | GND | Power supply | | B32 | GND | Power supply |

**Table 1-19 - Purpose of XS1 connector pins: PC/104 (ISA 8/16-bit) rows C, D**

| **ISA: Connector of PC/104 bus expansion (2x32 + 2x20 pins), rows C, D** | | | | | | |
|---|---|---|---|---|---|---|
| **Pin #** | **Purpose** | **Configuration** | | **Pin #** | **Purpose** | **Configuration** |
| C0 | GND | Power supply | | D0 | GND | Power supply |
| C1 | SBHE# | Output | | D1 | MEMCS16# | Input |
| C2 | LA23 | Output | | D2 | IOCS16# | Input |
| C3 | LA22 | Output | | D3 | IRQ10 | Input |
| C4 | LA21 | Output | | D4 | IRQ11 | Input |
| C5 | LA20 | Output | | D5 | IRQ12 | Input |
| C6 | LA19 | Output | | D6 | IRQ15 | Input |
| C7 | LA18 | Output | | D7 | IRQ14 | Input |
| C8 | LA17 | Output | | D8 | DACK0# | Output |
| C9 | MEMR# | Output | | D9 | DRQ0 | Input |
| C10 | MEMW# | Output | | D10 | DACK5# | Output |
| C11 | SD8 | Input/Output | | D11 | DRQ5 | Input |
| C12 | SD9 | Input/Output | | D12 | DACK6# | Output |
| C13 | SD10 | Input/Output | | D13 | DRQ6 | Input |
| C14 | SD11 | Input/Output | | D14 | DACK7# | Output |
| C15 | SD12 | Input/Output | | D15 | DRQ7 | Input |
| C16 | SD13 | Input/Output | | D16 | +5V | Power supply |
| C17 | SD14 | Input/Output | | D17 | - | |
| C18 | SD15 | Input/Output | | D18 | GND | Power supply |
| C19 | - | | | D19 | GND | Power supply |

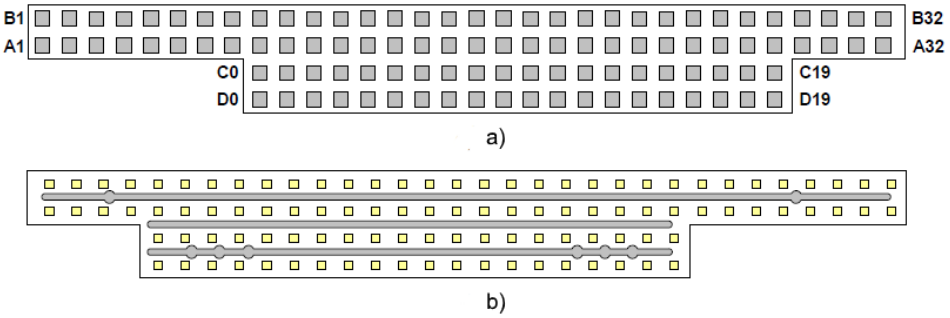



**Fig. 1-10 – Numbering pins of the XS1 connector a) top view of the device, b) bottom view of the device with the organizer installed into connector**

### 1.9.16 PC/104-plus expansion bus (PCI 32-bit)

The device can be connected to 3 x expansion boards of the PC/104-plus or PCI/104 format. The PCI104 bus (PCI, 32 bit, 33 MHz) supports up to 3 x PCI master devices.
The interface is routed to the XS2 connector (PCI/104 connector, 120 pins, 2 mm pitch).
The product has an AMP 1375799-1 connector and an AMP 1375801-1 organizer.

**Table 1-20 – Purpose of XS2 connector pins: PCI/104 (PCU 32-bit) rows A, B**

| PCI: Connector of PCI/104 bus expansion (4x30 pins), rows A, B | | | | | |
|---|---|---|---|---|---|
| **Pin #** | **Purpose** | **Configuration** | **Pin #** | **Purpose** | **Configuration** |
| A1 | GND | Power supply | B1 | - | - |
| A2 | VI/O | +3.3V (output) | B2 | AD2 | I/O |
| A3 | AD5 | I/O | B3 | GND | Power supply |
| A4 | C/BE0# | I/O | B4 | AD7 | I/O |
| A5 | GND | Power supply | B5 | AD9 | I/O |
| A6 | AD11 | I/O | B6 | VI/O | +3.3V (output) |
| A7 | AD14 | I/O | B7 | AD13 | I/O |
| A8 | - | - | B8 | C/BE1# | I/O |
| A9 | SERR# | PU (10K) | B9 | GND | Power supply |
| A10 | GND | Power supply | B10 | PERR# | PU (10K) |
| A11 | STOP# | I/O | B11 | - | - |
| A12 | - | - | B12 | TRDY# | I/O |
| A13 | FRAME# | I/O | B13 | GND | |
| A14 | GND | Power supply | B14 | AD16 | I/O |
| A15 | AD18 | I/O | B15 | - | - |
| A16 | AD21 | I/O | B16 | AD20 | I/O |
| A17 | - | - | B17 | AD23 | I/O |
| A18 | IDSEL0 | AD12 | B18 | GND | Power supply |
| A19 | AD24 | I/O | B19 | C/BE3# | I/O |
| A20 | GND | Power supply | B20 | AD26 | I/O |
| A21 | AD29 | I/O | B21 | +5V | Power supply |
| A22 | +5V | Power supply | B22 | AD30 | I/O |
| **PCI: Connector for expansion of PCI/104 bus (4x30 pins), rows A, B** | | | | | |
| **Pin #** | **Purpose** | **Configuration** | **Pin #** | **Purpose** | **Configuration** |

| | | | | | |
|---|---|---|---|---|---|
| A23 | REQ0# | Input | B23 | GND | Power supply |
| A24 | GND | Power supply | B24 | REQ2# | Input |
| A25 | GNT1 | Output | B25 | VI/O | +3.3V (output) |
| A26 | +5V | Power supply | B26 | CLK0 | Output |
| A27 | CLK2 | Output | B27 | +5V | Input |
| A28 | GND | Power supply | B28 | INTD# | Input |
| A29 | +12V | - | B29 | INTA# | Input |
| A30 | - | - | B30 | REQ3# | Input |

**Table 1-21 – Purpose of XS2 connector pins: PCI/104 (PCU 32-bit) rows C, D**

| PCI: Connector of PCI/104 bus expansion (4x30 pins), rows C, D | | | | | |
|---|---|---|---|---|---|
| **Pin #** | **Purpose** | **Configuration** | **Pin #** | **Purpose** | **Configuration** |
| C1 | +5V | Power supply | D1 | AD0 | I/O |
| C2 | AD1 | I/O | D2 | +5V | Power supply |
| C3 | AD4 | I/O | D3 | AD3 | I/O |
| C4 | GND | Power supply | D4 | AD6 | I/O |
| C5 | AD8 | I/O | D5 | GND | Power supply |
| C6 | AD10 | I/O | D6 | M66EN (GND) | |
| C7 | GND | Power supply | D7 | AD12 | I/O |
| C8 | AD15 | I/O | D8 | - | - |
| C9 | - | - | D9 | PAR | I/O |
| C10 | - | - | D10 | - | - |
| C11 | LOCK# | PU (10K) | D11 | GND | Power supply |
| C12 | GND | Power supply | D12 | DEVSEL# | I/O. |
| C13 | IRDY# | I/O | D13 | - | - |
| C14 | - | - | D14 | C/BE2# | I/O |
| C15 | AD17 | I/O | D15 | GND | Power supply |
| C16 | GND | Power supply | D16 | AD19 | I/O |
| C17 | AD22 | I/O | D17 | - | - |
| C18 | IDSEL1 | AD13 | D18 | IDSEL2 | AD14 |
| C19 | VI/O | +3.3V (output) | D19 | IDSEL3 | AD15 |
| C20 | AD25 | I/O | D20 | GND | Power supply |
| C21 | AD28 | I/O | D21 | AD27 | I/O |
| C22 | GND | Power supply | D22 | AD31 | I/O |
| C23 | REQ1# | Input | D23 | VI/O | +3.3V (output) |
| C24 | +5V | Power supply | D24 | GNT0# | Output |
| C25 | GNT2# | Output | D25 | GND | Power supply |
| C26 | GND | Power supply | D26 | CLK1 | Output |
| C27 | CLK3 | Output | D27 | GND | Power supply |
| C28 | +5V | Power supply | D28 | RST# | Output |
| C29 | INTB# | Input | D29 | INTC# | Input |
| C30 | GNT3# | Output | D30 | GND | Power supply |

The “Status” column indicates the direction of data transmission for the case when the device is the bus master.

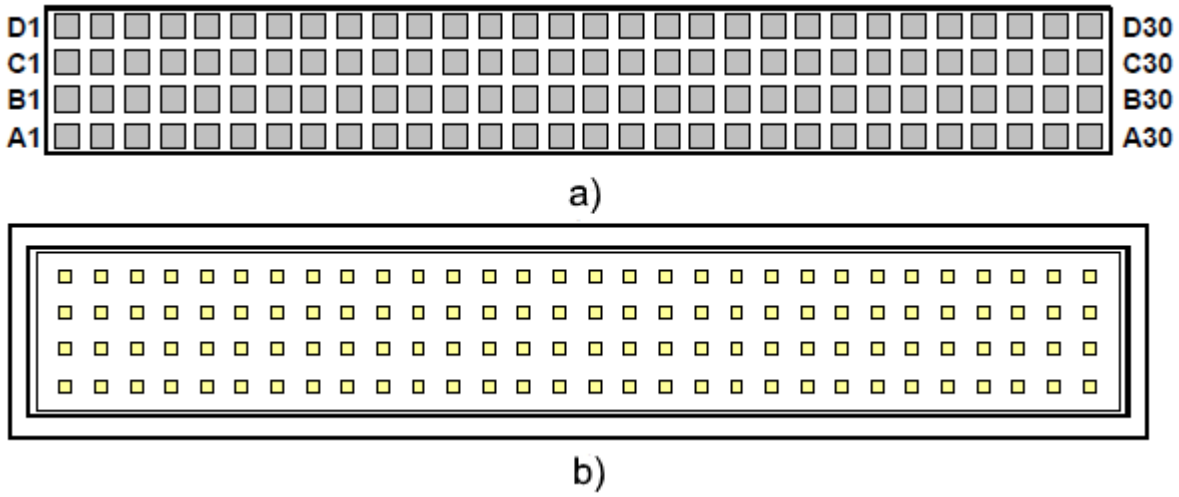



**Fig. 1-11 – Numbering XS2 connector pins a) top view of the device, b) bottom view of the device with the organizer installed into connector**

## 1.9.17 Diagnostic LEDs and XP9 connector

All the LEDs are located on the top side of the device. The purpose of the LEDs is shown in Table 1-23.

**Table 1-22 – Purpose of device LEDs**

| LED | Function |
|---|---|
| HL1 (green) | Indication of data exchange with integrated drive and Compact Flash |
| HL2 (red) * | User LED (port 0x281h) |
| HL3 (green) * | User LED (port 0x281h) |
| HL4 (red) | Indicating the state of input power supply voltage +5V |
| HL5 (yellow) | Indicating the state of secondary power supply voltages of the module |
| XS2 / HL6 | The connector makes it possible to connect the network controller activity LED (green) |
| XS2 / HL7 | The connector allows the connection of the network controller operating mode LED (yellow) |
| *The HL2 and HL3 LEDs are controlled via the FPGA IC register (I/O space, ISA bus) | |

XP9 male connector is provided for relocating indication from the board.

**Table 1-23 – Purpose of pins of XP9 indication connector**

| Connector pins | Function |
|---|---|
| XP9[1-2] | LAN1 activity (Fast Ethernet) |
| XP9[3-4] | LAN2 activity (GbE) |
| XP9[5-6] | Timeout for the watchdog timer integrated into the power supervisor |

| | |
|---|---|
| XP9[7-8] | Indication of reset |
| XP9[9-10] | Indication of exchange between SSD and Compact Flash |
| XP9[11-12] | User LED (port 0x281h, red) |
| XP9[13-14] | User LED (port 0x281h, green) |
| * Even pins correspond to the cathode “-“, odd pins – to the anode “+” for connection of LED. | |

### 1.9.18 Sensors

The device is equipped with the LM92CIM (National Semiconductor) temperature sensor, which makes it possible to measure the temperature on the product surface with a resolution of 12 bits (+ sign) within the range from -55 to +125 °C. The sensor is located on the board in the area of the Vortex86DX3 processor. The temperature sensor is connected to the I2C0 bus of the processor. Address on the I2C bus: write - 0x90, read - 0x91 (the most significant seven bits of the address: b'1001000 + the least significant bit: b'0 for writing, b'1 for reading). The LM92CIM sensor allows you to monitor the temperature of the controller in the processor area.
The measurement error is not standardized, the typical error is determined by the characteristics declared by the sensor manufacturers.
To use the sensor as a measuring one, such a sensor should be calibrated (the system that stores calibration factors can be arranged on the basis of FRAM non-volatile memory).
An example of sensor programming is given in subparagraph **4.9 “Working with I2C Devices”**.

### 1.9.19 Measuring the device’s power supply voltages

To measure the secondary supply voltages of the device, an analog-to-digital converter integrated into the Vortex86DX3 is used. The measurement error is not standardized, the typical error is determined by the characteristics declared by the Vortex86DX3 processor manufacturer.
Table 1-25 shows how the lines of the ADC_AUX [7: 0] port match the secondary supply voltages of the device.

**Table 1-24 – Purpose of ADC channels ADC_AUX [7:0]**

| ADC_AUX port | Power supply voltage |
|---|---|
| ADC_AUX [0] | +5 V (main supply voltage) at the input a voltage divider ½ is installed (implemented on 2 x 1 kOhm resistors, 1%) |
| ADC_AUX [1] | +3.3 V (supply voltage used to start the secondary converters: 0.95, 1.2, 1.35, 1.8, 3.3 V) |
| ADC_AUX [2] | +0.95 V (CPU core voltage) |
| ADC_AUX [3] | +1.2 V (PCIe, SATA supply voltage) |
| ADC_AUX [4] | +1.35 V (DDR3 supply voltage) |
| ADC_AUX [5] | +1.8 V (GPU, PLL, ANALOG supply voltage) |
| ADC_AUX [6] | +3.3 V (supply voltage of GPIO, ISA ports and other I/O ports) |
| ADC_AUX [7] | 0 V |

### 1.9.20 GPIO port

The GPIO port is implemented on the MCP23009-E/MG controller connected to the I2C0 bus of the Vortex86DX3 processor. Address on the I2C0 bus: write - 0x40, read - 0x41 (the most significant seven bits of the address: b'0100000 + the least significant bit: b'0 for writing, b'1 for reading).

The port lines are tolerant towards the 5V voltage level. Each port line can be configured as input or output. The par value of the pull-up resistor for each line is 10 kOhm.

By writing to the registers of the MCP23009-E/MG IC, you can configure each line separately as an input or output, you can set the output type: “open collector” or “push-pull”. After power is on, all the port lines are configured as inputs, for outputs the type is “push-pull”. Examples of GPIO port programming can be found in subparagraph **4.9 “Working with I2C Devices”**.

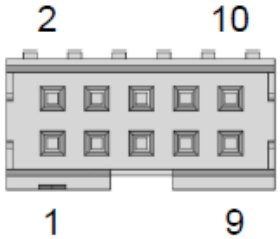
To ensure the accuracy of the state of the GPIO port lines after power-up, it is recommended to use a dedicated supply voltage line (+ 5VEXT, output No. 9 of XP10 connector) to power the logic connected to the GPIO port, and also use a high (+5 V) or low (GND ) levels directly in devices connected to the port. When connecting external devices, the common wire connection (XP10 connector, output 10) is mandatory.

The GPIO port is routed to the XP1 connector (98414-G06-10LF, FCI).

To make a cable for the GPIO port, it is recommended to use a 10-pin socket per ribbon cable with a pitch of 1 mm: 89947-710LF (FCI).

**Table 1-25 – Purpose of GPIO (XP1) port pins**

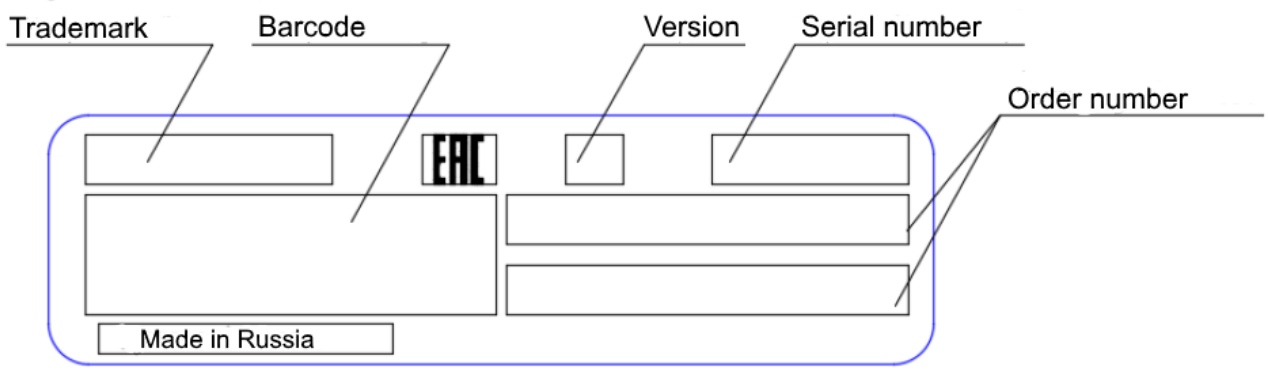
| GPIO: 98414-G06-10LF (FCI) | | | |
|---|---|---|---|
| **Pin #** | **Function** | **Pin #** | **Function** |
| 1 | GPIO (0) | 2 | GPIO (1) |
| 3 | GPIO (2) | 4 | GPIO (3) |
| 5 | GPIO (4) | 6 | GPIO (5) |
| 7 | GPIO (6) | 8 | GPIO (7) |
| 9 | +5VEXT | 10 | GND |



## 1.10 Labelling

The consumer container used for storing the product is labelled by means of an individual identifier (sticker).
The consumer container sticker contains the following information:

Product designation in accordance with the ordering information;
Device version;
Serial number of the device;
Manufacturer's trademark;
Barcode.




** The location of the fields with product-related information may differ slightly from the one shown in this figure.*

## 1.11 Packaging

The product is packed in individual antistatic packaging (bag) with the following dimensions: 152 x 254 mm and placed in a separate consumer container (cardboard box). The box dimensions are: 155 x 140 x 45 mm.

**Note**

Retain the original packaging for storing products in the future or to transport in case of a warranty claim. If it is necessary to transport or store the board, pack it the same way as it was packed at the time of receipt.

# 2 Intended Use

## 2.1 Operating limitations

The safety requirements listed at the beginning of this User Manual should be strictly observed. Installation and removal of the device, connection to the slots should be carried out only with the power supply turned off. Always observe the proper orientation of the device connectors during installation.

Operation of the device where the supply voltage that does not correspond to the one specified in **subparagraph 1.4 Power supply** is not allowed.

Using the device in harsh operating conditions without additional measures for fastening the Compact Flash in the connector (see subparagraph 1.9.7) is not allowed.

Operation of the device when exposed to external factors that do not correspond to the ones specified in subparagraph 1.2 Technical Specifications is not allowed.

## 2.2 Distribution of hardware interrupts

**Table 2-1 – Addresses of hardware interrupts**

| # | Default sources | Alternative sources |
|---|---|---|
| NMI | | |
| IRQ0 | *Reserved (system timer)* | |
| IRQ1 | | |
| IRQ2 | | |
| IRQ3 | | |
| IRQ4 | | |
| IRQ5 | | |
| IRQ6 | | |
| IRQ7 | | |
| IRQ8 | RTC (Real Time Clock) | |
| IRQ9 | | |
| IRQ10 | | |
| IRQ11 | | |
| IRQ12 | | |
| IRQ13 | *Reserved (CPU support)* | |
| IRQ14 | | |
| IRQ15 | | |

## 2.3 DMA channels

**Table 2-2 – DMA channels of the device**

| # | Source |
|---|---|
| DRQ0 | – |
| DRQ1 | – |
| DRQ2 | – |
| DRQ3 | – |
| DRQ5 | – |
| DRQ6 | – |
| DRQ7 | – |

## 2.4 I/O address space

**Table 2-3 – Distribution of I/O address space**

| Address | Function | Notes |
|---|---|---|
| – | – | – |
| 0x280h – 0x28Fh | UNIO, FBUS | Internal control registers (FPGA XC6S), FBUS registers |
| – | – | – |
| 0x290h – 0x29Fh | UNIO IO_A | UNIO digital I/O port (FPGA XC6S) |
| 0x2A0h – 0x2AFh | UNIO IO_B | UNIO digital I/O port (FPGA XC6S) |
| – | – | – |
| 0x2E8h – 0x2EFh | COM4 | Serial port COM4, RS-485 (Serial Port 2, Vortex86DX3) |
| 0x2F8h – 0x2FFh | COM2 | Serial port COM2, RS-232 (Serial Port 6, Vortex86DX3) |
| – | – | – |
| – | – | – |
| 0x3E8h – 0x3EFh | COM3 | Serial port COM3, RS-485 (Serial Port 1, Vortex86DX3) |
| 0x3F8h – 0x3FFh | COM1 | Serial port COM1, RS-232 (Serial Port 5, Vortex86DX3) |
| – | – | – |
| – | – | – |

## 2.5 Memory address space

**Table 2-4 – Addresses of memory devices**

| Address | Device | Notes |
|---|---|---|
| 00000 – 9FFFFh | DOS | DOS Area 640 Kbyte |
| A0000 – BFFFFh | VGA | Video memory area 128 Kbyte |
| C0000 – C7FFFh | VGA BIOS | VGA BIOS 32 Kbyte |
| C8000 – DFFFFh | – | – |
| E0000 – EFFFFh | System BIOS | Extended System BIOS area 64 Kbyte (16 Kbyte x 4) |
| F0000 – FFFFFh | System BIOS | System BIOS area 64 Kbyte |

| Address | Device | Notes |
|---|---|---|
| 10 0000 – MEMORY TOP * | DRAM | DDR3 SDRAM |
| MEMORY TOP * – FFE0 0000 | PCI | PCI |
| FFE0 0000 – FFFF FFFFh | High BIOS | High BIOS Area 2 Mbyte (mapped to PCI) |

* Installed DDR3 SDRAM – 2 GB.

## 2.6 Using the processor's GPIO ports

The Vortex86DX3 microchip has 10 x GPIO (General Purpose Input Output) I/O ports, available to the user through the internal registers of the microchip. Each port has 8 x I/O lines, each of which can be configured as input or output by programming the registers of the corresponding port.
For working with GPIO ports, 2 x 8-bit registers are used per port - a data register and a direction register. Each bit of the data register is mapped to the corresponding circuit on the board: bit 0 corresponds to the port 0 line (GPIO_Px0), bit 7 corresponds to the port 7 line (GPIO_Px7), etc. Each bit of the direction register is mapped to the corresponding circuit on the board: bit 0 corresponds to the port 0 line (GPIO_Px0), bit 7 corresponds to the port 7 line (GPIO_Px7), etc.

**Table 2-5 – GPIO control registers**

| | GPIO_P2 | GPIO_P7 | GPIO_P8 | GPIO_P9 | Description |
|---|---|---|---|---|---|
| Data register | 0x7Ah | 0x179h | 0x17Ah | 0x17Bh | |
| Direction register | 0x9Ah | 0x199h | 0x19Ah | 0x19Bh | 0: Line is input<br>1: Line is output |

Assigning the used GPIO ports is described in the table below.

**Table 2-6 – Purpose of GPIO ports**

| I/O port line | I/O port line direction | Description |
|---|---|---|
| GPIO_P0 [7:0] | Input / Output | Reserved.<br>The lines are used for COM1 port (RS-232). |
| GPIO_P1 [7:0] | Input / Output | Reserved.<br>The lines are used for COM2 port (RS-232). |
| GPIO_P2 [4:0] | Input / Output | Reserved.<br>The lines are used for configuring FPGA. |
| | | |
| GPIO_P2 [5] | Input | Reading the state of the X4 [1-2] "BIOS_DEFAULT" switch.<br>When this switch is set, the BIOS Setup settings are reset to the default state, at module's startup. |
| GPIO_P2 [6] | Input | Reading the state of the X4 [3-4] "BIOS_WRITE_PROTECT" switch.<br>When this switch is set, the assigned BIOS Setup settings are protected against changes. |
| GPIO_P2 [7] | Output | Reserved. |
| GPIO_P3 [3:0] | Input / Output | Reserved.<br>The lines are used for the SPI port (access to FRAM). |
| GPIO_P3 [5:4] | Input / Output | Reserved.<br>The lines are used for the I2C0 port (temperature sensor, GPIO port). |
| GPIO_P3 [6] | Output | Permitting operation of the watchdog timer integrated into the power supervisor (actuation interval is 1.6 s).<br>Set the line to '0' for permission. |
| GPIO_P3 [7] | Output | Reset/restart of the watchdog timer integrated into the power supervisor (actuation interval 1.6 s). To reset, it is necessary to change the state to the opposite ('1' → '0' or '0' → '1'). |
| | | |
| GPIO_P4 [7:0] | Input / Output | Reserved.<br>The lines are used for COM3 port (RS-485). |
| GPIO_P5 [7:0] | Input / Output | Reserved.<br>The lines are used for COM4 port (RS-485). |
| GPIO_P6 [7:0] | Input / Output | Reserved.<br>The lines are used for PATA interface (CompactFlash slot). |
| GPIO_P7 [7:0] | Input / Output | |
| GPIO_P8 [7:0] | Input / Output | |
| GPIO_P8 [7:0] | Input / Output | |
| | | |
| GPIO_PA [5:0] | Input | Reserved |
| GPIO_PA [7:6] | Output | Reserved.<br>Signal of transmission direction control for transmitter/receiver of RS-485 interface, COM4 port. |
| GPIO_PA [7] | Output | Reserved.<br>Signal of transmission direction control for transmitter/receiver of RS-485 interface, COM3 port. |

## 2.7 WDT0 and WDT1 watchdog timers

The Vortex86DX3 CPU chip has 2 x configurable hardware watchdog timers.
The WDT0 timer registers are accessed through port 65h and ports 22h (Address Index Register) and 23h (Data Register). To access the registers, it is necessary to write the address of the port to port 22h, the reading and/or writing of data of which is carried out through the 23h port. Tables 2-20… 2-28 provide a detailed description of the WDT0 watchdog timer control registers.
The WDT1 timer registers are accessed through ports 67h - 6Dh. Tables 2-29 ... 2-35 below provide a detailed description of the WDT1 watchdog timer control registers.

**Table 2-7 - WDT0 restart register**

| Address | Type of action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 65h | Write | RST_WDT0 | | | | | | | |
| | Read | - | - | - | - | - | - | - | - |

Any writing to this port will lead to restart of the WDT0 timer.

**Table 2-8 – Index register of the WDT0 port address**

| Address | Type of action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 22h | Write | ADDR_REG_WDT0 | | | | | | | |
| | Read | - | - | - | - | - | - | - | - |

ADDR_REG_WDT0. Indicates the address of the selected WDT0 watchdog timer register for access via data register 23h.

**Table 2-9 – Data register of the WDT0 port**

| Address | Type of action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 23h | Write | WRDATA_REG_WDT0 | | | | | | | |
| | Read | WRDATA_REG_WDT0 | | | | | | | |

WRDATA_REG_WDT0. Contains data for writing to the internal register of the WDT0 timer, which address is specified in the ADDR_REG_WDT0 field of the 22h index register.

WRDATA_REG_WDT0. Contains data when reading from the internal register of the WDT0 timer, which address is specified in the ADDR_REG_WDT0 field of the 22h index register.

**Table 2-10 – Register of WDT0 timer control**

| Address (in the register of address 22h) | Action | Bits (in the data register 23h) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 37h (40h) | Write | - | WDT0_WE | - | - | - | - | - | - |
| | Read | - | WDT0_WE | - | - | - | - | - | - |

WDT1_WE. Permission for WDT0 watchdog timer operation.
1 – permitted;
0 – prohibited (default value).

**Table 2-11 – Register of WDT0 event selection**

| Address (in the register of address 22h) | Action | Bits (in the data register 23h) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 38h (D0h) | Write | WDT0_SSEL | | | | | | | |
| | Read | | | | | | | | |

WDT0_SSEL. Selecting an event at the end of the timer WDT0 counting.

0000 – reserved;
0001 – IRQ[3];
0010 – IRQ[4];
0011 – IRQ[5];
0100 – IRQ[6];
0101 – IRQ[7];
0110 – IRQ[9];
0111 – IRQ[10];
1000 - IRQ[11];
1001 – IRQ[12];
1010 – IRQ[14];
1011 – IRQ[15];
1100 – NMI;
1101 – module reboot (default value);
1110 – reserved;
1111 – reserved.

**Table 2-12 - Register CNT0 of the WDT0 timer value**

| Address (in the register of address 22h) | Action | Bits (in the data register 23h) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 39h (00h) | Write | WDT0_CNT0 | | | | | | | |
| | Read | WDT0_CNT0 | | | | | | | |

WDT1_CNT0. Bits [7:0] of the WDT0_CNT[23:0] counter of WDT0 timer. Counter resolution is 30.5 μs.

**Table 2-13 – Register CNT1 of the WDT0 timer value**

| Address (in the register of address 22h) | Action | Bits (in the data register 23h) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 3Ah (00h) | Write | WDT0_CNT1 | | | | | | | |
| | Read | WDT0_CNT1 | | | | | | | |

WDT0_CNT1. Bits [15:8] of the WDT0_CNT[23:0] counter of WDT0 timer. Counter resolution is 30.5 µs.

**Table 2-14 - Register CNT2 of WDT0 timer value**

| Address (in the register of address 22h) | Action | Bits (in the data register 23h) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 3Bh (20h) | Write | WDT0_CNT2 | | | | | | | |
| | Read | WDT0_CNT2 | | | | | | | |

WDT0_CNT2. Bits [23:16] of the WDT0_CNT[23:0] counter of WDT0 timer. Counter resolution is 30.5 µs.

**Table 2-15 –WDT0 timer state register**

| Address (in the register of address 22h) | Action | Bits (in the data register 23h) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 3Ch (00h) | Write | WDT0_WDTF | WDT0_WDTRL | - | - | - | - | - | - |
| | Read | WDT0_WDTF | - | - | - | - | - | - | - |

WDT0_WDTF. Flag of WDT0 timer triggering.

1 – timer was triggered (writing “1” to this bit resets the flag);

0 – timer was not triggered.

WDT0_WDTRL. WDT0 timer reboot.

1 – Reboot of the WDT0_CNT counter;

0 – This value is not allowed to be written.

**Table 2-16 - WDT1 restart register**

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 67h | Write | RST_WDT1 | | | | | | | |
| | Read | - | - | - | - | - | - | - | - |

Any writing to this port will force the WDT1 timer to restart.

**Table 2-17 –WDT1 timer control register**

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 68h (00h) | Write | - | WDT1_WE | - | - | - | - | - | - |
| | Read | - | WDT1_WE | - | - | - | - | - | - |

WDT1_WE. Permission for WDT1 watchdog timer operation.

1 – Permitted;

0 – Prohibited (default value).

**Table 2-18 - WDT1 event selection register**

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 69h (00h) | Write | WDT1_SSEL | | | | - | - | - | - |
| | Read | - | - | - | - | - | - | - | - |

WDT1_SSEL. Selecting an even upon completion of the WDT1 timer counting.

0000 – Reserved (default value);
0001 – IRQ[3];
0010– IRQ[4];
0011– IRQ[5];
0100– IRQ[6];
0101– IRQ[7];
0110– IRQ[9];
0111– IRQ[10];
1000– IRQ[11];
1001– IRQ[12];
1010– IRQ[14];
1011– IRQ[15];
1100– NMI;
1101– module restart;
1110 – Reserved;
1111 – Reserved.

**Table 2-19 – Register CNT0 of WDT1 timer value**

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 6Ah (00h) | Write | WDT1_CNT0 | | | | | | | |
| | Read | WDT1_CNT0 | | | | | | | |

WDT1_CNT0. Bits [7:0] of WDT1_CNT[23:0] counter of WDT1 timer. Counter resolution is 30.5 µs.

**Table 2-20 – Register CNT1 of WDT1 timer value**

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 6Bh (00h) | Write | WDT1_CNT1 | | | | | | | |
| | Read | WDT1_CNT1 | | | | | | | |

WDT1_CNT1. Bits [15:8] of WDT1_CNT[23:0] counter of WDT1 timer. Counter resolution is 30.5 μs.

**Table 2-21 – Register CNT2 of WDT1 timer value**

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 6Ch (00h) | Write | WDT1_CNT2 | | | | | | | |
| | Read | WDT1_CNT2 | | | | | | | |

WDT1_CNT2. Bits [23:16] of WDT1_CNT[23:0] counter of WDT1 timer. Counter resolution is 30.5 μs.

**Table 2-22 - WDT1 timer state register**

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 6Dh (00h) | Write | WDT1_WDTF | - | - | - | - | - | - | - |
| | Read | WDT1_WDTF | - | - | - | - | - | - | - |

WDT1_WDTF. Flag of WDT1 timer triggering.
1 – timer was triggered (writing “1» to this bit resets the flag);
0 – timer was not triggered.

## 2.8 Description of internal registers

The set of control registers is specified below.

**BA = 0x280h**

| Address | Port | Notes |
|---|---|---|
| | | |
| BA+01h | LED control register | Software control of green and red user LEDs |
| | | |
| BA+05h | The port for controlling NMI/IRQ interrupt sources | The port for control of an interrupt over the NMI/IRQ line of the processor (External WDT, Power Fail, SYSTEM EVENT, ISA$IOCHK#). |

| Address | Port | Notes |
|---|---|---|
| BA+06h<br>BA+07h | Register for control of interrupts IRQ7,11,12,15<br>ISA control register | Port for control of interrupts IRQ7,11,12,15 of the processor (sets the source over the processor interrupt lines (UNIO, ISA$IRQx, ADC/DAC Ready, SYSTEM EVENT, MXKEY)).<br>Enabling/ Disabling ISA bus buffers. |
| | | |
| BA+0Ch | Interrupt state register | The register is designed for reading the state and permitting generation of an interrupt (IRQ15, IRQ14, IRQ12, IRQ11, IRQ7, IRQ5, IRQ4, IRQ3). |
| BA+0Dh | I/O port register | The register is designed for reading one page of the input port (IO_A, IO_B), is available independently of the loaded UNIO port firmware. |
| BA+0Eh | Register of the FPGA scheme version code | Register of the FPGA basic scheme version code |
| BA+0Fh | Register for selecting the active firmware of the digital I/O port (write)<br>Module code register | Selection of the active configuration of the port via the byte port BA + 0Fh.<br>Reading the module code (0x8Ah). |
| | | |
| BA+1xh | Registers of the IO_A I/O port | |
| BA+2xh | Registers of the IO_B I/O port | |

### 2.8.1 LED control register

Ensures possibility of software control of 2 LED emitters, as well as reading their current state (BA = 0x280h).

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BA+01h** | Write | - | - | - | - | - | - | LEDR | LEDG |
| **BA+01h** | Read | - | - | - | - | - | - | SLEDR | SLEDG |

LEDG Green LED control. Writes '1' - enabling the LED, writes '0' – disabling the LED.
LEDR Red LED control. Writes '1' - enabling the LED, writes '0' – disabling the LED.

SLEDG Green LED state. '1' - enabled, '0' – disabled (by default).
SLEDR Red LED state. '1' - enabled, '0' – disabled (by default).

### 2.8.2 Register for selecting the active firmware of the UNIO port

Selecting an active configuration of the UNIO port via the byte port.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x28Fh | Write | IO_B CODE | | | | IO_A CODE | | | |

IO_B CODE, IO_A CODE - numeric code of active firmware of digital input-output ports IO_A (XP16) and IO_B (XP18): "n00" – 0 (by default), "p55" – 1, "c02" – 2, "q04" – 3, , "t00" – 4. The "n00" firmware is selected at the reset, if no other firmware is selected in BIOS Setup.

### 2.8.3 System interrupt control/state register

Enables / disables the use of external signals when generating an NMI processor interrupt, the “SYSTEM EVENT” signal when writing to the port. The port also makes it possible for you to define the NMI hardware source and “SYSTEM EVENT”.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BA+05h | Write | IOCHK_EN | PFO_EN | WDO_EN | EXT_EN | LED_EN | - | SE_NMI_EN | NMI_EN |
| BA+05h | Read | IOCHK | PFO | WDO | EXT | LED_EN | - | SE_NMI_EN | NMI_EN |

IOCHK_EN 'ISA_IOCHK#' switching to the NMI line of the processor. Enable – '1' (by default), disable – '0'.

PFO_EN 'Power Fail' switching to the 'SYSTEM EVENT' line. Enable – '1', disable – '0' (by default).

WDO_EN 'WatchDog Timeout' switching to the 'SYSTEM EVENT' line. Enable – '1', disable – '0' (by default).

EXT_EN 'RMTRES' switching to the 'SYSTEM EVENT' line. Enable – '1', disable – '0' (by default).

SE_NMI_EN 'SYSTEM EVENT' switching to the NMI line of the processor. Enable – '1', disable – '0' (by default).

NMI_EN Enable NMI generation. Generation of the NMI interrupt of the processor by an event from 'ISA_IOCHK#' or "SYSTEM EVENT" – when writing '1' (by default); disable – when writing '0' (in this case on the NMI hardware line: 'Z'- state).

IOCHK Tag 'ISA_IOCHK#' - '1': active level of 'ISA_IOCHK#' signal.

PFO Tag 'POWER FAIL' - '1': lowering the "+5V" input supply voltage below the level of 4.6 V.

WDO Tag 'WATCHDOG Timeout' - '1': activation of the watchdog timer (WDT2) integrated into the supervisor.

EXT Tag 'RMTRES' - '1': external source of reset/interrupt (XP17-XP18).

LED_EN — Permit LED blinking when accessing the FPGA ports (~80 ms) and while switching to the active level (edge) on one of the IRQx, NMI interrupt lines (~160 ms). To activate the LED when there is an edge on the interrupt line, the relevant IRQx_EN bit of register BA+0Ch and NMI_EN of the register BA+05h must be set. Enable – '1', disable – '0' (is set by default).

**Attention!**
"SYSTEM EVENT" is generated by “or” from the following sources:
– "Power Fail" (lowering the power supply voltage value down to the level of 4.6V),
– "WatchDog Timeout" (actuation of the external watchdog timer),
– "RMTRES" (external source of reset / interrupt XP17 / XP18).
NMI signal of the processor is generated by “or” from the sources “SYSTEM EVENT” and “ISA_IOCHK”.

**Attention!**
After each generation of an interrupt on the "ISA_IOCHK#', 'POWER FAIL', 'WATCHDOG Timeout', 'RMTRES', the corresponding interrupt flag must be reset by sequentially writing '0' and then '1' to the relevant interrupt permitting bit. Failure to do so, no further generation of interrupts from this port will be performed.
For example, after generating an interrupt on the 'RMTRES' event, you must first reset the 'EXT_EN' flag to '0' in the interrupt handler, and set it to '1' afterwards.

### 2.8.4 Interrupt control registers

Using these registers, the source of interrupts on the IRQ7, IRQ11, IRQ12, IRQ15 lines of the processor can be set. Interrupts of the ISA bus (IRQ7, IRQ11, IRQ12, IRQ15) or interrupts of UNIO ports, external interrupt source, Power Fail signals, watchdog timer signals can be connected to each line independently.
The register is also used to enable/disable the buffer elements of the ISA bus. When the buffer elements are disabled by setting the bit to '0', the ISAE external devices connected to the ISA bus (PC/104) will not be available.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BA+06h** | Write/Read | i15SEL[1:0] | | i14SEL[1:0] | | i11SEL[1:0] | | i7SEL[1:0] | |
| **BA+07h** | Write | IUA | IUE | - | - | - | IOB_EN | IOA_EN | ISAE |
| **BA+07h** | Read | IUA | IUE | - | - | - | IOB_ST | IOA_ST | ISAE |

i7SEL[1:0] Selector code of IRQ7 line. Possible options: IRQ7 interrupt of ISA bus is not used (b'00, by default); UNIO (b'01); "SYSTEM EVENT" (b'10); ISA$IRQ7 (b'11).

i11SEL[1:0] Selector code of IRQ11 line. Possible options: IRQ11 interrupt of ISA bus is not used (b'00, by default); UNIO (b'01); ADC/DAC Ready (b'10); ISA$IRQ11 (b'11).

i14SEL[1:0] Selector code of IRQ14 line. Possible options: IRQ14 interrupt of ISA bus is not used (b'00, by default); ISA$IRQ3 (b'01); ISA$IRQ4 (b'10); ISA$IRQ14 (b'11).

i15SEL[1:0] Selector code of IRQ15 line. Possible options: IRQ15 interrupt of ISA bus is not used (b'00, by default); UNIO (b'01); "SYSTEM EVENT" (b'10); ISA$IRQ15 (b'11).

**Attention!**

To control the interrupt source using this register, you must set the interrupt used to the “Reserved” state on the “PCIPnP” BIOS Setup page, see PCI/PnP (additional PCI plug and play settings). Otherwise, the interrupt will be used for embedded devices.

IUA Combining interrupts of UNIO port. While writing '1' to this bit, the interrupts from channels IO_A[23:0], IO_B[23:0] are combined by “and”, if '0' is written - by “or” (default value, similar to combining interrupts of FPGA matrices within the UNIO96-5 module). Combining by “and“ is possible only if all the bits of permitting interrupts from all the UNIO (IOB_EN, IOA_EN) ports are set

IUE Permission of interrupt from UNIO port. When writing '1' to this bit, the general interrupt of UNIO port is permitted, when writing '0' – prohibited.

IOB_EN Permission of interrupt from UNIO IO_B - '1' (XP18) port. When writing '1' to this bit, the interrupt from the IO_B port is permitted, '0' – prohibited. Setting this bit to '0' also resets the state of IOB_ST flag.

IOA_EN Permission of interrupt from UNIO IO_A - '1' (XP16) port. When writing '1' to this bit, the interrupt from the IO_A port is permitted, '0' – prohibited. Setting this bit to '0' also resets the state of IOA_ST flag.

ISAE Control of the buffer elements of ISA bus. '0': disabled (outputs are in the 'Z' state). '1': buffer elements of ISA bus are enabled (by default – when switched on or reset).

IOB_ST Interrupt indicator of the port UNIO IO_B - '1' (XP18).

IOA_ST Interrupt indicator of the port UNIO IO_A - '1' (XP16).

**Attention!**
After each generation of interrupts from UNIO ports, the corresponding interrupt flag must be reset by sequentially writing '0' and then '1' to the relevant interrupt permission bit. Failure to do so, further generation of interrupts from this port will not be performed. E.g., after generating an interrupt from the IO_A port, you must first reset the 'IOA_EN' flag to '0' in the interrupt handler, then set it to '1'.

**Attention!**
To ensure that there is no false generation of interrupts immediately after enabling the generation of interrupts from the UNIO ports (after setting the relevant bits in the internal registers of the UNIO firmware), you must first configure the interrupts, and then set the interrupt permission bit from the UNIO port - 'IOA_EN', 'IOB_EN' in the register “Interrupt Management”.

### 2.8.5 Register of interrupt states

The register has the size of 8 bits and is designed for reading the state and permission of interrupt generation.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BA+0Ch | Write | IRQ15_EN | IRQ14_EN | IRQ12_EN | IRQ11_EN | IRQ7_EN | IRQ5_EN | IRQ4_EN | IRQ3_EN |
| BA+0Ch | Read | IRQ15 | IRQ14 | IRQ12 | IRQ11 | IRQ7 | IRQ5 | IRQ4 | IRQ3 |

IRQ15_EN…IRQ3_EN Permission for generating the relevant interrupt. By default: '1'.

IRQ15…IRQ3 Flag of interrupt state. For lines IRQ15, IRQ12, IRQ11, IRQ7 the source is selected by writing to the BA+06h port. For lines IRQ14, IRQ5, IRQ4, IRQ3 the source would be the relevant interrupt line of ISA bus. The flag is latched prior to its erasure. To erase it, it is necessary to sequentially reset and set the IRQ15_EN…IRQ3_EN bit. When writing to the BA+06h or BA+07h port, all the flags are also reset.

### 2.8.6 Register of I/O ports states

The register has the size of 8 bits and is designed for reading one page of input port (IO_A, IO_B, IO_C). The port is available regardless of the loaded UNIO port firmware.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BA+0Dh | Write | IO_PAGE | | | | | | | |
| BA+0Dh | Read | IO[7:0] | | | | | | | |

IO_PAGE Page number of I/O port from 0 to 9. The default state is “0” (the register is not connected to any of the pages).

"1" for IO_A[7:0], "2" for IO_A[15:8], "3" for IO_A[23:16],
"4" for IO_B[7:0], "5" for IO_B[15:8], "6" for IO_B[23:16],

IO[7:0] Register of the state of I/O port page.

### 2.8.7 Registers of the code of FPGA scheme version and device

Code of the version number/ revision of FPGA scheme and numeric code of the device are available by reading via the byte ports BA+0Eh and BA+0Fh.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BA+0Eh | Read | Ver | | | | Rev | | | |
| BA+0Fh | Read | BOARD_CODE | | | | | | | |

Ver – the numeric code of the version number of the FPGA scheme (from 0 to 9);

Rev – the numeric code of the FPGA scheme revision number (from 0 to 9);

BOARD_CODE – the numeric code of the device (0x3Fh).

### 2.8.8 Registers of UNIO digital I/O ports

The registers are designed for control of 48 digital I/O lines.

The ports are divided into 3 groups:

- IO_A[23:0] channels (similar to the FPGA1 matrix in the UNIO96-5 module, except for the Base Address: **BA1=BA+10h,** by default 0x290h);
- IO_B[23:0] channels (similar to the FPGA2 matric in the UNIO96-5 module, except for the Base Address: **BA2=BA+20h,** by default 0x2A0h);

### 2.8.9 Registers of UNIO port scheme identification codes

Each UNIO port within the device has its own identifier that matches the scheme code: n00, p55, c02, q04, t00, etc.

The identifier can be read through the byte ports with the addresses BAx + 0Eh, BAx + 0Fh, where BAx is the base address of the UNIO port in the device (by default, the base addresses for the IO_A ports are 0x290, IO_B - 0x2A0)

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BAx+0Eh | Read | a ... z | | | | | | | |
| BAx+0Fh | Read | SN[7:0] | | | | | | | |

a...z ASCII-code of the lowercase letters from a to z.

SN[7:0] Code of the scheme number, from 0 to 255.

### 2.8.10 Description of the UNIO port configuration option "n00"

Digital-to-analog interface of Grayhill, Opto-22.

This section describes the purpose of UNIO port with the BAx address, scheme code "n00". The option “n00” is compatible with the n00 option for the UNIOxx family and makes it possible to implement 24 channels of digital-to analog interface within a single FPGA matrix with the following capabilities:

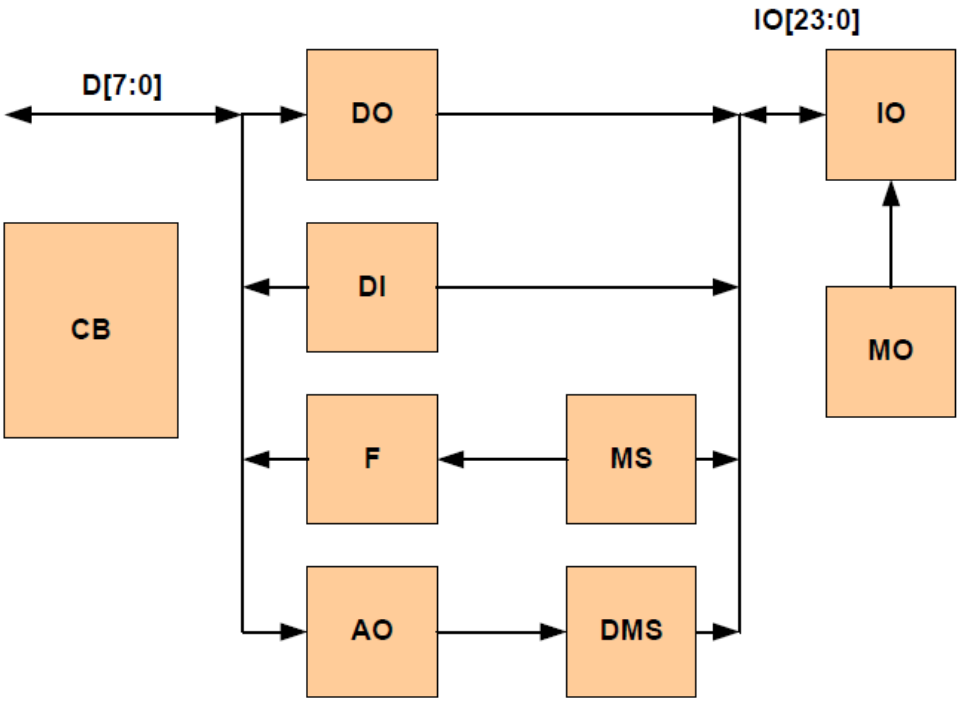
- Configuration of channels in an arbitrary combination (input/output, digital/analog);
- Issue / reading of digital signals;
- Measuring frequency over any channel: the number of periods of the measured frequency - 8, range: from 3.1 kHz to 5 MHz (period of the filling frequency 20 nsec) or from 3.1 Hz to 5 kHz (period of filling frequency 20 μs), accuracy: up to 16 bits;
- Serving the Grayhill analog input modules on any channel with a resolution 12 bits (conversion time: 750 μs) without using processor resources;
- Programming the Grayhill 73G Series analog output modules over any channel (programming time: 800 μs) without using processor resources;
- Generation of interrupts upon completion of frequency measurement and programming of 73G Series modules operations;
- Control of input states;
- Debouncing of the input signal when measuring the frequency of 100 nsec/1.6 μs.

The basic option “n00” differs from the UNIO firmware set by increasing the bit width of the data register of the frequency meter to 16 (the basic option had 14), the possibility of setting the input signal debouncing when measuring the frequency (100 nsec/1.6 μs), the possibility of choosing the filling frequency (20 nsec/20 μs), expanding the range of the measured frequency up to 3.1 Hz ... 5 MHz (the basic option had 12.4 KHz ... 5 MHz), the possibility of generating a frequency for each output from the row given in the table below.

Scheme components

The matrix scheme components contains the following functional blocks:

- Control block (CB) generates internal control signals
- Register of the mask of outputs (MO) defines the assignment of each of the channels both output and input
- Register of digital outputs (DO) sets the logical state of the output channels
- Buffer of digital inputs (DI) makes it possible to read the states of input or output channels
- Frequency meter (F)
- Generator of the serial 12-bit code (AO) for control of Grayhill 73G Series analog output modules
- Multiplexer 24→1 (MS) is used for connection of any channel to the frequency meter
- Demultiplexor 1→24 (DeMS) connects generator of serial code to any channel
- Input/output block (IO)




Control register is available via port with the address BAx+0 and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+00h** | Write | – | – | – | F_CLK | CLKD | IAO | IF | BANK |
| **BAx+00h** | Read | AO_RDY | F_RDY | F_ERR | – | – | – | – | – |

AO_RDY Readiness of the serial code generator. The bit is set upon completion of the code issuance. The bit is reset during issuance.

F_RDY Readiness of the frequency meter. The bit is reset during operation of the meter. The bit is set at the end of the measurement, in the event of the measurement error (ERR) or meter overflow (OVR).

F_ERR Error in the frequency meter channel. The bit is set where there is no frequency in the measurement channel. The bit is reset when the measurement channel is changed or the meter is started.

IAO Permission of interrupt generation at the end of the serial code generation. Setting a bit enables generation, resetting a bit disables generation.

IF Permission of interrupt generation at the end of frequency measurement. Setting a bit enables generation, resetting a bit disables generation.

BANK Bank of ports is designed for switching access to different groups of registers. The default state is “0”.

CLKD Code of debouncing time of input channel when measuring frequency: '0' = 100 ns, '1' = 1.6 µs. The default state is “0”.

F_CLK Code of the reference frequency period when measuring frequency: '0' = 20 ns, '1' = 20 µs. The default state is “0”.

Output mask registers have the size of 24 bits. Each bit defines the relevant channel to operate as an input (bit value “0”) or output (bit value “1”).

The register is available by writing and reading via ports with the addresses BAx+1 … BAx+3 when the BANK (BANK=1) bit is set. The bits in the register are numbered starting from the least significant bit.

| Address | BANK | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+01h** | 1 | IOx_MSK [7:0] | | | | | | | |
| **BAx+02h** | 1 | IOx_MSK [15:8] | | | | | | | |
| **BAx+03h** | 1 | IOx_MSK [23:16] | | | | | | | |

Registers of outputs and input state registers have a total size of 24 bits.

Such registers are accessed via the ports with the addresses BAx+1 … BAx+3 and when the BANK (BANK=0) bit is reset. The input is set by writing “1” to the bit of the relevant channel of output register (the output is reset by writing “0”).

The channel state is defined when reading the relevant bit of the input buffer. The bits are numbered starting from the least significant bit.

Availability of internal feedback makes it possible to control the output channels along the inner ring (in order to do so, you should use the register BA + 0Dh).

| Address | BANK | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+01h** | 0 | IOx [7:0] | | | | | | | |
| **BAx+02h** | 0 | IOx [15:8] | | | | | | | |
| **BAx+03h** | 0 | IOx [23:16] | | | | | | | |

Registers of the frequency outputs mask have a total size of 24 bits.

Such registers are accessed via the ports with the addresses BAx+9… BAx+10 when the BANK (BANK=0) bit is reset. Permission of the frequency output is gained by writing “1” to the bit of the relevant register channel (prohibition – by writing “0”). The bits in the register are numbered starting from the least significant bit.

| Address | BANK | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+09h** | 0 | IOx_F [7:0] | | | | | | | |
| **BAx+0Ah** | 0 | IOx_F [15:8] | | | | | | | |
| **BAx+0Bh** | 0 | IOx_F [23:16] | | | | | | | |

IOx_F[23:0] Number of the channel connected to the frequency meter F (code 0-23)

| Channel | Period |
|---|---|
| IO[0] | 250 ns |
| IO[1] | 500 ns |
| IO[2] | 1 µs |
| IO[3] | 2 µs |
| IO[4] | 4 µs |
| IO[5] | 5 µs |
| IO[6] | 8 µs |
| IO[7] | 10 µs |

| Channel | Period |
|---|---|
| IO[8] | 20 µs |
| IO[9] | 40 µs |
| IO[10] | 50 µs |
| IO[11] | 80 µs |
| IO[12] | 100 µs |
| IO[13] | 200 µs |
| IO[14] | 400 µs |
| IO[15] | 500 µs |

| Channel | Period |
|---|---|
| IO[16] | 800 ms |
| IO[17] | 1 ms |
| IO[18] | 2 ms |
| IO[19] | 4 ms |
| IO[20] | 5 ms |
| IO[21] | 8 ms |
| IO[22] | 10 ms |
| IO[23] | 20 ms |

Frequency meter (F) has two registers: control register F and data register F. To make the frequency meter work, the channel used should be set as “input”.

Control register F is available by writing and reading via the port with the address BAx+4 and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+04h** | Write | F_ST | – | – | CHF[4:0] | | | | |
| **BAx+04h** | Read | – | – | – | CHF[4:0] | | | | |

CHF[4:0] Number of the channel connected to the frequency meter F (code 0-23)

F_ST Start of the measurement. Setting a bit starts the frequency measurement via the selected channel. Reset of the bit interrupts the measurement (in this case, the value of the data register F is undefined!).

Data register F has a size of 16 bits and is available by reading via the port with the address BAx+6. Upon completion of the frequency meter’s operation, the user has access to a 16-bit code of the duration of 8 periods of the measured frequency (without the least significant bit of F0). The weight of F0 least significant bit is 20 nanoseconds.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+06h** | Read | F[8:1] | | | | | | | |
| **BAx+07h** | Read | F[16:9] | | | | | | | |

F[15:1] The duration code of 8 measured frequency periods without a zero bit (valid only when the F_ERR bit of register BAx + 00h is reset).

Generator of serial code (AO) has two registers: AO control register and AO data register.

For proper operation of the generator, the used channel must first be set as “output” by writing “1” to the corresponding bit of the output mask register, the state of the used channel must be set by writing “1” to the corresponding bit of the output register.

The AO control register is available by writing via port with the address BAx+5 and hs the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BAx+05h | Write | AO_ST | – | – | CHAO [4:0] | | | | |

CHAO[4:0] Number of channel for AO data output (code 0-23)

AO_ST Start of the AO generator. Setting this bit will start issuing a code from the AO data register. Resetting the bit will interrupt the code issue (while the state of the 73G Series is undefined!).

The AO data register has the size of 12 bits and is available by writing via the port with the BAx+6 address. It has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BAx+06h | Write | AO[7:0] | | | | | | | |
| BAx+07h | Write | | | | | AO[11:8] | | | |

AO[11:0] Data code of the AO serial code generator.

The firmware identifier is available by reading via byte ports with the addresses BA+0x0E, BA+0x0F and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BAx+0Eh | Read | 'n' | | | | | | | |
| BAx+0Fh | Read | SN [7:0] | | | | | | | |

'n' ASCII-code of the lowercase Roman letter 'n' (0x6E).
SN[7:0] The decimal code of scheme number (SN[7:0]=0).

Fragment of an example in C language for launching frequency meter, selection of measuring channel, period calculation is given below:

```
#define  BA       0x290 // base address


unsigned int  BA;             // Base address of the input
port float  f,p;              // Frequency & Period
unsigned int  ch;             // Channel number
unsigned int  d;             // Raw frequency data read from port


outportb(BA,0);               // set f_clk = 20 ns, bnk=0
// outportb(BA,0x10);         // set f_clk = 20 us, bnk=0
 outportb (BA+4,0x80|ch);  // start frequency measurement
 do { st=inportb(BA);
} while (( st != 0xC0 ) && ( st != 0xE0 )); // wait for frequency measurement ready or error
 d=inport(BAr+6);             // read period value f[16:1] p=(float)(d)*2*20/8;    // calculate
period f[16:0], ns f=100000*1/p;                    // calculate period f[16:0], KHz if f_clk=20 ns,
Hz if f_clk=20 us,
 printf  ("BA:%003X   Ch:%02d   D:%0004X   P:%6.0f   ",BAs,BAr,ch,d,p);
printf ("F:%8.3fKHz \n",f);
// printf ("F:%8.3fHz \n",f);
```

Fragment of an example in C language for launching serial code generator and channel selection is given below:

```
#define  AO_RDY 0x80     // AO ready flag
#define  BA0      0x280 // Control base address
#define  BA1      0x290 // UNIO base address
#define  BA2      0x2A0 // UNIO base address

        unsigned int data;     // data written to AO
        unsigned int st;              // AO status
        unsigned int ch;              // channel number
       outportb (BA0+0x0F,0x00); // select n00 configuration for UNIO port with BA1,BA2 address
        outport (BA1+6,data);         // write data to AO reg
        outportb (BA1+5,0x80|ch); // start AO generation
  do { st=inportb(BA1); } while ((st&AO_RDY)!=AO_RDY);  //wait until AO ready
```

**Table 2-23 - Table of external connections for “n00” firmware**

| Pin # | Function | Pin # | Function |
|---|---|---|---|
| 1 | IO(12) | 2 | +5V_EXT |
| 3 | IO(13) | 4 | IO(10) |
| 5 | IO(14) | 6 | IO(11) |
| 7 | IO(15) | 8 | IO(9) |
| 9 | IO(23) | 10 | IO(8) |
| 11 | IO(21) | 12 | IO(22) |
| 13 | IO(16) | 14 | IO(20) |
| 15 | IO(18) | 16 | IO(17) |
| 17 | IO(19) | 18 | IO(7) |
| 19 | IO(0) | 20 | IO(6) |
| 21 | IO(1) | 22 | IO(5) |
| 23 | IO(2) | 24 | IO(4) |
| 25 | IO(3) | 26 | GND |

### 2.8.11 Description of UNIO port configuration option “p55”

24-channel I/O port (emulator of IC 8255 mode 0).

This section describes the purpose of the UNIO port with address BAx (0xA110), code “p55”.

The “p55” option is fully compatible with the option p55 for the UNIOxx family and makes it possible to implement 24 channels of digital-to-analog interface within a single FPGA matrix, with the following capabilities:

- 24 digital I/O channels;
- Possibility of programming the channel direction (2 groups with 8 channels each and 2 groups with 4 channels each);
- Programmable debouncing time by inputs: 100 ns, 1.6 µs, 4 ms, 120 ms;
- Programmable edge of the event for each group of 8 inputs: 1 → 0, 0 → 1, (1 → 0 + 0 → 1);
- Generation of the masked interrupt from each group of 8 inputs.

The scheme of the “p55” option includes a block of control signals, control register of channel operation direction, a 24-bit register of outputs, I/O buffers, an input debouncing block and an event registration block.

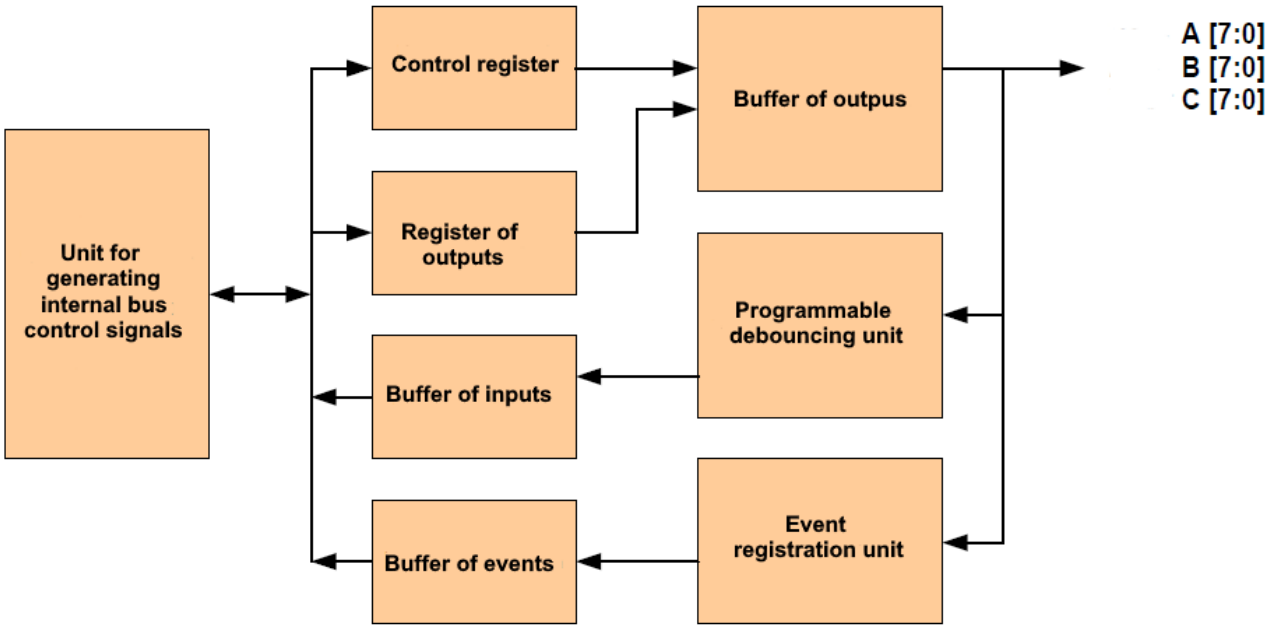


**Table 2-24 – Table of "p55" firmware registers**

| Designation | Channel | Address | Note |
|---|---|---|---|
| Port A | IO [7 : 0] | BAx + 0 | Designed for setting outputs and reading the status of inputs/outputs |
| Port B | IO [15 : 8] | BAx + 1 | |
| Port C | IO [23 : 16] | BAx + 2 | |
| Control register | 23 – 0 | BAx + 3 | Designed for setting the direction (input/output) of channel groups |
| Register of input debouncing time and edge of events | 7 – 0<br>15 – 8<br>23 – 16 | BAx + 4 | Designed for setting debouncing time by channel groups |
| Register of interrupt mask from the event block | 7 – 0<br>15 – 8<br>23 – 16 | BAx + 5 | Enabling/disabling event block interrupts |
| Event logging block | 7 – 0<br>15 – 8<br>23 – 16 | BAx + 6<br>BAx + 7<br>BAx + 8 | Reset and read of events over each channel |
| Scheme identifier | | BAx + 0Eh<br>BAx + 0Fh | |

Control register is available via the port with the address BAx+0 and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+03h** | Write/Read | – | – | – | Port A | Port C [7:4] | – | Port B | Port C [3:0] |

Programming of channel groups to input is carried out by writing '1' (default value after reset) to the relevant bit, to output - by writing '0'.

Ports A, B, C are available by writing and reading and is used to set the values of the output lines or read the status of the input and output lines. When reading the status of inputs/outputs, it is necessary to give special consideration to the delay of the debouncing block 100 ns… 120 ms.

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+00h** | Read | A [7:0] | | | | | | | |
| **BAx+01h** | Read | B [7:0] | | | | | | | |
| **BAx+02h** | Read | C [7:0] | | | | | | | |

After power-on or reset, all the channels are set to input, the output registers are reset to zero.

Register of input debouncing time is available by writing/reading and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+04h** | Write/Read | FR_C [1:0] | | FR_B [1:0] | | FR_A [1:0] | | T [1:0] | |

FR_C[1:0], FR_B[1:0], FR_A[1:0] Event edge code:

0x00 – not used, 0x01 – ↑ (leading edge), 0x02 – ↓ (falling edge),

0x03 – ↑ or ↓ (leading or falling edge)

T[1:0] Debouncing time code:

0x00 – 100 ns, 0x01 – 1.6 μs, 0x02 – 4 ms, 0x00 – 120 ms.

Interrupt mask register is designed for permitting/prohibiting interrupts from the event block, available via the byte port with the address BAx+13. The port is available by writing/reading and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+05h** | Read | – | – | – | – | – | IGR [2:0] | | |

IGR[2:0] Enable interrupts from input groups (byte-wise). When the bits are set, interrupt from the event block is permitted, respectively for channels [23:16], [15: 8], [7: 0]. The interrupt signal will be reset only after the corresponding bits in the event register are reset. Resetting the bits disables generation of interrupts.

Event registration block is available by writing and reading via the byte ports with the addresses BA+6.. BA+8 and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+06h** | Read | EV [7:0] | | | | | | | |
| **BAx+07h** | Read | EV [15:8] | | | | | | | |
| **BAx+08h** | Read | EV [23:16] | | | | | | | |

EV[23:0] Event register. The register bits are set when the state of the corresponding channel [23: 0] changes (the event edge is determined by the bits FR_x [1: 0]).

Only one event is memorized for each input. For registering the next event, it is necessary to reset the corresponding bit of the event register (writing '1' to the bit where the event is occurred).

The firmware identifier is available by reading via the byte ports with the addresses BA+0x0E, BA+0x0F and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+0Eh** | Read | 'p' | | | | | | | |
| **BAx+0Fh** | Read | SN [7:0] | | | | | | | |

'p' ASCII-code of the lowercase Roman letter 'p' (0x70).
SN[7:0] Decimal code of scheme number (SN[7:0]=55).

**Table 2-25 – External connections for the "p55" firmware**

| Pin # | Designation | Function | Pin # | Designation | Function |
|---|---|---|---|---|---|
| 1 | IO(12) | Port B [4] | 2 | +5V_EXT | – |
| 3 | IO(13) | Port B [5] | 4 | IO(10) | Port B [2] |
| 5 | IO(14) | Port B [6] | 6 | IO(11) | Port B [3] |
| 7 | IO(15) | Port B [7] | 8 | IO(9) | Port B [1] |
| 9 | IO(23) | Port C [7] | 10 | IO(8) | Port B [0] |
| 11 | IO(21) | Port C [5] | 12 | IO(22) | Port C [6] |
| 13 | IO(16) | Port C [0] | 14 | IO(20) | Port C [4] |
| 15 | IO(18) | Port C [2] | 16 | IO(17) | Port C [1] |
| 17 | IO(19) | Port C [3] | 18 | IO(7) | Port A [7] |
| 19 | IO(0) | Port A [0] | 20 | IO(6) | Port A [6] |
| 21 | IO(1) | Port A [1] | 22 | IO(5) | Port A [5] |
| 23 | IO(2) | Port A [2] | 24 | IO(4) | Port A [4] |
| 25 | IO(3) | Port A [3] | 26 | GND | – |

Fragment of an example in C language and port reading is given below:

```
#define  BA0          0x310 // Base address of internal control registers
#define  BA1         0xA110 // Base address of UNIO port
#define  A_DIR       0x10    // port A DIR = INPUT
#define  B_DIR       0x02    // port B DIR = INPUT
#define  C74_DIR      0x80    // port C[7:4] DIR = INPUT
#define  C30_DIR      0x01    // port C[3:0] DIR = INPUT
unsigned int IO_A,IO_B,IO_C;          // variable input states
unsigned int data=0x55;               // variable of output state
outportb (BA0+0x0F,0x01); // Select configuration p55 of UNIO port
outportb (BA1+3, A_DIR+B_DIR+C74_DIR+C30_DIR); // set the A,B,C ports as inputs


inportb (BA1+0,IO_A);         // Read A port
inportb (BA1+1,IO_B);         // Read B port
inportb (BA1+2,IO_C);         // Read C port
printf ("Port A: %02Xh, Port B: %02Xh, Port C: %02Xh", IO_A, IO_B, IO_C);


outportb (BA1+3, B_DIR+C74_DIR+C30_DIR); // ports B,C – inputs, port A - output
outportb (BA1+0, data); // set the lines of port A by the value of the "data" variable
```

### 2.8.12 Description of the configuration option "c02" of UNIO port

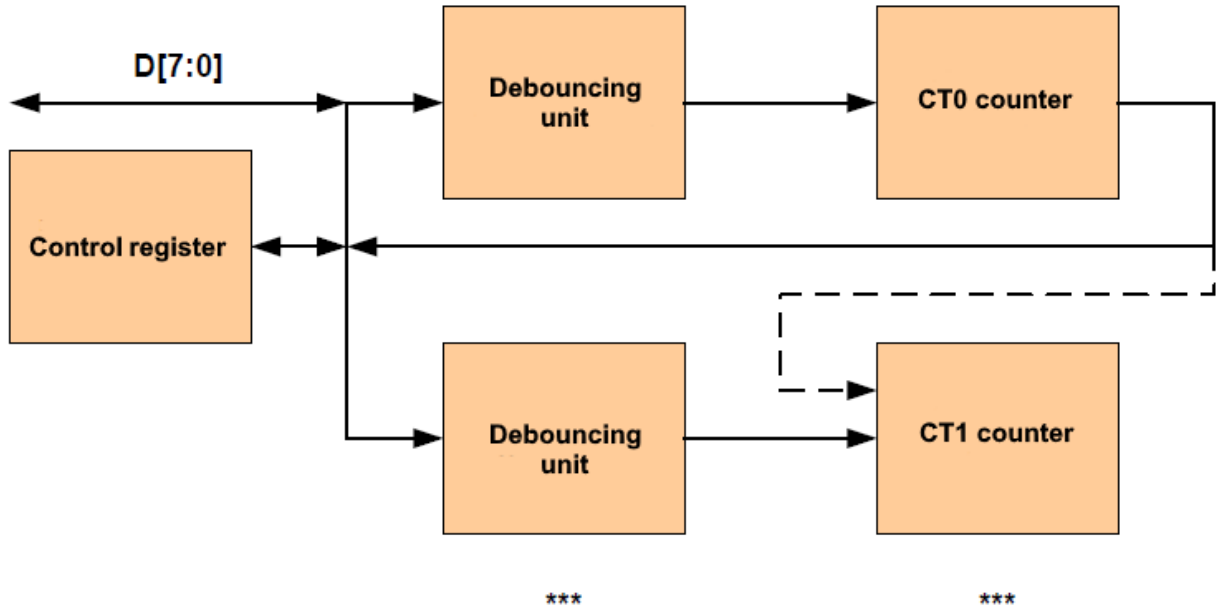
24 counters with debouncing of counter inputs and programmable transfer.

This section describes the purpose of UNIO port with the address BAx (0xA110), scheme code "c02".
The "c02" option is compatible with the option c02 for UNIOxx family and makes it possible to implement twenty four 16-bit / twelve 32-bit counters within a single FPGA matrix with the following capabilities:

- Twenty four 16-bit / twelve 32-bit counters;
- Debouncing programmable time by inputs: 100 ns, 1.6 μs, 4 ms, 120 ms;
- Programmable transfer of counters.

The difference from the basic option “c02” from the UNIO firmware set lies in increasing the number of counters up to 24 (the basic option had 16), the possibility of generating an interrupt in the event of counters overflow, and the presence of a module identifier register.

The block diagram of the two counter channels is shown below.




The control register is available by writing via the port with the address BAx+0 and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+00h** | Write | PE3 | PE2 | PE1 | PE0 | T [1:0] | | BANK [1:0] | |

PE3, PE2, PE1, PE0 Permission of counter transfers. If the bit is set in the relevant bank, it is allowed to transfer from the counter with address BA+00h to counter BA+02h, from the counter with address BA+04h to counter BA+06h, from the counter with address BA+08h to counter BA+0Ah.

T[1:0] Time code of counter input debouncing: '00' = 100 ns, '01' = 1.6 μs, '10' = 4 ms, '11' = 120 ms.
BANK[1:0] Code of the 0…3 port bank.

The counter reset registers are available by writing via the byte ports BAx+0x01, BAx+0x02, BAx+0x03 and have the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+01h** | Write | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| **BAx+02h** | Write | R15 | R14 | R13 | R12 | R11 | R10 | R9 | R8 |
| **BAx+03h** | Write | R23 | R22 | R21 | R20 | R19 | R18 | R17 | R16 |

R[23:0] Bits of counter reset. When a bit is set to '1', the CTx counter with the relevant number will be reset (zeroed).

Registers for controlling interrupts from counters are available **by writing** via the byte port with the address BAx+0Ch where the relevant BNK[1:0] code is set and have the following format:

| Address | BANK[1:0] | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+0Ch** | 0 | – | – | CTi [5] | CTi [4] | CTi [3] | CTi [2] | CTi [1] | CTi [0] |
| **BAx+0Ch** | 1 | – | – | CTi [11] | CTi [10] | CTi [9] | CTi [8] | CTi [7] | CTi [6] |
| **BAx+0Ch** | 2 | – | – | CTi [17] | CTi [16] | CTi [15] | CTi [14] | CTi [13] | CTi [12] |
| **BAx+0Ch** | 3 | – | – | CTi [23] | CTi [22] | CTi [21] | CTi [20] | CTi [19] | CTi [18] |

CTi [23:0] Bits that enable generation of interrupts in case of counter overflow. By default, all bits are reset to '0'.

Registers of interrupt states from counters are available **by reading** via the byte port with the address BAx+0Ch where the relevant BNK[1:0] code is set and have the following format:

| Address | BANK[1:0] | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+0Ch** | 0 | – | – | CTiF [5] | CTiF [4] | CTiF [3] | CTiF [2] | CTiF [1] | CTiF [0] |
| **BAx+0Ch** | 1 | – | – | CTiF [11] | CTiF [10] | CTiF [9] | CTiF [8] | CTiF [7] | CTiF [6] |
| **BAx+0Ch** | 2 | – | – | CTiF [17] | CTiF [16] | CTiF [15] | CTiF [14] | CTiF [13] | CTiF [12] |
| **BAx+0Ch** | 3 | – | – | CTiF [23] | CTiF [22] | CTiF [21] | CTiF [20] | CTiF [19] | CTiF [18] |

CTiF [23:0] Flags of interrupts in the event of overflow of the relevant counters. When an interrupt occurs, the corresponding bit is latched, in order to be able to reinstall it, it is necessary to sequentially reset to '0' and then reinstall the relevant bit CTi [x] to '1'.

Registers of CT0… CT23 counters states are available by reading via the word ports (16 bits) with the addresses BAx+00h … BAx+0Ah with the relevant installed BNK(1:0) code. A true value can only be obtained by reading the entire 16-bit value of the counter (least significant byte, then the most significant byte). If you set up a transfer to read the fair 32-bit value, you need to take additional steps.

| Address | BANK[1:0] | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+00h** | 0 | CT0[15:0] | | | | | | | |
| **BAx+02h** | 0 | CT1[15:0] | | | | | | | |
| **BAx+04h** | 0 | CT2[15:0] | | | | | | | |
| **BAx+06h** | 0 | CT3[15:0] | | | | | | | |
| **BAx+08h** | 0 | CT4[15:0] | | | | | | | |
| **BAx+0Ah** | 0 | CT5[15:0] | | | | | | | |
| **BAx+00h** | 1 | CT6[15:0] | | | | | | | |
| **BAx+02h** | 1 | CT7[15:0] | | | | | | | |
| **BAx+04h** | 1 | CT8[15:0] | | | | | | | |
| **BAx+06h** | 1 | CT9[15:0] | | | | | | | |
| **BAx+08h** | 1 | CT10[15:0] | | | | | | | |
| **BAx+0Ah** | 1 | CT11[15:0] | | | | | | | |
| **BAx+00h** | 2 | CT12[15:0] | | | | | | | |
| **BAx+02h** | 2 | CT13[15:0] | | | | | | | |
| **BAx+04h** | 2 | CT14[15:0] | | | | | | | |
| **BAx+06h** | 2 | CT15[15:0] | | | | | | | |
| **BAx+08h** | 2 | CT16[15:0] | | | | | | | |
| **BAx+0Ah** | 2 | CT17[15:0] | | | | | | | |
| **BAx+00h** | 3 | CT18[15:0] | | | | | | | |
| **BAx+02h** | 3 | CT19[15:0] | | | | | | | |
| **BAx+04h** | 3 | CT20[15:0] | | | | | | | |
| **BAx+06h** | 3 | CT21[15:0] | | | | | | | |
| **BAx+08h** | 3 | CT22[15:0] | | | | | | | |
| **BAx+0Ah** | 3 | CT23[15:0] | | | | | | | |

CTx[15:0] Counter value for channel IO [x], where x is a channel number from 0 to 23. In case of setting a transfer, counting will be carried out on an even channel (0, 2, 4, 6, 8, 10).

The firmware identifier is available for reading via byte ports with the addresses BA+0x0E, BA+0x0F and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+0Eh** | Read | 'c' | | | | | | | |
| **BAx+0Fh** | Read | SN [7:0] | | | | | | | |

'c' ASCII-code of the lowercase Roman letter 'c' (0x63).
SN[7:0] Decimal code of the scheme number (SN[7:0]=2).

**Table 2-26 – External connections for the “c02» firmware**

| Pin # | Designation | Function | Pin # | Designation | Function |
|---|---|---|---|---|---|
| 1 | IO(12) | Input CT12 | 2 | +5V_EXT | – |
| 3 | IO(13) | Input CT13 | 4 | IO(10) | Input CT10 |
| 5 | IO(14) | Input CT14 | 6 | IO(11) | Input CT11 |
| 7 | IO(15) | Input CT15 | 8 | IO(9) | Input CT9 |
| 9 | IO(23) | Input CT23 | 10 | IO(8) | Input CT8 |
| 11 | IO(21) | Input CT21 | 12 | IO(22) | Input CT22 |
| 13 | IO(16) | Input CT16 | 14 | IO(20) | Input CT20 |
| 15 | IO(18) | Input CT18 | 16 | IO(17) | Input CT17 |
| 17 | IO(19) | Input CT19 | 18 | IO(7) | Input CT7 |
| 19 | IO(0) | Input CT0 | 20 | IO(6) | Input CT6 |
| 21 | IO(1) | Input CT1 | 22 | IO(5) | Input CT5 |
| 23 | IO(2) | Input CT2 | 24 | IO(4) | Input CT4 |
| 25 | IO(3) | Input CT3 | 26 | GND | – |

A fragment of the example in C language for writing and reading the port is shown below:

```
#define  BA0           0x310 // Base address of internal control registers
#define  BA1         0xA110 // Base address of UNIO port
unsigned int CT0;      // Counter value variable
outportb (BA0+0x0F,0x02); // Choose the c02 configuration of the UNIO
port
outportb (BA1+0, 0x04); // prohibit counter transfers, debouncing code 0 (1.6 µs), BANK=0
outportb (BA1+1, 0x01); // reset the CT0 counter

inport (BA1+0,CT0); // read the value of CT0 counter
printf ("CT0: %0004Xh", CT0);         // display the 16-bit value of CT0 counter
```

### 2.8.13 Description of the UNIO port configuration option "q04"

8 PWM signal conditioners. This section describes the purpose of the UNIO port with the address BAx (0xA110), scheme code "q04".

The "q04" option is compatible with the q02 option for the UNIOxx family and makes it possible to implement eight PWM signal conditioners within a single FPGA matrix with the following possibilities:

- 12-bit accuracy of setting the half-periods of the PWM signal;
- Ability to connect each conditioner to one of the 3 port channels;
- 4 ranges of output frequencies:

1.25 kHz – 2.5 MHz (one bit of half-period = 100 nsec),

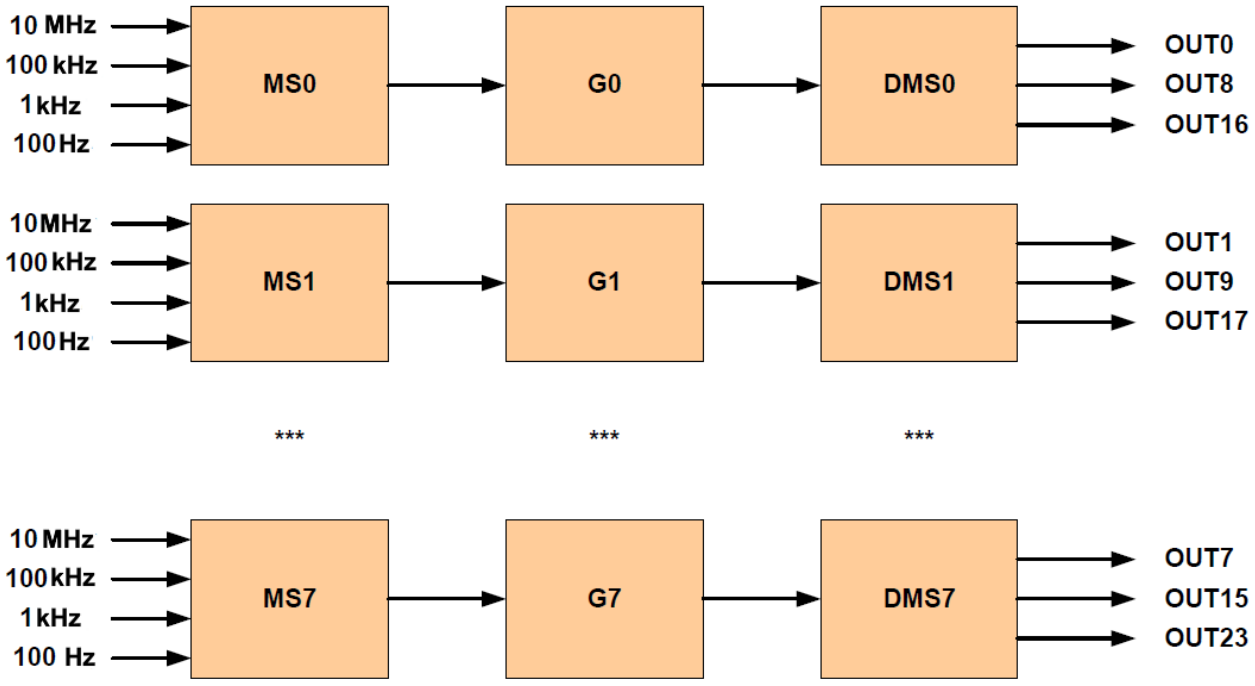
12.5 Hz – 25 kHz (one bit of half-period = 10 μs),

0.125 Hz – 250 Hz (one bit of half-period = 1 ms),

0.0125 kHz – 25 Hz (one bit of half-period = 10 ms).

The difference from the basic "q02" option from the set of UNIO firmware lies in increasing the number of PWM conditioners to 8 (the basic option had 6), implementing synchronization of the conditioners, using a fixed frequency with a period of 10 ms instead of an external reference frequency for generating PWM.

The scheme of one FPGA matrix of the "q04" option includes 12-bit PWM conditioners G0 ... G7, MS0 ... MS7 multiplexers for connecting the reference frequency and DMS0 ... DMS7 demultiplexers for connecting the conditioner to one of the port outputs.



Registers of PWM conditioners are designed to select the range, connect outputs and set the duration of half-periods. The registers are available by writing via word or byte ports with addresses BAx + 0… BAx + 0x0F.

| Address | Bits | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | … | 2 | 1 | 0 |
| **BAx+00h** | 0R[1:0] | | 0N[1:0] | | 0T[11:0] | | | | | |
| **BAx+02h** | 1R[1:0] | | 1N[1:0] | | 1T[11:0] | | | | | |
| **BAx+04h** | 2R[1:0] | | 2N[1:0] | | 2T[11:0] | | | | | |
| **BAx+06h** | 3R[1:0] | | 3N[1:0] | | 3T[11:0] | | | | | |
| **BAx+08h** | 4R[1:0] | | 4N[1:0] | | 4T[11:0] | | | | | |
| **BAx+0Ah** | 5R[1:0] | | 5N[1:0] | | 5T[11:0] | | | | | |
| **BAx+0Ch** | 6R[1:0] | | 6N[1:0] | | 6T[11:0] | | | | | |
| **BAx+0Dh** | 7R[1:0] | | 7N[1:0] | | 7T[11:0] | | | | | |

xR[1:0] Code of the Gx breaker range, where x=0…7 (permissible codes are from 0 to 3, see the range table).

xN[1:0] Code of Gx conditioner output, where x=0…7, defines one of the 3 outputs, through which a PWM signal will be transferred (codes from 0 to 2, see the table of external connections).

xT[11:0] Code of positive and negative half-period of the Gx conditioner, where x = 0 ... 7 (valid codes are from 1 to 4095, code 0 for prohibiting conditioning). The sequence of writing the positive and negative half-period codes is determined by the state of the half-period trigger in each conditioner (zeroing after power-on or reset). When the half-period trigger is reset, the duration of the positive half-period is recorded, when the half-period trigger is set, the duration of the negative half-period is recorded. Each writing to the register of Gx conditioner changes the state of the trigger's half-period to the opposite one (see example). The weight of the least significant bit is determined by the range code xR [1:0].

**Attention!**
The actual duration of the positive and negative half-period is always +1 more. To avoid incorrect operation of the PWM signal conditioners, it is not allowed to change only the least significant byte of the duration code.

The synchronization port of the generators is available for reading via the byte port with the address BA+0x0C. The general reset of generators is performed during this port is read. Reset of the generators is used for synchronizing the generators and does not affect the state of the outputs. When this register is read, triggers of half-periods of all the generators are reset.

The reset port of generators half-period triggers is available for reading via the byte port with the address BA+0x0D. When this register is read, triggers of half-periods of all the generators are reset.

**Table 2-27 – Ranges of the “q04” conditioner**

| Code xR[1:0] | Filling frequency | Period of filling frequency | Range of Gx conditioner output frequency |
|---|---|---|---|
| 0 | 10 MHz | 100 nsec | 1.25 kHz – 2.5 MHz |
| 1 | 100 kHz | 10 μs | 12.5 Hz – 25 kHz |
| 2 | 1 kHz | 1 ms | 0.125 Hz – 250 Hz |
| 3 | 100 Hz | 10 ms | 0.0125 kHz – 25 Hz |

The firmware identifier is available for reading via the byte ports with the addresses BA+0x0E, BA+0x0F and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+0Eh** | Read | 'q' | | | | | | | |
| **BAx+0Fh** | Read | SN [7:0] | | | | | | | |

'q' ASCII-code of the lowercase Roman letter 'q' (0x71).

SN[7:0] Decimal code of scheme number (SN[7:0]=4).

A fragment of an example of starting the G0 conditioner, the calculation of the period and duty cycle of the signal in the C language is shown below:

```
#define  BA0            0x310 // Base address of internal control registers
#define  BA1          0xA110 // Base address of UNIO port
 outportb (BA0+0x0F,0x03); // select the q04 configuration of UNIO
port
 unsigned int clk;     // 2-bit code of the filling frequency
unsigned int nch;      // 2-bit code of output channel
unsigned int Tp, Tm;  // 12-bit code of positive and negative half-period
float Freq,Q;
float scale[]={10000000.,100000.,1000.,100.};  // Filling frequency
inportb (BA+12,1);     // Resetting the trigger of half-period, if neccessary
outport (BA+0,(clk<<14)|(nch<<12)|Tp));  // Positive half-period of G0
outport (BA+0,(clk<<14)|(nch<<12)|Tm));  // Negative half-period of G0
Freq=1./((Tp+Tm+2)/scale[clk]);              // Frequency of output signal
Q=(float)(Tp+Tm+2)/(float)(Tp+1);     // Duty cycle of output signal
printf("Actually installed: F:%-9.4f Гц, Q:%-9.4f\n\n", Freq, Q);
```

**Table 2-28 – External connections for the “q04” firmware**

| Pin # | Function | Code of the xN[1:0] conditioner output | PWM conditioner | Pin # | Function | Code of the xN[1:0] conditioner output | PWM conditioner |
|---|---|---|---|---|---|---|---|
| 1 | IO(12) | 1 | G4 | 2 | +5V_EXT | – | – |
| 3 | IO(13) | 1 | G5 | 4 | IO(10) | 1 | G2 |
| 5 | IO(14) | 1 | G6 | 6 | IO(11) | 1 | G3 |
| 7 | IO(15) | 1 | G7 | 8 | IO(9) | 1 | G1 |
| 9 | IO(23) | 2 | G7 | 10 | IO(8) | 1 | G0 |
| 11 | IO(21) | 2 | G5 | 12 | IO(22) | 2 | G6 |
| 13 | IO(16) | 2 | G0 | 14 | IO(20) | 2 | G4 |
| 15 | IO(18) | 2 | G2 | 16 | IO(17) | 2 | G1 |
| 17 | IO(19) | 2 | G3 | 18 | IO(7) | 0 | G7 |
| 19 | IO(0) | 0 | G0 | 20 | IO(6) | 0 | G6 |
| 21 | IO(1) | 0 | G1 | 22 | IO(5) | 0 | G5 |
| 23 | IO(2) | 0 | G2 | 24 | IO(4) | 0 | G4 |
| 25 | IO(3) | 0 | G3 | 26 | GND | – | – |

### 2.8.14 Description of the UNIO port configuration option "t00"

4 x 16-bit timers with debouncing and programmable transfer.

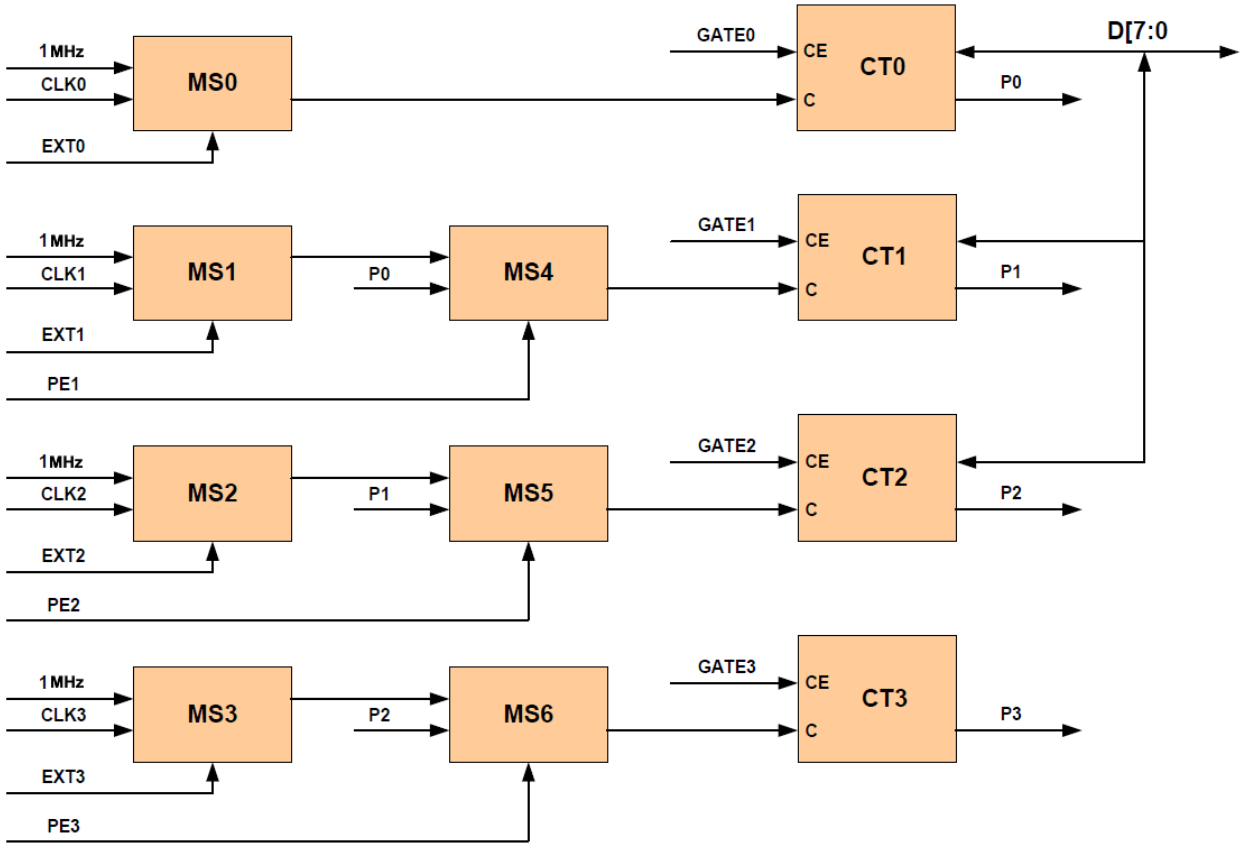
This section describes the assignment of the UNIO port with the address BAx, scheme code “t00”.

The option “t00” is compatible with option t00 for UNIOxx family and allows to implement four 16-bit timers within a single FPGA matrix with the following capabilities:

- Four 16-bit timers with the possibility of combining (up to 64 bits);
- Counting from the external frequency CLK0…CLK3 (debouncing 60 ns);
- Counting from the internal frequency of 1 MHz;
- External counting permission GATE0…GATE3;
- Generation of an interrupt when a preset value is reached (for any counter);
- Use of counter transfers for external connections P0… P3 (active level - 0).

The scheme of one FPGA matrix for the basic option “t00” includes 16-bit counters CT0 ... CT3 by module $n=2^0...2^{16}$; multiplexers MS0 ... MS3 for connecting an external (CLK0 ... CLK3) or internal frequency (1 MHz); multiplexers MS4 .. MS6 for connection of P0 .. P3 counter transfers. The external signals GATE0 ... GATE3 can be used to enable counters. The P0 ... P3 counters transfers can be used for external connections.

The block diagram of 2 counter channels is specified below.




The control register is available by writing through the port with the address BAx + 0 and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **BAx+01h** | Read | – | PE3 | PE2 | PE1 | MC3 | MC2 | MC1 | MC0 |
| **BAx+00h** | Write / Read | EXT3 | EXT2 | EXT1 | EXT0 | INT3 | INT2 | INT1 | INT0 |

PE3, PE2, PE1 Enabling counter transfers. When the PEx bit is set, the CTx counter counts the Px transfers of the CTx-1 counter (x = 1..3). If the bit is reset, the CTx counter is running on frequency in accordance with the setting of the EXTx bit. These bits are used to increase the timer capacity to 32/48/64 bits.

MC3…MC0 Maximum match flags. The flags are set when the counters reach their maximum values. The flags are reset by writing '0' to the required bits. The match flags are latched, therefore you must reset the corresponding bit of the flag prior to use it again.

EXT3…EXT0 The bits of using the external frequency for counters CT0..CT3. When the bits are set, the counters use the external frequency. If the bits are reset – the internal frequency of 1 MHz.

INT3…INT0 Enabling generation of interrupts. Setting the bits enables the generation of interrupts from the CT0… CT3 counters, resetting a bit - disables. Interrupt signals appear when the states of the counters match with the maximum values written in the registers MAXC0 ... MAXC3. The interrupts are confirmed by software by resetting the corresponding flags MC3...MC0.

The MAXC0…MAXC3 registers are available by writing through the word ports BAx + 0x02, BAx + 0x04, BAx + 0x06, BAx + 0x08. These 16-bit registers store the maximum values (or the counting module of the counters), upon reaching which the counters are reset and the flags MC0 ... MC2 are set.

| Address | Action | Bits | | | | |
|---|---|---|---|---|---|---|
| | | 15 | 14 | … | 1 | 0 |
| BAx+02h | Write | MAXC0[15:0] | | | | |
| BAx+04h | Write | MAXC1[15:0] | | | | |
| BAx+06h | Write | MAXC2[15:0] | | | | |
| BAx+08h | Write | MAXC3[15:0] | | | | |

**Attention!** When the most significant byte is written to the MAXCx register, the corresponding CTx counter is reset.

The state registers of the counters CT3…CT0 are available **for reading** via the word ports with the addresses BAx+0x02, BAx+0x04, BAx+0x06, BAx+0x08. They allow to count 16-bit states of timers at any time.

| Address | Action | Bits | | | | |
|---|---|---|---|---|---|---|
| | | 15 | 14 | … | 1 | 0 |
| BAx+02h | Read | CT0[15:0] | | | | |
| BAx+04h | Read | CT1[15:0] | | | | |
| BAx+06h | Read | CT2[15:0] | | | | |
| BAx+08h | Read | CT3[15:0] | | | | |

**ATTENTION!** The value of *CTx equal to MAXCx* can never be read, since when the maximum value is reached, the counters will change to the zero state.

The firmware identifier is available for reading via the byte ports with the addresses BA+0x0E, BA+0x0F and has the following format:

| Address | Action | Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BAx+0Eh | Read | 't' | | | | | | | |
| BAx+0Fh | Read | SN [7:0] | | | | | | | |

't' ASCII- code of the lowercase Roman letter 'c' (0x74).

SN[7:0] Decimal code of scheme number (SN[7:0]=0).

**Table 2-29 - Table of external connections for “t00” firmware**

| Pin # | Function | Purpose | Pin # | Function | Purpose |
|---|---|---|---|---|---|
| 1 | IO(12) | – | 2 | +5V_EXT | – |
| 3 | IO(13) | – | 4 | IO(10) | GATE3 |
| 5 | IO(14) | – | 6 | IO(11) | P3 |
| 7 | IO(15) | – | 8 | IO(9) | CLK3 |
| 9 | IO(23) | – | 10 | IO(8) | P2 |
| 11 | IO(21) | – | 12 | IO(22) | – |
| 13 | IO(16) | – | 14 | IO(20) | – |
| 15 | IO(18) | – | 16 | IO(17) | – |
| 17 | IO(19) | – | 18 | IO(7) | GATE2 |
| 19 | IO(0) | CLK0 | 20 | IO(6) | CLK2 |
| 21 | IO(1) | GATE0 | 22 | IO(5) | P1 |
| 23 | IO(2) | P0 | 24 | IO(4) | GATE1 |
| 25 | IO(3) | CLK1 | 26 | GND | – |

# 3 Installation and configuration

The device is made in PC/104-plus format. Expanding the functionality of the device is possible by connecting the additional PC/104 and PC / 104-plus expansion modules.

## 3.1 Setting the switches

For hardware configuration of the device, the switches are used, which general description is given in Table 3-1.

**Table 3-1 – Purpose of switches for device configuration**

| Switches | Description |
|---|---|
| SW1 | Reset button |
| X1 | Device reset by the watchdog timeout of the power supervisor (1.6 s). |
| X2, X3 | Connecting the remote reset signal (XP12 connector) to the hardware line device reset (X2 [1-2]), interrupt line (X2 [2-3]) or processor's non-maskable interrupt line (X2 [2-3] + X3 [1-2]).<br>Connecting the signal for lowering the input supply voltage below 4.6 V to the interrupt line (X2 [3-4]) or the processor's non-maskable interrupt line (X2 [3-4] + X3 [1-2]). |
| X4 | Reset of BIOS Setup settings ( X4[1-2] ).<br>Protection of the installed BIOS Setup user settings (X4 [3-4]). |
| X6 | COM3 port: connection of biasing resistor for 680 Ohm and 120 Ohm terminator, selection of half-duplex/full duplex operation mode |
| X7 | COM4 port: connection of biasing resistor for 680 Ohm and 120 Ohm terminators, selection of half-duplex/full duplex operation mode. |
| X20 | Binding the levels of UNIO (IO_A, IO_B) port lines. |

Description of X6, X7 and X20 switches is given in the relevant sections of the paragraph "Structure and functioning".

## 3.2 Configuring device parameters (BIOS SETUP)

Device configuration parameters are stored in internal non-volatile ferromagnetic random-access memory (FRAM) and can be changed in BIOS Setup.

The device configuration parameters are set during booting, when the **<Del>** key is pressed on the keyboard connected to the USB port, by **<F4>** key on the keyboard of the remote terminal, when the device is connected via the console serial COM-port.

# 4 Software

## 4.1 Basic software

Upon delivery, the integrated flash drive of the device contains programs that ensure the device's readiness for use: the built-in FreeDOS operating system.

The latest versions of documentation, BIOS and utilities can be downloaded from the manufacturer's and distributor's ftp servers.

## 4.2 Establishing connection between PC and the device

To establish a connection between a personal computer (PC) and the device, you must:

1. When powered off, connect the VTC-9F cable with the null-modem adapter to the PC's COM port and the COM1 or COM2 connector of the device (by default, the COM1 port is set as the console port).
2. Install the terminal software suite supporting the data communications protocol XMODEM/CRC (e.g., HYPERTERMINAL, TELEMAX, TERM90, TERM95, PUTTY), with the parameters of serial line communication:
    1. PC port (COM1 / COM2)
    2. 8 bit data
    3. 1 stop bit
    4. No parity check
    5. Data exchange rate 115,200 bit/sec.
3. Turn on the power or press the RESET button if steps 1, 2 are not required and the power is on. If the connection is successfully established after booting the operating system, a DOS prompt line will appear on the PC screen: **C:>**
4. To boot the operating system without executing the commands of the **CONFIG.SYS** and **AUTOEXEC.BAT** files, after turning on the power or RESET, press the following key combination <Ctrl-B> or <Ctrl-C> for the step-by-step execution of commands.

## 4.3 Operation of the device with AT keyboard and VGA monitor

When connecting a USB keyboard and a VGA monitor to the device, the CPC316 controller can be used as a regular AT(x86)-compatible computer. Launching and debugging programs in this case is performed as usual and is not covered in this UM.

## 4.4 BIOS SOC Vortex86DX3 interface for reading serial number

To store the device parameters, the FRAM array is used. For storing the entire system information, the size of 1Kbyte is used (might slightly differ depending on BIOS version).

### 4.4.1 Array structure

```
_FRAM     STRUCT

          db        256 dup (0)        ; Reserve for storing the CMOS
copy

dSerNum dd          0                  ; Product number

_FRAM     ENDS
```

Service interrupt combined with printer service interrupt.

To call the service, the **INT 17H** interrupt is used with a parameter in the AH = 0ADh register.

The values of other parameters passed in processor registers are shown below.

Where the function number (AL) is specified incorrectly, AX = -1 (0FFFFh) is returned.

### 4.4.2 Obtaining serial number of the device

Input: AL = 6

Output: AX = result code (0 – no error)

CX:DX = serial number.

## 4.5 SOC Vortex86DX3 BIOS interface for reading/writing to FRAM

Integrated FRAM is also available for storing user data.

To call the FRAM read/write service, the **INT 17H** interrupt is used with a parameter in the AH = 0ADh register.

The values of other parameters transferred in processor registers are shown below. Where the function number (AL) is specified incorrectly, the AX = -1 (0FFFFh) is returned.

### 4.5.1 Reading user data from FRAM

Input: AL = 0Ch

BX = data start address in FRAM user area

CX = number of bytes read

DS:DX = indicates to the buffer for reading

Output:

AX = result code (0 – no error, -2 (0FFFEh) – parameter error, invalid address)

BX = maximum allowed address (size of the user area -1)

CX = number of bytes actually read

This function reads the specified bytes of the FRAM user area into the calling program's buffer.

### 4.5.2 Writing user data to FRAM

Input: AL = 0Dh

BX = data start address in FRAM user area

CX = number of bytes written

DS:DX = indicates to the written data

Output:

AX = result code (0 – no error, -2 (0FFFEh) – parameter error, invalid address)

BX = maximum allowed address (size of the user area -1)

CX = number of bytes actually written

* this function writes data to the FRAM user area.

## 4.6 Service programs

This chapter describes a set of drivers for working with I/O equipment connected to the device.

### 4.6.1 BIOS update utilities

The **ANYBIOS.EXE** program is designed for modifying the BIOS with writing to the integrated SPI-Flash of the processor in the device.

To modify the BIOS, run the program with the "**w**" key and specify the BIOS file name "**bios.bin**" as a parameter and the key for skipping the MAC address recording of the Ethernet controller integrated into the processor:

**anybios.exe w bios.bin skipmac**

### 4.6.2 CMOS_RST.EXE utility (remote reset of BIOS settings)

The **CMOS_RST.EXE** is designed for resetting the BIOS settings to the default state (similar to the action of the BIOS Setup "Load Optimal Defaults" item). To reset the settings using the **CMOS_RST.EXE** program, you should connect the COM1 port or COM2 of the device of the product with a PC COM port with a null modem cable and turn on the power supply of the product (the settings will be reset and written to CMOS and FRAM, then a hardware reset will be

automatically carried out and the controller will start with the default settings). Windows OS must be installed on the PC you are using.

Syntax:

**cmos_rst.exe [COM]** where **[COM]** is the number of the COM-port used in the PC, this would be COM1 by default.

## 4.7 BIOS Update

The BIOS is stored in flash memory integrated into the Vortex86DX3 SoC and connected to the SPI interface.

During the BIOS update, please note that after the image update and reboot, the optimal (factory) BIOS Setup settings will be loaded. In this case, the console I/O settings will be changed to the factory settings (the Redirection After BIOS POST = "Boot Loader" mode). Therefore, if you need to use the integrated console I/O, then each time you boot the device during the BIOS update procedure, you must enter BIOS Setup and select the required settings for the console I/O and BIOS in general.

## 4.8 Switching the reference frequency for COM1, COM2, COM3, COM4

The default reference frequency for the Vortex86DX3 SoC integrated serial ports COM1, COM2, COM3, COM4 is set to 1.8432 MHz (24 MHz/13). It is also possible to set the reference frequency to 24 MHz.
Setting the reference clock frequency for COM1..4 is available through the registers in PCI Config Space Bus: Dev: Func - 00.07.00 (0x00003800)

UART1 - Reg 0x50 bit 30 ( UART clock selection. 0: 24MHz/13 (default), 1: 24MHz )
UART2 - Reg 0xA0 bit 22 ( UART clock selection. 0: 24MHz/13 (default), 1: 24MHz )
UART3 - Reg 0xA4 bit 22 ( UART clock selection. 0: 24MHz/13 (default), 1: 24MHz )
UART4 - Reg 0xA8 bit 22 ( UART clock selection. 0: 24MHz/13 (default), 1: 24MHz )

**Switching example (DOS, WatcomC or BorlandC):**

```
#include <conio.h>
#include <dos.h>
void Set_Base24MHz_UART1()
{
        uint32_t pci_reg;
        disable();                          // disable interrupts
        outpd( 0xCF8, 0x80003850 );         // pci_cfg_index
        pci_reg = inpd( 0xCFC );            // read
        pci_reg = pci_reg | 0x40000000;     // set BIT30 (UART1)
       // pci_reg = pci_reg | 0x00400000;   // set BIT22 (UART2..4)
        outpd( 0xCF8, 0x80003850 );         // pci_cfg_index
```

```
        outpd( 0xCFC, pci_reg );            // write register
        enable();                           // enable interrupts
}
```

## 4.9 Working with I2C devices

As an example of working with i2c devices, you can use the vortex86_i2c library (library files "vortex86_i2c.c", "vortex86_i2c.h" and sample files are available on the manufacturer's and distributor's ftp servers.

The library is collected within a free Open Watcom C / C ++ package.

The library implements the following functions:

- uint16_t **I2C_GetBase**() – returns the base address in the I/O space used for i2c buses
- void **I2C_SetBase**( uint16_t ba ) – sets a base address in the I/O space that is used for i2c buses
- void **I2C_PowerOff**( uint8_t channel ) – disables the specified i2c bus
- void **I2C_PowerOn**( uint8_t channel ) – enables the specified i2c bus
- uint16_t **I2C_Init**( uint8_t channel ) – configures the specified bus and returns the base address
- uint16_t **I2C_Start**( uint8_t channel, uint8_t addr, uint8_t gen_stop, uint16_t timeout ) – generates the start on the bus and sets the device address. The function will return the remainder of the timeout parameter
- uint16_t **I2C_ReadByte**( uint8_t channel, uint8_t * dat, uint8_t lastbyte, uint16_t timeout ) – reads one byte. For the last byte, the lastbyte parameter must not be equal to 0. The function returns the remainder of the timeout parameter
- uint16_t **I2C_WriteByte**( uint8_t channel, uint8_t dat, uint8_t lastbyte, uint16_t timeout ) – writes one byte. For the last byte, the lastbyte parameter must not be equal to 0. The function returns the remainder of the timeout parameter
- uint16_t **I2C_ReadByteReg**( uint8_t channel, uint8_t addr, uint8_t reg, uint8_t * dat, uint16_t timeout ) reads a byte from the i2c device with the addr address from register reg. The function will return the remainder of the timeout parameter
- uint16_t **I2C_ReadWordReg**( uint8_t channel, uint8_t addr, uint8_t reg, uint16_t * dat, uint16_t timeout ) reads a word from i2c device with the addr address from register reg. The function will return the remainder of the timeout parameter
- uint16_t **I2C_WriteByteReg**( uint8_t channel, uint8_t addr, uint8_t reg, uint8_t dat, uint16_t timeout ) writes a byte to the i2c device with the addr address to register reg. The function will return the remainder of the timeout parameter
- uint16_t **I2C_WriteWordReg**( uint8_t channel, uint8_t addr, uint8_t reg, uint16_t dat, uint16_t timeout ) writes a word to i2c device with the addr address to register reg. The function will return the remainder of the timeout parameter

### Example of working with the LM92CIM digital temperature sensor (National Semiconductor)

First, the bus is configured

```
I2C_SetBase( I2C_DEF_BASE_ADDR );

I2C_Init( I2C_CHANNEL0 );
```

Next, the microcircuit’s identifier is read

```
timeout = I2C_ReadWordReg( I2C_CHANNEL0, LM92_WR_ADDR, 0x07, &id, DEF_TIMEOUT );
```

If the identifier is correct, the temperature will be read

```
timeout = I2C_ReadWordReg( I2C_CHANNEL0, LM92_WR_ADDR, 0x00, &rd_temp, DEF_TIMEOUT
);
```

Converted to the usual form and routed to the console

```
rd_temp = rd_temp >> 3;
if( rd_temp & 0x2000 ) {
        temp = -1;
        rd_temp = 0x2000 - rd_temp;
}
temp = temp * rd_temp * 0.0625;
printf( "  Temperature=%.4f\r\n", temp );
```

**Example of working with the MCP23009-E/MG IC, on which the GPIO port (XP1) is implemented**

First, the bus is configured,

```
#define MCP23009_ADDR          0x40
#define DEF_TIMEOUT            200
…
I2C_SetBase( I2C_DEF_BASE_ADDR );
I2C_Init( I2C_CHANNEL0 );
```

Reading the status of the MCP23009-E/MG register, where “reg” is the register address from 0x00h to 0x0Ah in accordance with the documentation for MCP23009-E/MG, the status of the inputs is written to the “data” variable

```
timeout = I2C_ReadWordReg( I2C_CHANNEL0, MCP23009_ADDR, reg, &data, DEF_TIMEOUT );
```

Writing to the MCP23009-E/MG register

```
timeout = I2C_WriteByteReg( I2C_CHANNEL0, MCP23009_ADDR, reg, data, DEF_TIMEOUT );
```

For example, setting all the lines as inputs and reading their status

```
timeout = I2C_WriteByteReg( I2C_CHANNEL0, MCP23009_ADDR, 0x00, 0xFF, DEF_TIMEOUT );
timeout = I2C_ReadWordReg( I2C_CHANNEL0, MCP23009_ADDR, 0x09, &data, DEF_TIMEOUT );
```

For example, setting the “open collector” output type, setting the state of the GPIO (7:4) lines as '1', GPIO (3:0) - as '0', and setting all the lines as outputs

```
timeout = I2C_WriteByteReg( I2C_CHANNEL0, MCP23009_ADDR, 0x05, 0x04, DEF_TIMEOUT );
timeout = I2C_WriteByteReg( I2C_CHANNEL0, MCP23009_ADDR, 0x09, 0xF0, DEF_TIMEOUT );
timeout = I2C_WriteByteReg( I2C_CHANNEL0, MCP23009_ADDR, 0x00, 0x00, DEF_TIMEOUT );
```

# 5 Basic Input / Output System (BIOS)

To enter BIOS Setup, while booting the system during the POST (Power On Self Test) procedure, press the DEL key on the keyboard or the F4 key on the console PC keyboard (when the Console Redirect option is enabled). An example of the screen during the POST procedure is shown in Fig. 5-1.

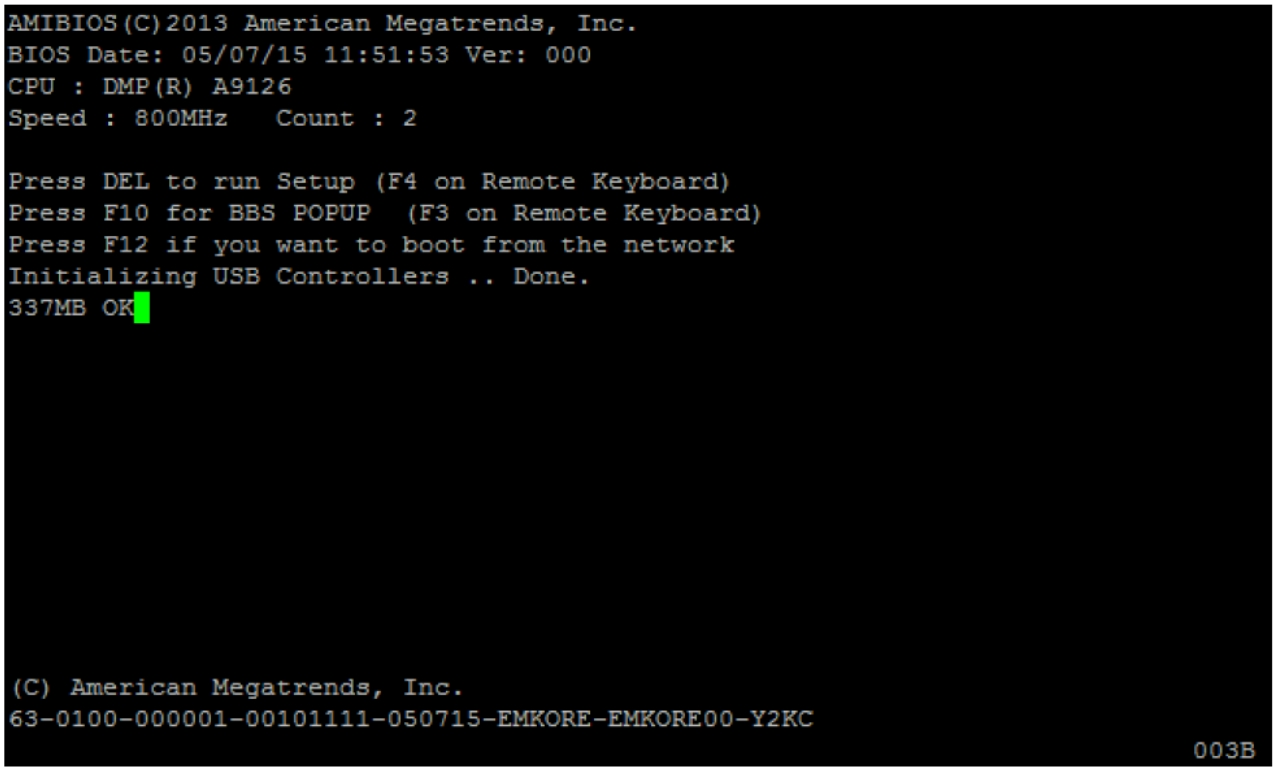



**Fig. 5-1 – Screen during the device boot (POST)**

Using the BIOS Setup Utility, you can change the BIOS (Basic Input Output System) parameters and control special modes of the device. This program uses the menu system to make changes and to enable or disable special features.

The information fields (highlighted in gray) are used to display additional information on the device and/or its settings and are not available for changes by the user. Default values are underlined when describing menu items. Information fields are italicized. Setting incorrect values can lead to system malfunctions.

**Attention!**

To reset BIOS Setup settings where it is impossible to enter the settings menu, you must use the cmos_rst.exe utility for remote reset via the COM port (COM1, COM2).

## 5.1 Main

This tab contains description of BIOS version, RAM and CPU installed. There are also two items responsible for setting the current time and date. The Main menu screen and item descriptions are shown below.

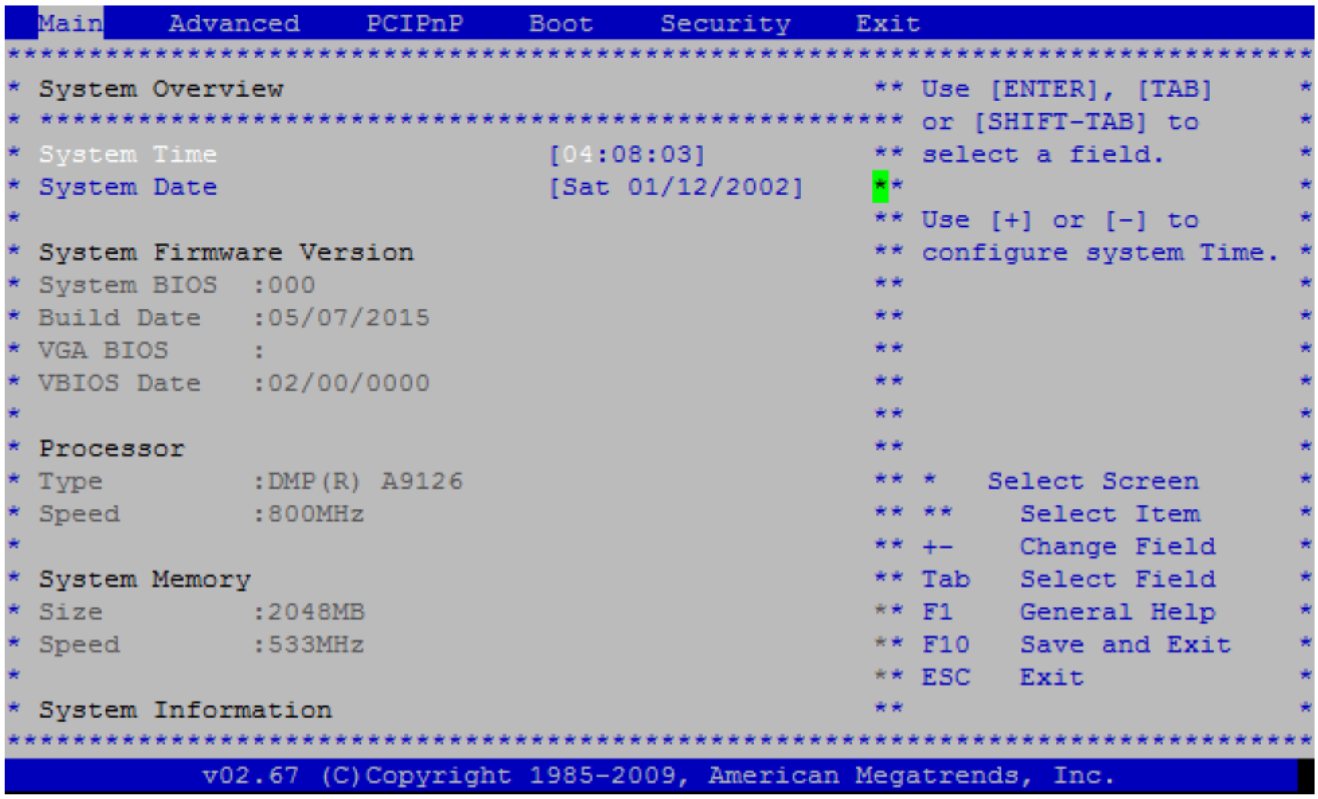



**Fig. 5-2 – Screen of the "Main» menu**

**Table 5-1 – Description of the "Main" menu**

| Menu item | Purpose |
|---|---|
| ***System BIOS***<br>*(BIOS version information)* | *Version*<br>*Build Date*<br>*ID* |
| ***VGA BIOS*** | *Version*<br>*Build Date*<br>*ID* |
| ***Fastwel FPGA Firmware***<br>*(Versions and IDs of integrated FPGAs firmware)* | *XC6S Rev.* – version of the system FPGA<br>*XC6S ID* – ID of the UNIO port firmware |
| ***Processor***<br>*(information field)* | Information related to the CPU installed:<br>*Vortex A9126* – version of the Vortex86DX3 processor<br>*Speed* – processor clock speed |

| | |
|---|---|
| ***System Memory***<br>*(information field)* | Information on the DDR2 SDRAM installed:<br>*Size*<br>*Speed* |

| Menu item | Purpose |
|---|---|
| System Time | Current time in format [hour/min/sec] |
| System Date | Current date in format [month/day/year] |

## 5.2 Advanced

This tab contains the items responsible for the operation of the soldered ATA Flash Disk, processor Cache memory, IDE bus, console I/O and USB devices. The “Advanced” menu screen and item descriptions are shown below.

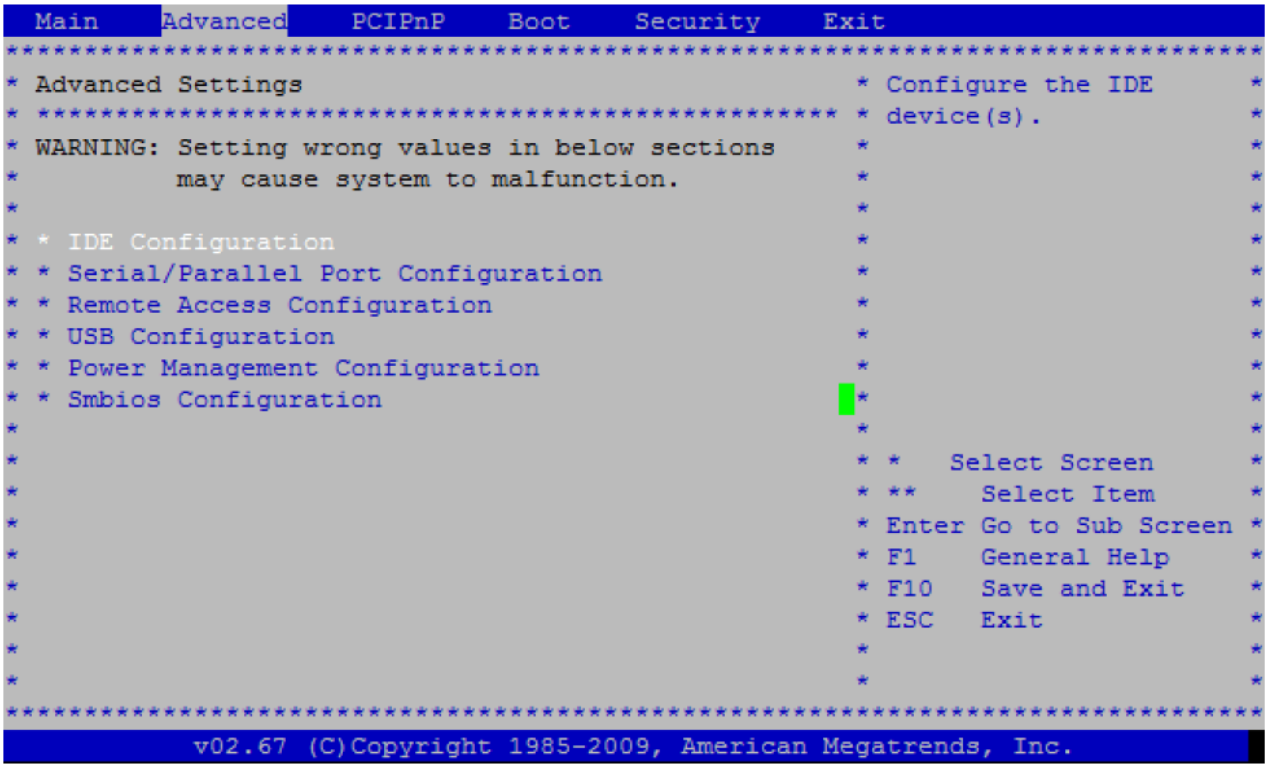



**Fig. 5-3 – Screen of the “Advanced” menu**

**Table 5-2 – Descirption of the “Advanced” menu**

| Menu item | Purpose |
|---|---|
| **IDE Configuration** (submenu) | Controlling the operation of devices on the bus IDE/SATA/SD |
| **Serial / Parallel Port Configuration** (submenu) | Serial and parallel port settings |

| | |
|---|---|
| **Remote Access Configuration** (submenu) | Settings of the console I/O |
| **USB Configuration** (submenu) | Settings of USB ports. These settings apply to all 4 x USB ports |
| **Menu item** | **Purpose** |
| **Power Management Configuration** (submenu) | ACPI, APM settings |
| **Smbios Configuration** (submenu) | SMBIOS settings |

### 5.2.1 IDE Configuration

Screen of the “IDE Configuration” menu and description of items are given below.

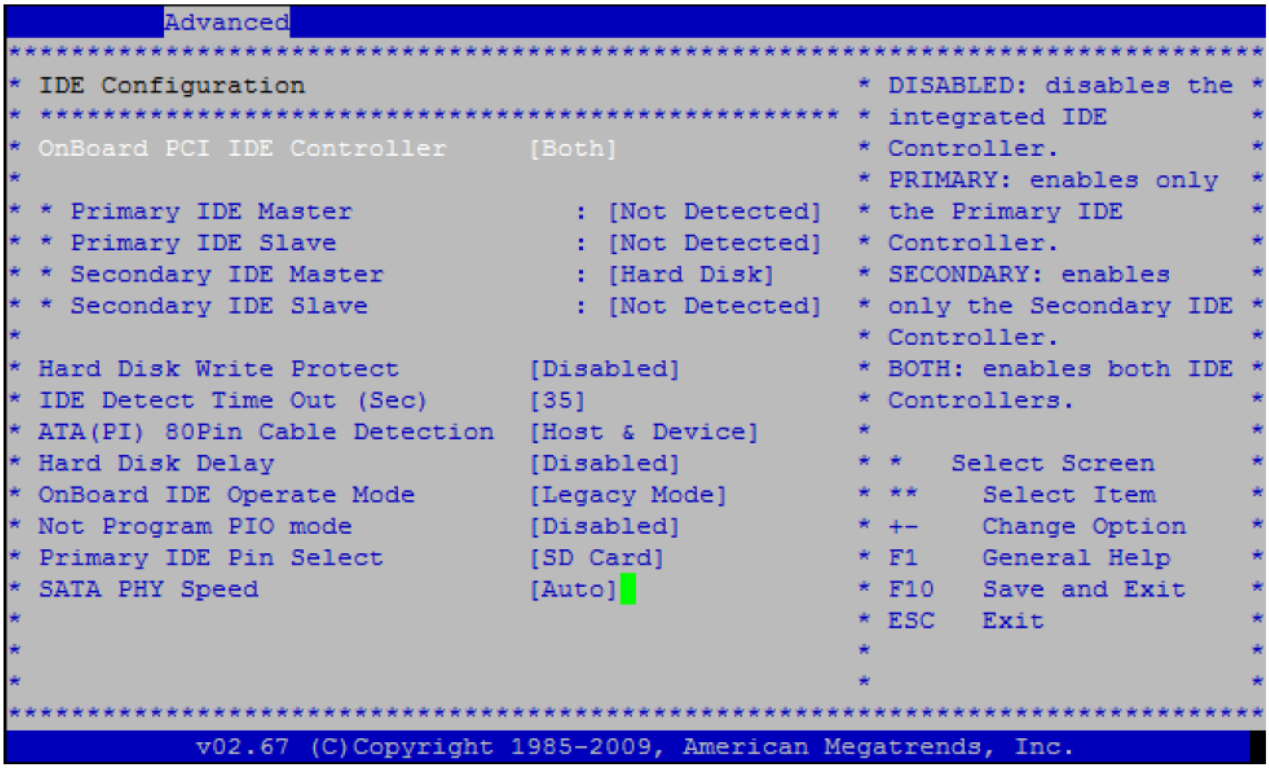



**Fig. 5-4 – Screen of the “IDE Configuration” menu**

**Table 5-3 – Description of the “IDE Configuration” menu**

| Menu item | Purpose | |
|---|---|---|
| **Onboard PCI IDE Controller** | Managing operation of the integrated PCI IDE bus controller. | |
| | [Both] | Operation is enabled |
| | [Disabled] | Operation is disabled |
| **Primary IDE Master** (submenu) | This section stores information on the connected IDE device, working in Master mode if a microSD card is installed in the slot. | |

| | | |
|---|---|---|
| **Primary IDE Slave** (submenu) | Not used. | |
| **Secondary IDE Master** (submenu) | This section stores information on the connected IDE device, working in the Slave mode. Integrated SATA drive for 8 GB is connected. | |
| **Menu item** | **Purpose** | |
| **Secondary IDE Slave** (sumenu) | Not used. | |
| Hard Disk Write Protect | Permission for setting the protect for writing to IDE devices | |
| | [Enabled] | Setting the protect |
| | [Disabled] | Remove the protect |
| IDE Detect Time Out (Sec) | Waiting limit for determining ATA/ATAPI device, in seconds. Available values are: | |
| | [0], [5], [10], [15], [20], [25], [30] ,[35] | |
| ATA(PI) 80Pin Cable Detection | Selecting a Method for identifying an 80-core ATA(PI) cable | |
| | [Host & Device] | Check by system and IDE devices |
| | [Host] | Check only by the system |
| | [Device] | Check only by IDE devices |
| Hard Disk Delay | Additional delay when identifying IDE devices | |
| | [Disabled] | Disabled |
| | [1], [2], [4], [8] | Time for additional delay in seconds |
| Onboard IDE Operate Mode | IDE controller operation mode | |
| | [Legacy] | Legacy mode is selected |
| | [Native] | Native mode is selected (only for Windows XP and 7) |

### 5.2.1.1 Secondary IDE Master

The screen of the “Primary IDE Master” and description of the items are given below. The “IDE Primary Master and Slave” menus are fully identical to the “Secondary IDE Master” menu.

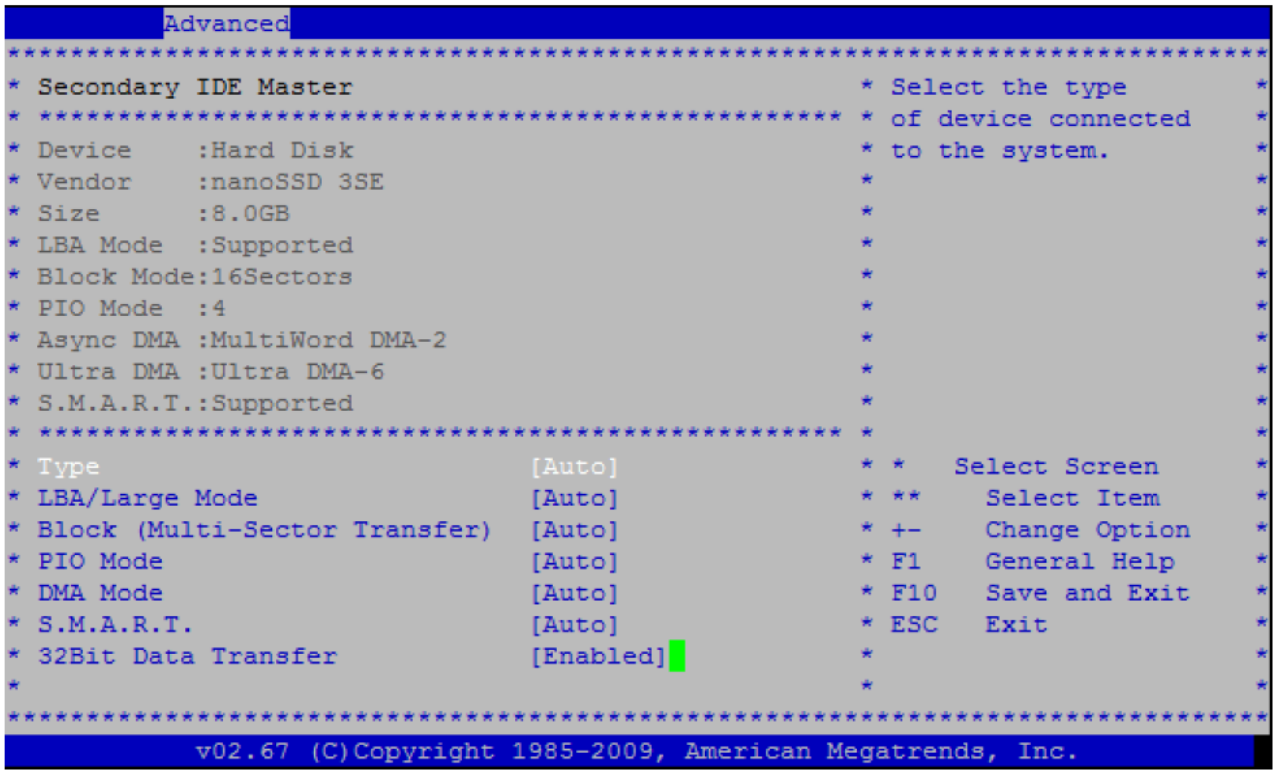



**Fig. 5-5 – Screen of the “Secondary IDE Master” menu**

**Table 5-4 – Description of the “Primary IDE Master” menu**

| Menu item | Purpose | |
|---|---|---|
| Type | Type of the device, connected to this IDE channel | |
| | [Not Installed] | Prohibition against the search for connected devices |
| | [Auto] | Automatic detection of the connected device type |
| | [CD/DVD] | Identify the connected device as the CD/DVD drive |
| | [ARMD] | Identify the connected device as the ATAPI removable media (ZIP, LS-120) |
| LBA/Large Mode | Addressing type of the device connected to this IDE channel | |
| | [Auto] | Automatic detection of LBA mode support |
| | [Disabled] | Prohibition against LBA mode detection, the Large Mode used |
| Block (Multi-Sector Transfer) | Block data transfer mode | |
| | [Auto] | This option allows the BIOS to automatically detect whether the Multi-Sector Transfers mode is supported on the current channel. This option allows the BIOS to automatically detect the amount of sectors per block for transfer from hard drive to memory. Data to/from the device will be transmitted several sectors per unit of time.<br>The default value. |

| Menu item | Purpose | |
|---|---|---|
| | [Disabled] | This option disables BIOS from using the Multi-Sector Transfer mode on the current channel. Data to/from the device will be transmitted one sector per unit of time. |
| PIO Mode | Programmed I/O (PIO) mode | |
| | [Auto] | This option allows BIOS to automatically determine if the device supports the PIO mode. This setting is recommended to be used when it is impossible to determine the supported mode of the connected device. |
| | [0] | Set the PIO 0 mode for the connected device. The data transfer rate in this mode is up to 3.3 MB/sec. |
| | [1] | Set the PIO 1 mode for the connected device. The data transfer rate in this mode is up to 5.2 MB/sec. |
| | [2] | Set the PIO 2 mode for the connected device. The data transfer rate in this mode is up to 8.3 MB/sec. |
| | [3] | Set the PIO 3 mode for the connected device. . The data transfer rate in this mode is up to 11,1. MB/sec. |
| | [4] | Set the PIO 4 mode for the connected device. The data transfer rate in this mode is up to 16.6. MB/sec. |
| DMA Mode | Data transfer mode DMA (Direct Memory Access) | |
| | [Auto] | Recommended value for the most efficient data transfer. BIOS will automatically detect the most suitable DMA mode. |
| | [SWDMA0]<br>[SWDMA1]<br>[SWDMA2] | Single Word DMA modes |
| | [MWDMA0]<br>[MWDMA1]<br>[MWDMA2] | Multi Word DMA modes |
| S.M.A.R.T. | Smart Monitoring, Analysis, and Reporting Technology | |
| | [Auto] | BIOS will automatically detect and support the connected device. It is recommended to use this option if detection and support of the connected drive is impossible. |
| | [Enabled] | This option enables BIOS to use the SMART function when working with the connected drives |
| | [Disabled] | This option disables BIOS from the use of SMART function when working with the connected drives. |
| 32-bit Data Transfer | 32-bit data transfer mode | |
| | [Enabled] | This option enables the use of 32-bit data transfer for the connected device. |
| | [Disabled] | This option disables the use of 32-bit data transfer for the connected device. |

### 5.2.2 Remote Access Configuration

Screen of the “Remote Access Configuration” menu and description of the items are given below.

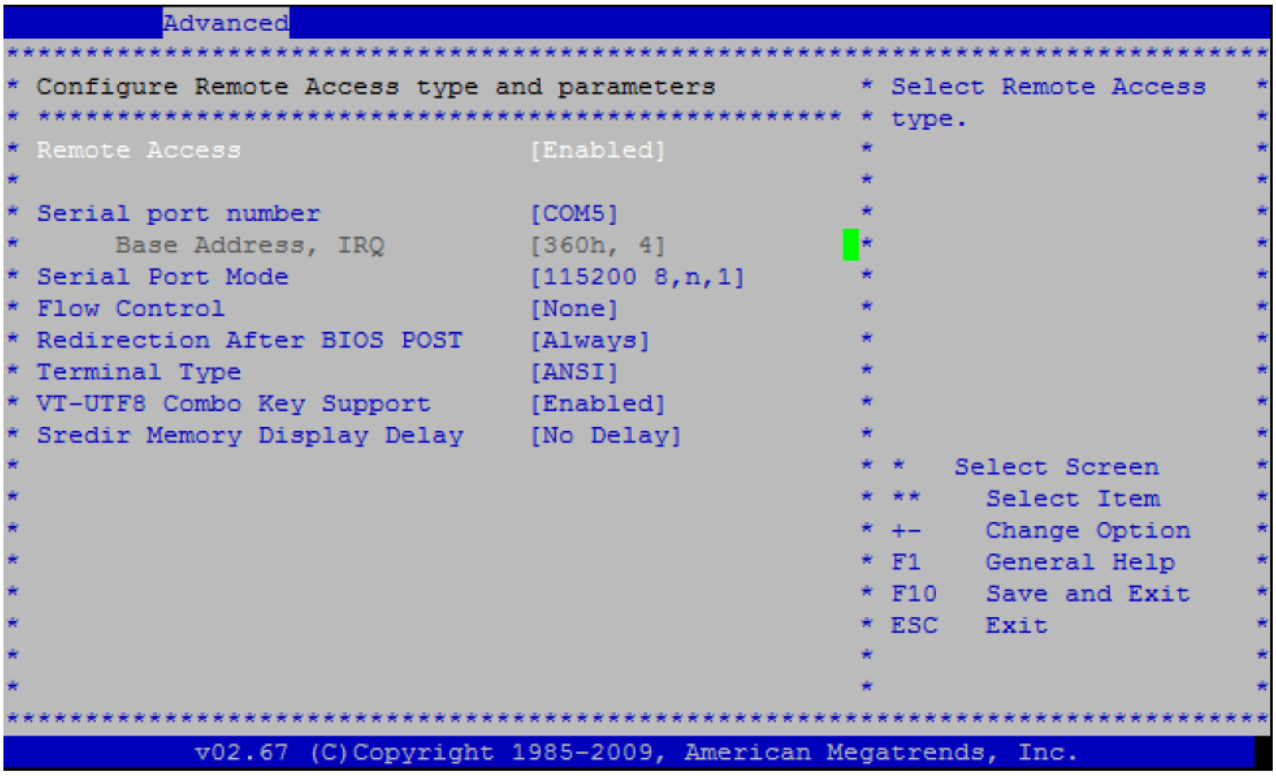



**Fig. 5-6 – Screen of the “Remote Access Configuration” menu**

**Table 5-5 – Description of the “Remote Access Configuration” menu**

| Menu item | Purpose | |
|---|---|---|
| **Remote Access** | Console I/O | |
| | [Disabled] | Console I/O is disabled |
| | [Enabled] | Console I/O is enabled, additional options for configuring console I/O parameters become available. |
| **Serial port number** | Selecting the console I/O serial port | |
| | [COM1] | COM1 port is used as a console I/O port |
| | [COM2] | COM2 is used as a console I/O port |
| | [COM3] | COM3 is used as a console I/O port |
| | [COM4] | COM4 port is used as a console I/O port |
| **Serial port mode** | Operation mode of the console I/O port | |
| | [115200 8,n,1], | Data transfer rate 115.2 kbaud, 8-bit, no parity check, 1 stop bit |
| | [57600 8,n,1], | Data transfer rate 57.6 kbaud, 8-bit, no parity check, 1 stop bit |
| **Menu item** | **Purpose** | |
| | [38400 8,n,1], | Data transfer rate 38.4 kbaud, 8-bit, no parity check, 1 stop bit |
| | [19200 8,n,1], | Data transfer rate 19.2 kbaud, 8-bit, no parity check, 1 stop bit |
| | [09600 8,n,1], | Data transfer rate 9.6 kbaud, 8-bit, no parity check, 1 stop bit |

| | | |
|---|---|---|
| **Flow Control** | Controlling the stream of characters for console port | |
| | [None] | No |
| | [Hardware] | CTS/RTS hardware control |
| | [Software] | XON/XOFF software control |
| **Redirection After BIOS POST** | The mode of the console I/O port operation after passing the POST procedure by the BIOS | |
| | [Disabled] | Disabling the console I/O after passing POST by BIOS |
| | [Boot Loader] | Console I/O is active during the time BIOS passes the POST procedure and at the time of OS system booting |
| | [Always] | Console I/O is always active. Some operating systems may not work while this option is selected. |
| **Terminal Type** | Type of the terminal | |
| | [ANSI] | ANSI standard |
| | [VT100] | VT100 standard |
| | [VT-UTF8] | VT-UTF8 standard |
| **VT-UTF8 Combo Key Support** | Support of VT-UTF8 symbols for ANSI/ME100 terminals | |
| | [Disabled] | Support is disabled |
| | [Enabled] | Support is enabled |
| **Sredir Memory Display Delay** | Delay with device loading when displaying information about installed RAM to console PC | |
| | [No Delay] | No delay |
| | [Delay 1 Sec], | Set the delay of 1 sec. |
| | [Delay 2 Sec], | Set the delay of 2 sec. |
| | [Delay 4 Sec] | Set the delay of 4 sec. |

### 5.2.3 USB Configuration

Screen of the “USB Configuration” menu and description of items is given below.

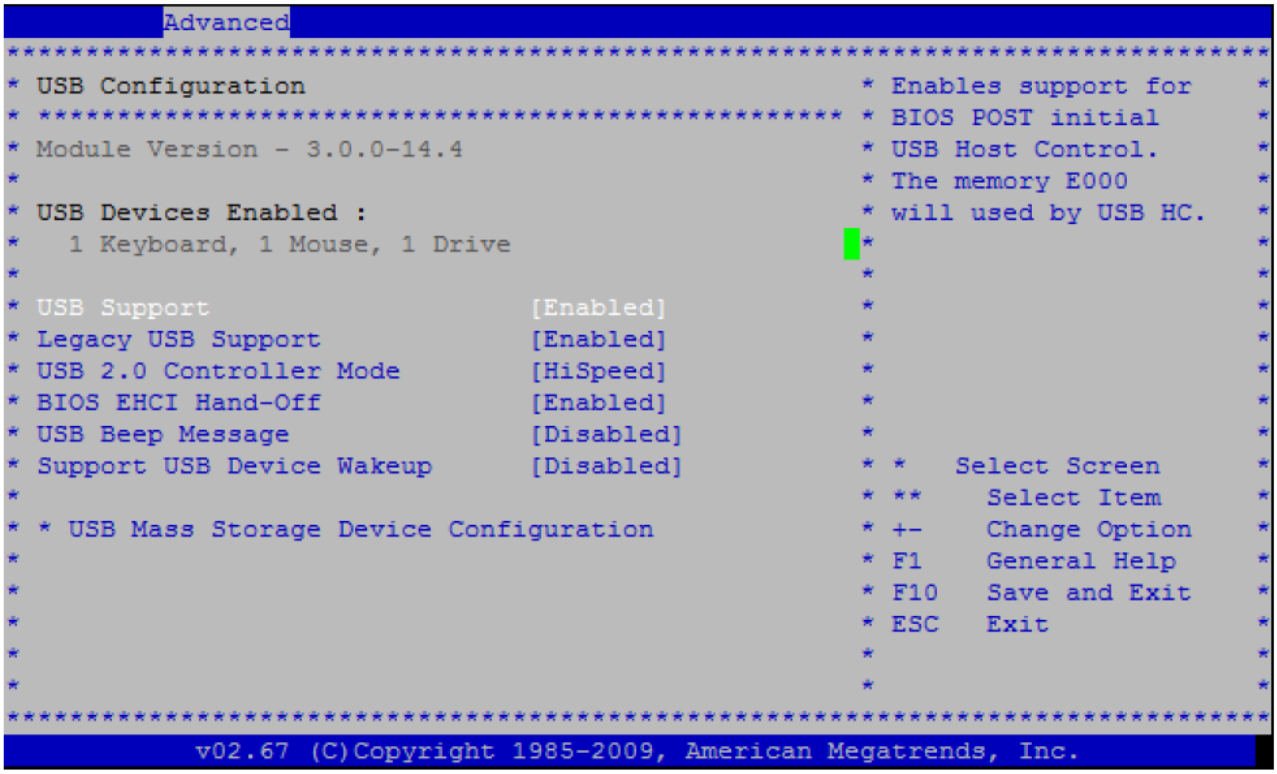


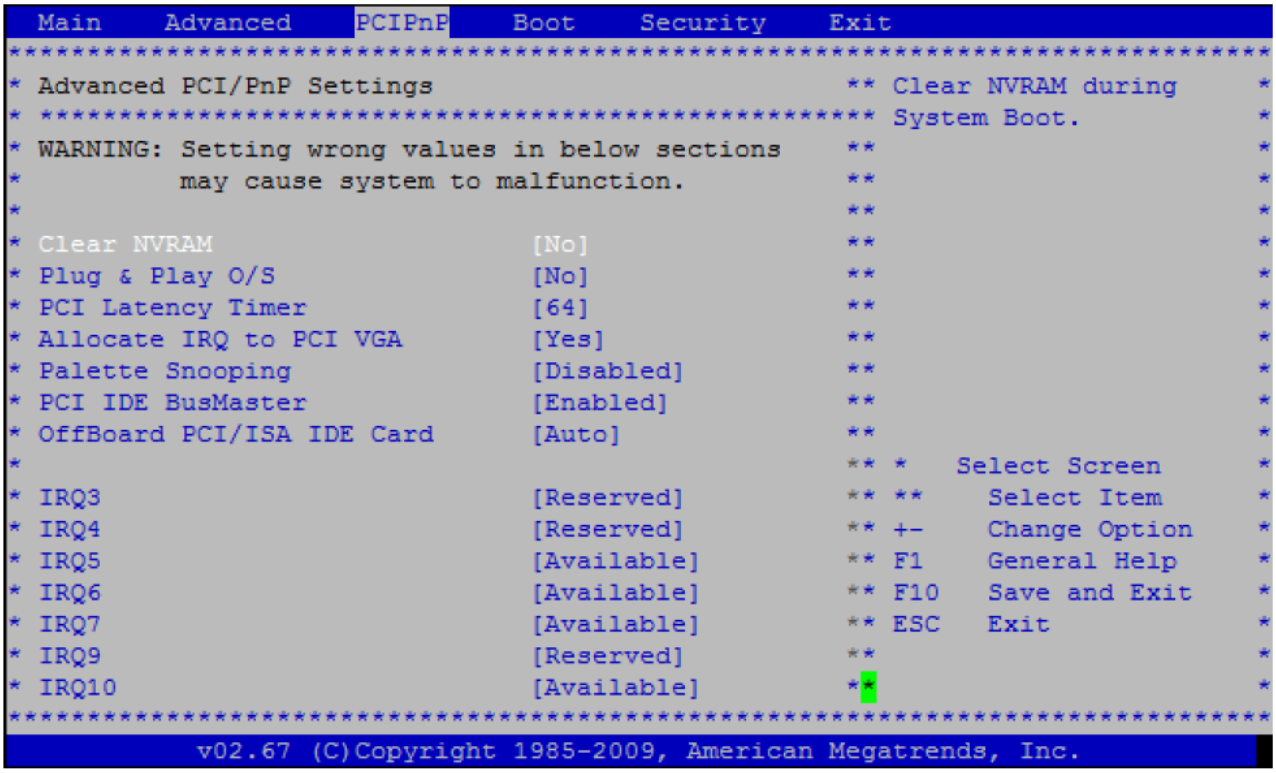

**Fig. 5-7 – Screen of the “USB Configuration” menu**

**Table 5-6 – Description of the “USB Configuration” menu**

| Menu item | Purpose | |
|---|---|---|
| **Legacy USB Support** | Support of the Legacy USB mode. | |
| | [Disabled] | The Legacy USB mode is disabled |
| | [Enabled] | The Legacy USB mode is enabled |
| | [Auto] | The Legacy USB mode is activated only if at least one USB device is connected |
| **USB 2.0 Controller Mode** | Determining the data exchange rate with USB devices | |
| | [HiSpeed] | Rate of data exchange 25-480 Mb/sec |
| | [FullSpeed] | Rate of data exchange 0.5-12 Mb/sec (USB 1.0/1.1 mode) |
| **USB EHCI Hand-Off** | Support by BIOS for the mechanism for transferring control of EHCI (Enhanced Host Controller Interface) interface between devices | |
| | [Disabled] | Controlled by operating system |
| | [Enabled] | Controlled by BIOS |

## 5.3 PCI / PnP

This tab contains items responsible for the operation of PCI and ISA buses, as well as interrupt switching management. The screen of PCI/PnP menu and menu descriptions are shown below.



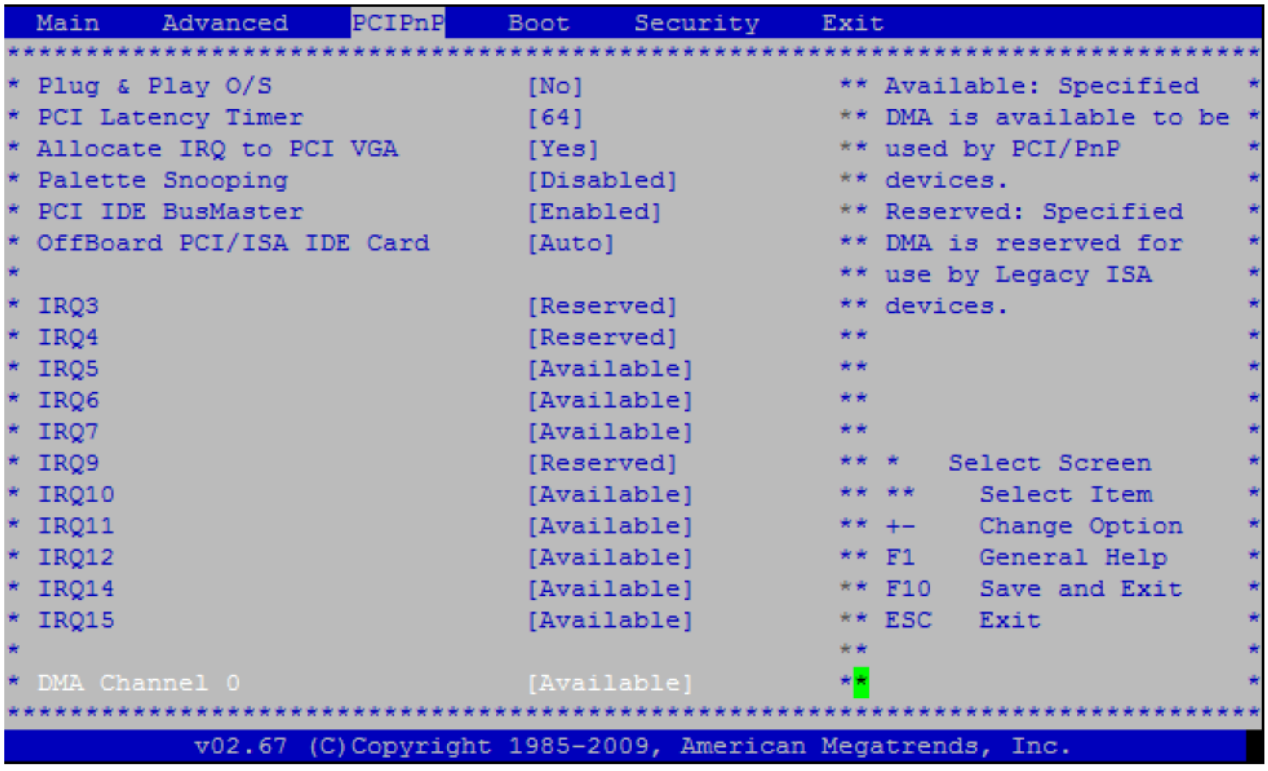




**Fig. 5-8 – Screen of the “PCI/ PnP” menu**

**Table 5-7 – Description of the "PCI/ PnP" menu**

| Menu item | Purpose | |
|---|---|---|
| Clear NVRAM | Reset of PnP parameters table | |
| | [No] | Without change |
| | [Yes] | Reset the table after reboot |
| Plug & Play O/S | OS with PnP support is installed | |
| | [No] | No |
| | [Yes] | Yes |
| PCI Latency Timer | The maximum number of PCI bus cycles during which a device connected to this bus can keep it busy while transferring the data. | |
| | [32], [64], [96], [128], [160], [192], [224], [248] | |
| Allocate IRQ to PCI VGA | Allowing the interrupt assignment to the graphics card on the PCI bus | |
| | [No] | Do not assign the PCI interrupt to the graphics card |
| | [Yes] | Assign the PCI interrupt to the graphics card |
| Palette Snooping | Synchronizing the colors of the graphics card and the image captured using the I/O graphics card (video editing card).<br>Function enabled | |
| | [Disabled] | The function is disabled. Recommended value |
| | [Enabled] | The function is enabled |
| PCI IDE BusMaster | Enabling the Bus Mustering PCI Mode for the IDE Bus Controller | |
| | [Disabled] | Disable the use of the Bus Mastering Mode |
| | [Enabled] | Enable the use of the Bus Mastering Mode |
| OffBoard PCI/ISA IDE Card | Selecting an external PCI/ISA card for the IDE bus controller | |
| | [Auto] | Automatic detection of the presence of PCI/ISA card of the IDE bus controller. Recommended value. |
| | [PCI Slot1], [PCI Slot2], [PCI Slot3], [PCI Slot4], [PCI Slot5], [PCI Slot6] | Indicate that an IDE bus controller card is installed in the relevant PCI slot |
| IRQ3<br>IRQ4<br>IRQ5<br>IRQ6<br>IRQ7<br>IRQ9<br>IRQ10<br>IRQ11<br>IRQ12<br>IRQ14<br>IRQ15 | Reservation of IRQ interrupt for internal Legacy devices Vortex86DX3 | |
| | [Available] | Enable the use of this interrupt by the PCI/PnP devices |
| | [Reserved] | Disable the use of this interrupt by the PCI/PnP devices, reserve for the Legacy devices.<br>Setting to the "Reserved" option will enable to use the IRQ line by external ISA devices (not PnP).<br>To use the IRQ line by external ISA devices (not PnP) you need to make sure that the line is unoccupied by other internal Vortex86DX3 devices. |
| DMA Channel 0 | Reservation of DMA channel for internal Legacy Vortex86DX3 devices | |

| | | |
|---|---|---|
| DMA Channel 1<br>DMA Channel 3<br>DMA Channel 5<br>DMA Channel 6<br>DMA Channel 7 | [Available] | Enable the use of this DMA channel by the PCI/PnP devices |
| | [Reserved] | Disable the use of this DMA channel by the PCI/PnP devices, reserve for the Legacy devices. |
| **Menu item** | **Purpose** | |
| Reserved Memory Size | Reserving memory for ISA bus devices by BIOS program | |
| | [Disabled] | Disable reserving memory for ISA bus devices by BIOS. Recommended value. |
| | [16k], [32k], [64k] | Reserve the specified memory capacity for ISA bus devices |

## 5.4 Boot

This tab contains menu items responsible for device booting modes, as well as for choosing the IDE device which will be used for booting the operating system. The screen of the “Boot” menu and description of menu items are specified below.


**Fig. 5-9 – Screen of the “Boot” menu**

**Table 5-8 – Description of the “Boot” menu**

| Menu item | Purpose |
|---|---|
| **Boot Settings Configuration** (submenu) | - |
| **1st Boot Device Priority** (submenu) | - |

### 5.4.1 Boot Settings Configuration

Screen of the “Boot Settings Configuration” menu and description of the menu items are given below.

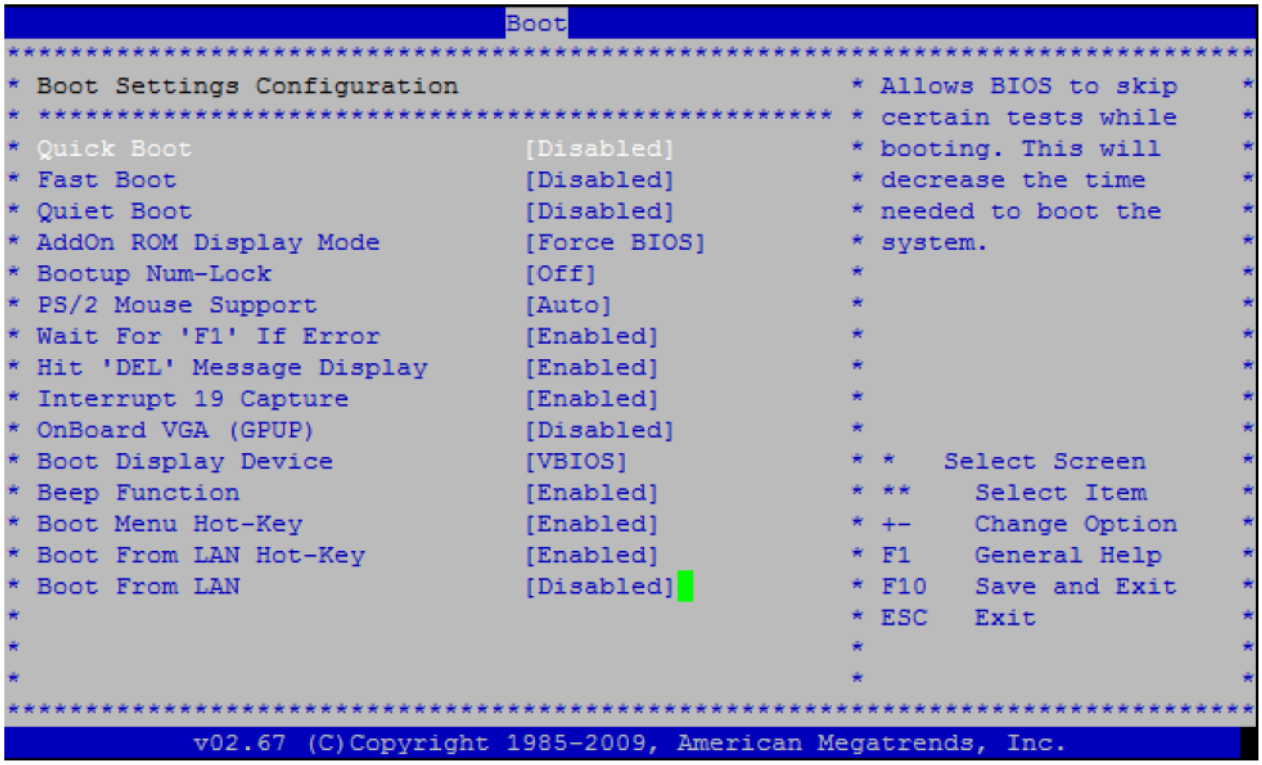



**Fig. 5-10 – Screen of the “Boot Settings Configuration” menu**

**Table 5-9 – Description of the “Boot Settings Configuration” menu**

| Menu item | Purpose | |
|---|---|---|
| **Quick Boot** | Quick boot | |
| | [Disabled] | Choosing this value provides complete self-test of the system when the power is switched on. |
| | [Enabled] | Choosing this value allows you to reduce the number of tests at the time of power-on and by doing so speed up the boot process. |
| **Add On ROM Display Mode** | The mode for displaying expansion cards | |
| | [Force BIOS] | This value allows displaying data from the BIOS of expansion cards to the monitor at the time of system boot. |
| | [Keep Current] | This value allows the computer system to display only P.O.S.T. information during the boot time |
| **Bootup Num-Lock** | Num Lock during the boot process | |
| | [Off] | Disabling the Num Lock during the boot process |
| | [On] | Enabling Num Lock during the boot process |
| **PS/2 Mouse Support** | Support of PS/2 mouse | |
| | [Disabled] | Support is disabled |

| | | |
|---|---|---|
| | [Enabled] | Support is enabled |
| **Menu item** | **Purpose** | |
| | [Auto] | Automatic detection of support. Recommended value |
| **Wait for 'F1' If Error** | Waiting for F1 to be pressed in the event of an error | |
| | [Disabled] | This option does not require waiting for user intervention in the event of an error. Select this value only if you know the reason why the BIOS error might appear. |
| | [Enabled] | Enable the BIOS system to wait for "F1" to be pressed in the event of an error at the time of booting |
| **Hit 'DEL' Message Display** | Displaying the message "Hit Del to enter Setup" during memory initialization | |
| | [Disabled] | Displaying the message is disabled |
| | [Enabled] | Displaying the message is enabled |
| **Interrupt 19 Capture** | Capturing the INT19 software interrupt | |
| | [Disabled] | BIOS will disable additional controllers to capture the INT19 interrupt |
| | [Enabled] | BIOS will enable additional controllers to capture the INT19 interrupt |

## 5.5 Security

Screen of the "Security" menu and description of menu items are specified below.

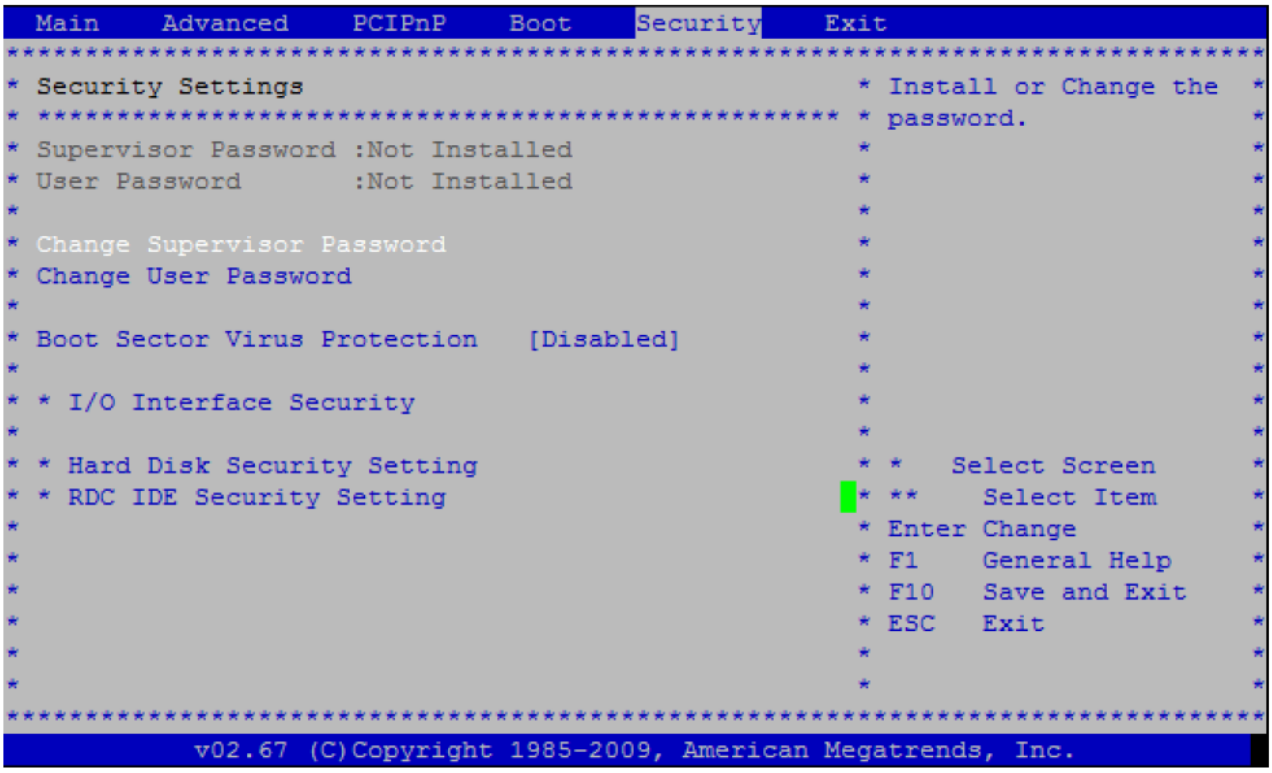



**Fig. 5-11 – Screen of the "Security" menu**

**Table 5-10 – Description of the "Security» menu**

| Menu item | Purpose | |
|---|---|---|
| Change Supervisor Password | Changing the password to allow system boot process (the request is displayed at the time of P.O.S.T.) | |
| Change User Password | Changing password for accessing to BIOS Setup (requested when entering BIOS Setup) | |
| Boot Sector Virus Protection | Protection of boot sector against viruses | |
| | [Disabled] | Selecting this value disables protection of the boot sector against viruses. |
| | [Enabled] | Selecting the "Enabled» value activates protection of the boot sector against viruses.<br>If any program (or virus) executes the Disk Format command or tries to write to the boot sector on the hard disk, a warning will be displayed on the monitor. While attempting to access the boot sector with protection enabled, the following messages appear:<br>**Boot Sector Write!**<br>**Possible VIRUS: Continue (Y/N)?_**<br>The following message appears after any attempt to format any hard drive via BIOS INT 13 Hard disk drive Service:<br>**Format!!!**<br>**Possible VIRUS: Continue (Y/N)?_** |

Submenu “I/O Interface Security” and description of menu items are given below.

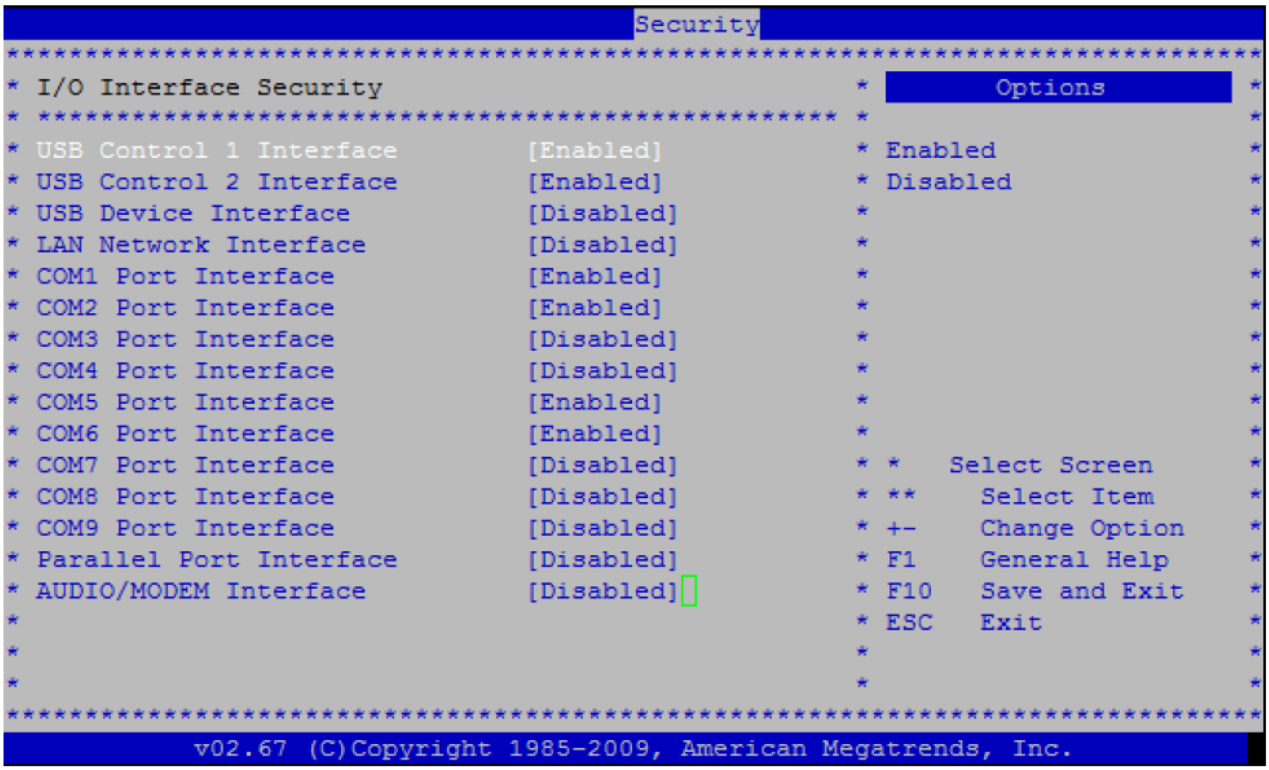



**Fig. 5-12 – Screen of the “SouthBridge Configuration” menu**

**Table 5-11 – Description of the “SouthBridge Configuration” menu**

| Menu item | Purpose | |
|---|---|---|
| **USB Control 1 Interface** | Controlling operation of the 0th and 1st USB ports | |
| | [Enabled] | Enable operation of the ports |
| | [Disabled] | Disable operation of the ports |
| **USB Control 2 Interface** | Controlling operation of the 2nd and 3rd USB ports | |
| | [Enabled] | Enable operation of the ports |
| | [Disabled] | Disable operation of the ports |
| **USB Device Interface** | [Enabled] | Enable operation of the ports |
| | [Disabled] | Disable operation of the ports |
| **LAN Network Interface** | Controlling operation of the Ethernet (LAN) integrated controller | |
| | [Enabled] | Enable operation of the controller |
| | [Disabled] | Disable operation of the controller |
| **COMx Port Interface** | [Enabled] | Enable operation of the COMx controller |
| | [Disabled] | Disable operation of the COMx controller |
| **Parallel Port Interface** | [Enabled] | Enable operation of the controller |
| | [Disabled] | Disable operation of the controller |
| **AUDIO/MODEM Interface** | [Enabled] | Enable operation of the controller |
| | [Disabled] | Disable operation of the controller |

## 5.6 Exit

Screen of the “Exit” menu and the relevant description are given below.

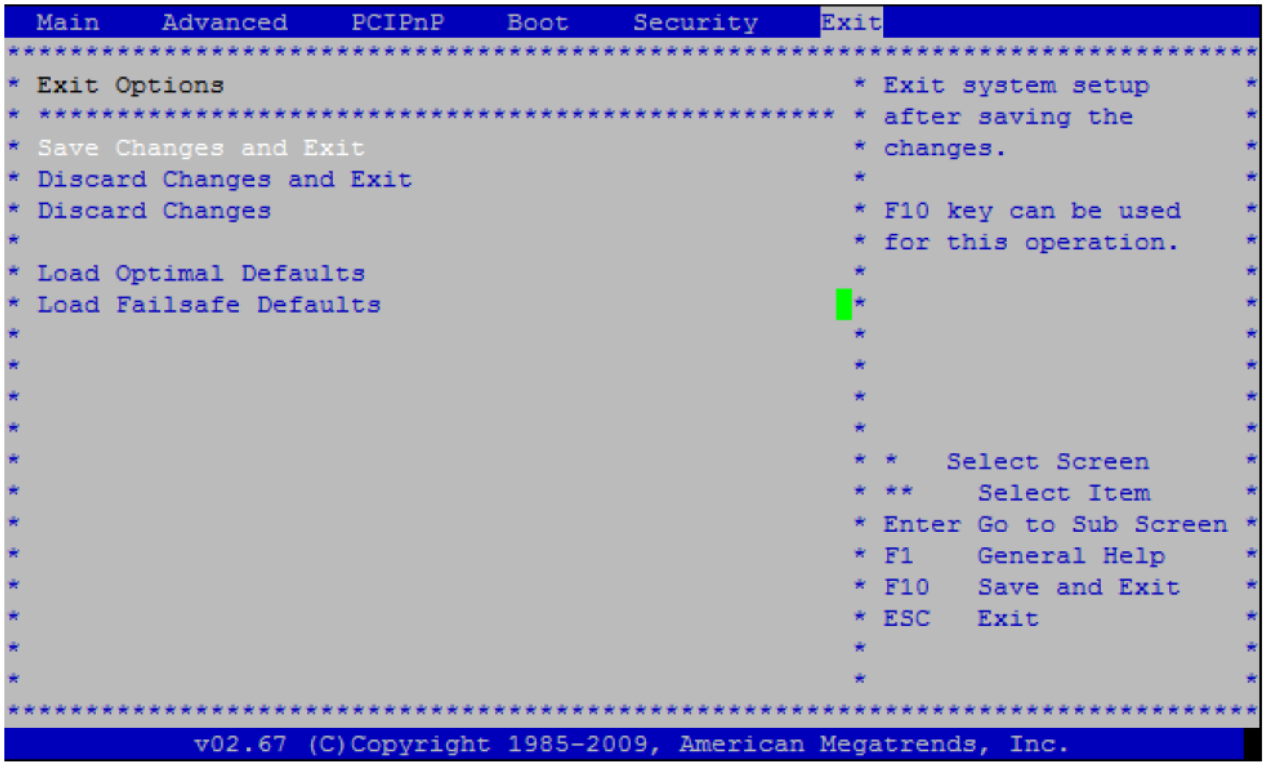



**Fig. 5-13 – Screen of the “Exit” menu**

**Table 5-12 – Description of the “Exit» menu (output)**

| Menu item | Purpose |
|---|---|
| Save Changes and Exit | Save settings in the CMOS and FRAM memory and exit BIOS Setup |
| Discard Changes and Exit | Exit without saving settings to CMOS and FRAM |
| Discard Changes | Discard the changes made in the settings without exiting BIOS Setup |
| Load Optimal Defaults | Load optimal (factory) settings without exiting BIOS Setup |

# ANNEX A

## Device programming FAQ

**1. Problem with the console via COM-port. You can enter the BIOS Setup, but when DOS starts, it is no longer possible to get there using keyboard via terminal. What is the reason?**

The most likely reason for this problem is BIOS Setup. By default, the remote console integrated into the AMI BIOS is enabled only until the BIOS transfers control to the operating system. In order to enable the console I/O integrated into the AMI BIOS, you need to change the BIOS Setup settings - in the “Advanced -> Remote Access Configuration” section, and set the “Redirection after BIOS POST” parameter to “Always” (when the device is delivered, this parameter is set by default in “Boot Loader”). However, it should be noted that the console implemented in the AMI BIOS uses a system timer. You can also use the FreeDOS tools (OS preinstalled by default), such as the **MODE** (change the parameters of I/O devices) and **CTTY** (change the standard I/O device) commands in the AUTOEXEC.BAT file:

```
MODE COMm[:] [BAUD[HARD]=b] [PARITY=p] [DATA=d] [STOP=s]
CTTY COMm
```

**COMm** – COM- port used (COM1, COM2, COM3, COM4). By default, in the BIOS Setup settings for the COM1 (RS-232) port and for the COM2 (RS-232) port, the base addresses are 3F8h and 2F8h, respectively. In order to use COM1 or COM2 (RS-232) for console I/O, the BAUD parameter must be set properly.

**BAUD** – baud rate code: 96 – 9600 bit/s, 192 – 19200 bit/s.

**BAUDHARD** – baud rate code: 96 – 9600 bit/s, 192 – 19200 bit/s, 384 – 38400 bit/s, 1152 – 115200 bit/s.

**PARITY** – parity (Even, Odd, Mark, Space, None).

**DATA** – number of data bits (7, 8)

**STOP** – number of stop bits (1, 2)

Examples of recording in the AUTOEXEC.BAT file:

```
MODE COM1 BAUDHARD=1152 PARITY=NONE DATA=8 STOP=1
CTTY COM1

MODE COM2 BAUD=96 PARITY=NONE DATA=8 STOP=1
CTTY COM2
```

However, it is necessary to consider certain restrictions during the work with the console when the FreeDOS operating system is running (it is preinstalled on the integrated flash drive at the time of delivery), precisely: pressing such keys as “Backspace” and "←", "→" (while no such problems were found when working with MSDOS).

# ANNEX B

## Terms and abbreviations

| Abbreviation | Definition |
|---|---|
| ACPI | Advanced Configuration and Power Interface |
| AGP | Accelerated Graphics Port |
| AGTL | Advanced Gunning Transceiver Logic |
| BIOS | Basic Input-Output System |
| DAC | Digital-Analog Converter |
| DDR SDRAM | Double Data Rate Synchronous Dynamic Random Access Memory |
| DMA | Direct Memory Access |
| DMI | Direct Media Interface |
| DVMT | Dynamic Video Memory Technology |
| ECC | Error Correction Code |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| EHCI | Enhanced Host Controller Interface (Universal Serial Bus specification) |
| EIDE | Enhanced Integrated Drive Electronics |
| EOS | Electrical Overstress |
| ESD | Electrostatically Sensitive Device<br>Electrostatic Discharge |
| FSB | Frequency System Bus |

| Abbreviation | Definition |
|---|---|
| FWH | Firmware Hub |
| GMCH | Graphics and Memory Controller Hub |
| I²C™ | Inter Integrated Circuit |
| LCD | Liquid crystal display |
| LPC | Low Pin Count |
| LVDS | Low Voltage Differential Signal |
| MDI | Media Dependent Interface |
| PC | Personal Computer |
| PIO | Programmed Input/Output |
| PLCC | Plastic Leaded Chip Carrier |
| PM | Peripheral Management Controller |
| POST | Power On Self Test |
| PSB | Processor System Bus |
| PWM output | Pulse-Width Modulation |

| | |
|---|---|
| RAMDAC | Random Access Memory Digital-to-Analog Converter |
| RTC | Real Time Clock |
| SMB | System Management Bus |
| **Abbreviation** | **Definition** |
| SMBus | System Management Bus |
| SODIMM | Small Outline Dual In-Line Memory Module |
| SSD | Solid State Disk |
| TFT | Thin Film Transistor |
| TTL | Transistor-Transistor Logic |
| UART | Universal Asynchronous Receiver-Transmitter |
| UHCI | Universal Host Controller Interface |
| USB | Universal Serial Bus |
| UTP | Unshielded Twisted Pair |

# ANNEX C: DISCLAIMER

This Disclaimer contains special operating conditions of Fastwel in the following areas: intellectual property, warranty policy, conditions of the order and delivery.

## 1 INTELLECTUAL PRORETY

1.1 If any infraction, interference, improper use, illegitimate exploitation and/or violation of the industrial and/or intellectual property rights of any third party and/or property, exploitation during the use of Fastwel Embedded Module will take place – Fastwel does not guarantee to replace the materials, computer programs, procedures or equipment affected by the complaint and under no circumstances doesn't bear responsibility in any form for possible refusal in case of such a replacement.
1.2 Use of the Fastwel products as well as the objects of intellectual property containing in them, in the ways and for the purposes, not provided by the present user manual and datasheet isn't allowed without preliminary written approval of Fastwel.
1.3 Fastwel is not responsible for possible incidents and losses, related to the operation of end devices, in which the original Fastwel equipment is used.

## 2 WARRANTY POLICY

2.1 When the detected flaws in an element can be corrected without decreasing the foreseen technical features and functionality for it, User may demand Fastwel the urgent correction of the failures in additionally agreed period and an increasing of the period of the guarantee of the element equal as the time elapsed from the formal request to repair the failures, until the receipt of the repaired element. All costs associated to the correction of failures, included those of assembly, dismantle, transport, tests, etc., if they exist, shall be prosecuted according the Warranty Policy of Fastwel.

## 3 ORDER AND DELIVERY CONDITIONS

3.1 The general rule is that all Fastwel equipment prices are determined with due consideration of delivery under the EXW terms and conditions (Incoterms 2010). Delivery of the products under other terms and conditions should be preliminary agreed and stated in writing between the parties.
3.2 Unless otherwise expressly agreed with Fastwel, all the deliveries of Fastwel equipment will be carried out only after the official purchase order is obtained and provided that the ordered products have been prepaid in full. Other terms and conditions of cooperation should be made in writing.
3.3 Any delivery of Fastwel electronics is submitted with the right package in accordance with the current rules and standards in the Member States of the European Economic Area. The purchaser independently bears all risks regarding the compliance of package and marking of Fastwel products with legislation requirements being in effect at the place of purchased products destination (in the buyer's country). The specified condition excludes unequivocally any liability of Fastwel for possible non-compliance of package and marking of products with the requirements of legislation of the country of products destination.
3.4 In general, all components of the supply are properly protected with respect to freight, in order to avoid any damage to the supply, third parties, environmental damages or unrelated goods, as consequence of wrong packaging.
3.5 Each package unit is labeled on the exterior area with the indications of product's Part Number and Serial Number.
3.6 The support documents for the order should be made either in English or in Russian unless otherwise agreed between parties in writing.
3.7 Fastwel does not pay penalties and does not cover costs associated with delay in the delivery of the products caused by actions of the third parties, force-majeure etc. - Fastwel doesn't bear any responsibility for non-execution or inadequate execution of the obligations in a case when it is caused by actions of the third parties (for example producers or suppliers of accessories), force majeure etc.
3.8 Fastwel declares that independently and at any time without damage, it has an exclusive right to define and change functionality architecture, bill of materials of its products without any preliminary coordination and approvals of the third parties.

## 4 OTHER CONDITIONS

4.1 Fastwel has the obligation to respect the current Russian legislation (including, but not limited to environmental, labor, social laws) in each moment and to apply it to its embedded electronics considering all and each execution phase, that is to say, from the design until the commissioning and subsequent maintenance. In this regard Fastwel is not liable to the user or other persons in connection with possible changes of the company's rules (including, but not limited to warranty, ordering policy) caused by changes of the Russian legislation.
4.2 Unless otherwise expressly agreed in writing, Fastwel provides no training for assembly\installation\adjustment\operation of its equipment.