GPC® AM4

General Purpose Controller AVR Mega, 4 type

TECHNICAL MANUAL

Intelligent module of the Abaco® BLOCK 4 series, 100x50 mm size; optional plastic mount for connection to DIN 46277-1 and DIN 46277-2 Ω rails. CPU ATmega103 at 5,5296 MHz, with: 128K internal FLASH EPROM, 4K internal RAM, 4K internal EEPROM. 8 A/D converter lines, 10 bits resolution, 0÷2,49 V, on standard pin out connector; two 8 bits timer counters, capable to generate PWM signals; one 16 bits timer counter, capable to generate two 10 bits PWM signals; watch dog completely managed by software; analog comparator; 15 interrupt sources (both internal and external). 32K of external RAM on socket. Real Time Clock able to display day, month, year, week day, seconds, minutes and hours. It can be programmed to issue an interrupt at intervals defined by software. Back up circuit for external RAM and RTC, with LITHIUM battery. 16 TTL I/O lines, on a standard pin out connector. Configuration jumper readable from software. 1 serial lines buffered in RS232 or RS422. RS485 or current loop. Expansion connector for ABACO® I/O BUS, on 26 pins standard connector. ISP interface. Facility of operation in Idle mode or Power Down mode. Single external power supply 5Vdc, 50 mA, protected by TransZorb™.

Wide range of developement software such as: Assembler; DDS MICRO-C AVR; ICC AVR; AVR BASIC; etc.
IMPORTANT

Although all the information contained herein have been carefully verified, grifo® assumes no responsibility for errors that might appear in this document, or for damage to things or persons resulting from technical errors, omission and improper use of this manual and of the related software and hardware.

grifo® reserves the right to change the contents and form of this document, as well as the features and specification of its products at any time, without prior notice, to obtain always the best product.

For specific informations on the components mounted on the card, please refer to the Data Book of the builder or second sources.

SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

⚠️ Attention: Generic danger

⚡️ Attention: High voltage

Trade marks

GPC®, grifo®: are trade marks of grifo®.

Other Product and Company names listed, are trade marks of their respective companies.
# GENERAL INDEX

**INTRODUCTION** ................................................................................................................... 1

**CARD VERSION** ................................................................................................................... 1

**GENERAL FEATURES** ............................................................................................................... 2
  - POWER SUPPLY ................................................................................................................ 3
  - SERIAL COMMUNICATION ................................................................................................. 3
  - CLOCK ...................................................................................................................................... 3
  - ABACO® I/O BUS ...................................................................................................................... 3
  - CPU ........................................................................................................................................... 4
  - MEMORY DEVICES .............................................................................................................. 4
  - DIGITAL I/O LINES ........................................................................................................... 6
  - REAL TIME CLOCK ............................................................................................................. 6
  - WATCH DOG ..................................................................................................................... 6
  - A/D CONVERTER ............................................................................................................... 6
  - TIMER COUNTER ............................................................................................................... 6
  - ISP INTERFACE ............................................................................................................... 7

**TECHNICAL FEATURES** .......................................................................................................... 8
  - GENERAL FEATURES ............................................................................................................ 8
  - PHYSICAL FEATURES ........................................................................................................... 8
  - ELECTRIC FEATURES .......................................................................................................... 9

**INSTALLATION** ................................................................................................................... 10
  - CONNECTIONS ................................................................................................................. 10
    - CN2 - EXTERNAL BACK UP BATTERY CONNECTOR .............................................. 10
    - CN1 - ABACO® I/O BUS CONNECTOR ......................................................................... 11
    - CN3 - SERIAL LINE CONNECTOR ............................................................................... 12
    - CN5 - DIGITAL I/O LINES CONNECTOR ...................................................................... 18
    - CN6 - A/D CONVERTER CONNECTOR .......................................................................... 20
    - CN9 - ISP INTERFACE CONNECTOR ............................................................................ 22
  - TRIMMERS AND CALIBRATION ...................................................................................... 23
  - I/O CONNECTION ................................................................................................................. 23
  - DIGITAL I/O INTERFACES ................................................................................................. 24
  - VISUAL SIGNALATIONS ..................................................................................................... 24
  - JUMPERS ................................................................................................................................ 25
    - 2 PINS JUMPERS ........................................................................................................... 25
    - 3 PINS JUMPERS ........................................................................................................... 26
  - SOLDER JUMPERS .............................................................................................................. 26
  - INTERRUPTS .................................................................................................................. 26
  - SERIAL COMMUNICATION SELECTION ...................................................................... 28
  - BACK UP ................................................................................................................................. 30
  - CONFIGURATION INPUT ................................................................................................... 30
  - MULTIPLEXED PINS ............................................................................................................ 31
  - RESET AND WATCH DOG ................................................................................................ 32
FIGURE INDEX

FIGURE 1: BLOCK DIAGRAM ............................................................................................................. 5
FIGURE 2: CARD PHOTO .................................................................................................................. 7
FIGURE 3: CN2 - EXTERNAL BACK UP BATTERY CONNECTOR ......................................................... 10
FIGURE 4: CN1 - ABACO® I/O BUS CONNECTOR ........................................................................ 11
FIGURE 5: CN3 - SERIAL LINE CONNECTOR .................................................................................... 12
FIGURE 6: SERIAL COMMUNICATION DIAGRAM ......................................................................... 13
FIGURE 7: RS 232 PIN OUT AND CONNECTION EXAMPLE .......................................................... 14
FIGURE 8: RS 422 PIN OUT AND POINT TO POINT CONNECTION EXAMPLE .................................. 14
FIGURE 9: RS 485 PIN OUT AND POINT TO POINT CONNECTION EXAMPLE .................................. 14
FIGURE 10: RS 485 PIN OUT AND NETWORK CONNECTION EXAMPLE ........................................ 15
FIGURE 11: 4 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE ....................... 16
FIGURE 12: 2 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE ....................... 16
FIGURE 13: CONNECTORS, MEMORIES, TRIMMER, BATTERY, ETC. LOCATION ............................ 17
FIGURE 14: CN5 - DIGITAL I/O LINES CONNECTOR ..................................................................... 18
FIGURE 15: DIGITAL I/O SIGNALS CONNECTION DIAGRAM ...................................................... 19
FIGURE 16: CN6 - A/D CONVERTER CONNECTOR ......................................................................... 20
FIGURE 17: A/D CONVERTER INPUTS DIAGRAM ........................................................................ 21
FIGURE 18: CN9 - ISP INTERFACE CONNECTOR .......................................................................... 22
FIGURE 19: VISUAL SIGNALATIONS TABLE .................................................................................... 24
FIGURE 20: JUMPERS SUMMARIZING TABLE ................................................................................ 25
FIGURE 21: 2 PINS JUMPERS TABLE ........................................................................................... 25
FIGURE 22: 3 PINS JUMPERS TABLE ........................................................................................... 26
FIGURE 23: JUMPERS LOCATION .................................................................................................. 27
FIGURE 24: SERIAL COMMUNICATION DRIVERS LOCATION ...................................................... 29
FIGURE 25: MULTIPLEXED PINS TABLE ...................................................................................... 31
FIGURE 26: COMPONENTS MAPS (SOLDERING SIDE AND COMPONENTS SIDE) ......................... 33
FIGURE 27: I/O ADDRESSES TABLE ............................................................................................. 36
FIGURE 28: MEMORY ALLOCATION ............................................................................................ 37
FIGURE 29: MEMORIES ACCESS .................................................................................................. 38
FIGURE 30: AVAILABLE CONNECTIONS DIAGRAM ................................................................. 43
FIGURE A1: MODULE DIMENSION FOR PIGGY BACK MOUNTING ............................................. A-1
FIGURE A2: PIGGY BACK MOUNTING .......................................................................................... A-2
FIGURE A3: WEIDMULLER RAIL MOUNTING .............................................................................. A-2
FIGURE C1: PPI EXPANSION ELECTRIC DIAGRAM .................................................................... C-1
FIGURE C2: SPA 03 ELECTRIC DIAGRAM ..................................................................................... C-2
FIGURE C3: QTP 16P ELECTRIC DIAGRAM .................................................................................. C-3
FIGURE C4: QTP 24P ELECTRIC DIAGRAM (1 OF 2) .................................................................. C-4
FIGURE C5: QTP 24P ELECTRIC DIAGRAM (2 OF 2) .................................................................. C-5
FIGURE C6: ABACO® I/O BUS INPUT OUTPUT ELECTRIC DIAGRAM ...................................... C-6
FIGURE C7: BUS INTERFACE ELECTRIC DIAGRAM ................................................................... C-7
FIGURE C8: IAC 01 ELECTRIC DIAGRAM ..................................................................................... C-8
INTRODUCTION

The use of these devices has turned - IN EXCLUSIVE WAY - to specialized personnel.

The purpose of this handbook is to give the necessary information to the cognizant and sure use of the products. They are the result of a continual and systematic elaboration of data and technical tests saved and validated from the manufacturer, related to the inside modes of certainty and quality of the information.

The reported data are destined- IN EXCLUSIVE WAY- to specialized users, that can interact with the devices in safety conditions for the persons, for the machine and for the enviroment, impersonating an elementary diagnostic of breakdowns and of malfunction conditions by performing simple functional verify operations , in the height respect of the actual safety and health norms.

The informations for the installation, the assemblage, the dismantlement, the handling, the adjustment, the reparation and the contingent accessories, devices etc. installation are destined - and then executable - always and in exclusive way from specialized warned and educated personnel, or directly from the TECHNICAL AUTHORIZED ASSISTANCE, in the height respect of the manufacturer recommendations and the actual safety and health norms.

The devices can't be used outside a box. The user must always insert the cards in a container that respect the actual safety normative. The protection of this container is not threshold to the only atmospheric agents, but specially to mechanic, electric, magnetic, etc. ones.

To be on good terms with the products, is necessary guarantee legibility and conservation of the manual, also for future references. In case of deterioration or more easily for technical updates, consult the AUTHORIZED TECHNICAL ASSISTANCE directly.

To prevent problems during card utilization, it is a good practice to read carefully all the informations of this manual. After this reading, the user can use the general index and the alphabetical index, respectly at the begining and at the end of the manual, to find information in a faster and more easy way.

CARD VERSION

The present handbook is reported to the GPC® AM4 card release 220399 and later. The validity of the bring informations is subordinate to the number of the card release. The user must always verify the correct correspondence among the two denotations. On the card the release number is present in more points both board printed diagram (serigraph) and printed circuit (for example between the CPU an the crystal on the component side).
GENERAL FEATURES

The GPC® AM4 belongs to the CPUs 4 type 100x50 mm size. It is a powerful control low cost module capable of operating in stand alone mode or as an intelligent peripheral, and/or remoted, in a wider telecontrol or acquisition network.

The GPC® AM4 module can be secured in a plastic mount for connection to Omega rails DIN 46277-1 and DIN 46277-3, thereby dispensing with the need of a rack and allowing a less costly mounting direct to the electrical control panel. Thanks to this small size, the GPC® 324 put into the same plastic rails that contains the Peripheral I/O, forming in this way an unique BLOCK element.

The GPC® AM4 can also be mounted as a macro CPU module on a peripheral card of the end user, in Piggy Back (stack through) mode.

The card most interesting characteristic is the used µP: the ATMEL ATmega 103 in fact is a very fast RISC microprocessor that includes a rich list of resources suitable to solve the common industrial automation problems.

At present there are some developing software tools which allow the card to be used either in assembly or with evoluted languages. Noteworthy among these are the development tools as the C and BASIC compilers, that are executed on a standard Personal Computer and that generate executable code for the card. This code is comfortably down loaded into the µP internal FLASH EPROM through a proper ISP programmer, connected to a serial line of the developing P.C.

For getting a quick prototype, cards such as SPA 03 and SPA 04 on which it is possible to mount the GPC® AM4 in Piggy Back mode, are used. The on board presence of the ABACO® I/O BUS connector, allows to drive directly I/O cards as: ZBR xxx, ZBT xxx, CAN 14, ADC 812, DAC 212 and so on, and through ABB 03, ABB 05 it is possible to manage all the peripheral cards available on ABACO® BUS.

- Intelligent module of the Abaco® BLOCK 4 series, 100x50 mm size.
- Optional plastic mount for connection to DIN 46277-1 and DIN 46277-3 Ω rails.
- CPU ATmega103 at 5,5296 MHz, with:
  128K internal FLASH EPROM;
  4K internal SRAM;
  4K internal EEPROM;
  8 A/D converter lines, 10 bits resolution, 0÷2,49 V, on standard pin out connector;
  Two 8 bits timer counters, capable to generate PWM signals;
  One 16 bits timer counter, capable to generate two 10 bits PWM signals;
  Watch dog completely managed by software;
  Analog comparator;
  15 interrupt sources (both internal and external);
- 32K of external SRAM on socket.
- Real Time Clock able to display day, month, year, week day, seconds, minutes and hours. It can be programmed to issue an interrupt at intervals defined by software.
- Back up circuit for external SRAM and RTC, with LITHIUM battery.
- 16 TTL I/O lines, on a standard pin out connector.
- Configuration jumper readable from software.
- 1 serial lines buffered in RS232 or RS422, RS485 or current loop.
- Expansion connector for ABACO® I/O BUS, on 26 pins standard connector.
- ISP interface compatible with µISP-S3 and µISP-S4 programmers by EQUINOX.
- Facility of operation in Idle mode or Power Down mode.
- Single external power supply 5Vdc, 50 mA, protected by TransZorb™.
- Wide range of development software such as: Assembler; DDS MICRO-C AVR; ICC AVR; AVR BASIC; Embedded PASCAL; etc.
Here follows a description of the board's functional blocks, with an indication of the operations performed by each one. To easily locate these blocks and verify their connections please refer to figure 1.

**POWER SUPPLY**

The card must be powered only with **+5 Vdc through the pin 25 (GND) and pin 26 (+5Vdc) of the CN1 connector**. The power supply circuit generates all the necessary voltages for the card and it is designed for reducing the consumption (the microprocessor power down modes are available) and for increasing the electrical noise immunity. In fact, as low as 48 mA of consumption for the normal working mode, allow the user to supply the board by batteries, solar panels, small power supplies, etc. Please remember that on board there is a protection circuit against voltage peaks by TransZorb™.

**SERIAL COMMUNICATION**

Serial communication is completely software settable for physical protocol and for speed which ranges from 2400 to as high as 115,2K Baud, with the available clock frequency. These settings are performed programming the UART inside the microprocessor, so for further informations, please refer to the manufacturer documentation or to appendix B of this manual. By hardware it is possible to select, through some on board jumpers, the electric communication protocol. In detail the line can be buffered in four different electrical protocols: **RS 232**, **current loop**, **RS 485**, **RS 422**; in this last cases also directionality and line activation is programmable.

**CLOCK**

GPC® AM4 is provided with a circuitry that generates the CPU clock frequency; this frequency is used also to generate the frequencies needed to the other sections of the board (Timer, serial lines, SPI, etc.). The default clock value is **5,5296 MHz** and if different values are required, please contact grifo®. We would remark that the CPU clock frequency normally coincides with the crystal oscillation frequency and it can also be divided by software through the power management section.

**ABACO® I/O BUS**

One of the most important features of GPC® AM4 is its possibility to be interfaced to industrial ABACO® I/O BUS. Thanks to this standard connector, the card can be connected to some of the numerous grifo® boards, both intelligent and not. For example the user can directly use cards for analog signals acquisition, cards for analog signals generation (D/A), cards for digital I/O signals management, cards with timers and counters, cards for temperature controls, etc. Even custom boards, designed to satisfy specific needs of the end user, can be connected. Using ABB 03 or ABB 05 mother boards it is possible manage also the BUS ABACO® single EURO cards. So GPC® AM4 becomes the right component for each industrial automation system, in fact ABACO® I/O BUS makes the card easily expandable with the best price/performance ratio.
CPU

GPC® AM4 board is designed to employ the AVR Mega 103 microcontroller manufactured by ATMEL. This 8 bits CPU is code compatible with the AVR RISC family so it features: a very high speed of execution; a good set of instructions (about 120) that includes: arithmetic and logic instructions, branch instructions, data transfer instructions, bit set reset and test instructions; efficient vectored interrupts management; wide range of addressing modes. Remarkable are the following features:

- 32 x 8 general purpose working register
- 128K bytes of In System Programming FLASH EPROM
- 4K bytes of internal SRAM
- 4K bytes of In System Programming EEPROM
- On chip analog comparator
- Programmable watch dog timer
- Programmable serial UART
- Master/Slave SPI serial interface
- Two 8 bit timers/counters with separate prescaler and PWM
- Expanded 16 bit timer/counter system with separate prescaler, compare and capture mode, dual 8/9/10 bits PWM
- 8 channels, 10 bit A/D converter
- 32 programmable I/O lines
- Low power idle, power save and power down modes
- Software selectable clock frequency
- External and internal interrupt sources

For further informations about this component please refer to the manufacturer documentation, or see appendix B of this manual.

MEMORY DEVICES

On the card are available 164K bytes of memory divided with a maximum of 128K FLASH EPROM 32K SRAM and 4K EEPROM. The GPC® AM4 memory use must be defined considering the application to realize or the specific requirements of the user. Normally the card is equipped with all the described memory and it can't be expanded nor reduced.

With the on board back up circuit there is the possibility to keep data, also when power supply is failed; in this way the card is always able to maintain parameters, logged data, system status and configuration, etc. without using expensive external UPS. The back up circuit is supplied by a on board lithium battery or an external battery to be connected to a specific connector.

The addressing of memory devices is directly controlled by the microprocessor, that provides to allocate the devices inside its address space.

For further information about memory configuration, sockets description and jumpers connection, please refer to "ADDRESSES AND MAPS" chapter and to "MEMORY SELECTION" paragraph.
**Figure 1: Block diagram**

- **CPU**: ATMega103
  - UART, ISP
  - I/O port, timer counter, SPI, A/D, etc.
- **CN3**: Serial line
- **CN9**: ISP interface
- **CN2**: External lithium
- **CN5**: 18 lines
- **CN6**: 10 lines
- **CN1**: ABACO® I/O bus
- **MULTIPLEXER**: Current loop drivers
- **CN18**: ABACO® I/O bus
- **RTC**: Real Time Clock
- **IC9**: SRAM
- **CONTROL LOGIC**: Multiplexer
- **3V LITHIUM ON BOARD**: Battery powers the circuit.
DIGITAL I/O LINES

On the board are available 16 digital I/O signals at TTL level with direction settable for each lines, directly managed by the microprocessor. These signals are connected to a 20 pins connector that is compatible with the standard I/O ABACO® connector, so the GPC® AM4 card can be connected to any of the numerous grifo® boards compliant to the same pinout.

By software the functions of these lines can be defined thanks to 9 microprocessor registers. Please remember that the I/O lines can also be connected and used in conjunction with some internal peripheral device, as SPI, Timers Counters, Interrupts, etc.

REAL TIME CLOCK

GPC® AM4 board is provided with a complete Real Time Clock device capable to manage hours, minutes, seconds, day of month, month, year and day of week in stand alone mode. The component is supplied by the back up circuitry to warrant data integrity in every working condition and is completely software programmable acting on 16 registers addressable in the CPU memory address space, by a specific control logic. The RTC section can generate interrupts at software programmable rates, for diverting the CPU from its normal tasks or awakening it from one of its low consumption working modes.

WATCH DOG

GPC® AM4 board is provided with a watch dog circuitry that, if used, allows to exit from infinite loop or abnormal conditions not managed by the application program. This circuitry is made by an astable section with a programmable intervent time (12÷1600 msec), it is completely software managed (by accessing an internal microprocessor register) and gives the board an exterme degree of safety.

A/D CONVERTER

The analog section of GPC® AM4 is based on the internal A/D converter with 8 input channels with 10 bits max resolution. By software the user selects the channel to convert, starts and stops the conversion, defines the conversion rate, enable end of conversion interrupt and so on, through the management of 4 microprocessor internal registers. The analog voltage input can be in the range 0÷2.490V as specified in "TRIMMER AND CALIBRATION" paragraph. All the input analog lines are provided of a filter capacitor that increases the noisy immunity and data stability.

TIMER COUNTER

The GPC® AM4 provides 3 general purpose timer counters, two with 8 bits and one with 16 bits. For each one can be defined the prescaler and the operation modes through software programmation of 18 microprocessor registers. The timer counters can either be used as a timer with an internal clock time base, as a counter with an external signal wich triggers the counting or as a pulse width modulated (PWM) section. For further information please refer to "MULTIPLEXED PINS" paragraph and appendix B of this manual.
ISP INTERFACE

The card supports the In System Programming technique that features serially downloadable memory allowing both code and data areas to be updated in system, without physically removing any device from the GPC® AM4 board. This serial programming is based around the industry standard SPI protocol which is a 3 wire bus featuring two data lines and a clock line. The ISP interface is compatible with Micro ISP-S3 and Micro ISP-S4 programmers by EQUINOX, that has extremely fast programming speeds due to the use of dedicated hardware to generate the ISP waveforms.

FIGURE 2: CARD PHOTO
TECHNICAL FEATURES

GENERAL FEATURES

Devices:
- 16 digital TTL input output lines
- Two 8 bit timers, counters
- One 16 bits timer, counter
- 4 PWM signals
- 1 RS 232, RS 422, RS 485, current loop serial line
- 8 A/D converter lines
- 1 reset contact
- 1 astable watch dog
- 1 real time clock
- 1 configuration jumper
- 1 ABACO® I/O BUS interface
- 1 ISP interface

Memory:
- IC 6: 64K x 16 (128K x 8) FLASH EPROM
  - 4K x 8 static RAM
  - 4K x 8 EEPROM
- IC 9: 32K x 8 static RAM

CPU:
- ATMEL AVR Mega 103

Clock frequency:
- 5.5296 MHz

A/D resolution:
- 10 bits

A/D conversion time:
- 5÷648 µsec (poor accuracy)
- 87÷324 µsec (high accuracy)

Watch dog intervent time:
- 12÷1600 msec

PHYSICAL FEATURES

Size (W x H x D):
- 100 x 50 x 25 mm (without container)
- 110 x 60 x 60 mm (with DIN rails container)

Weight:
- 60 g (without container)
- 120 g (with DIN rails container)

Connectors:
- CN1: 26 pins, male, vertical, low profile connector
- CN2: 2 pins, male, vertical, low profile connector
- CN3: 6 pins, Plug, female
- CN5: 20 pins, male, vertical, low profile connector
- CN6: 20 pins, male, vertical, low profile connector
- CN9: 10 pins, male, vertical, low profile connector
Temperature range: from 0 to 50 Centigrad degrees
Relative humidity: 20% up to 90% (without condens)

**ELECTRIC FEATURES**

Power Supply: +5 Vdc
Consumption on 5 Vdc: 48 mA in default configuration
On board back up battery: 3,0 Vdc; 180 mAh
External back up battery: 3,6÷5 Vdc
Back up current: 12 µA
A/D analog inputs: 0÷2,490 V
 Comparator analog inputs: 0÷5,000 V
Analog inputs impedance: 100 MΩ
RS 422, RS 485 line termination:
  - Line termination resistance= 120 Ω
  - Positive pull-up resistance= 3,3 KΩ
  - Negative pull-up resistance= 3,3 KΩ
INSTALLATION

In this chapter there are the information for a right installation and correct use of the card. The user can find the location and functions of each connectors, trimmers, jumpers and some explanatory diagrams.

CONNECTIONS

The GPC® AM4 module has 6 connectors that can be linkeded to other devices or directly to the field, according to system requirements. In this paragraph there are connectors pin outs, a short signals description (including the signals direction) and connectors location (see figure 13), plus some figures that describe how the interface signals are connected on the card.

CN2 - EXTERNAL BACK UP BATTERY CONNECTOR

CN2 is a 2 pins, vertical, male connector with 2,54mm pitch. Through CN2 the user can connect an external battery for SRAM and RTC back up when the power supply is switched off (for further information please refer to "BACK UP" and "ELECTRIC FEATURES" paragraphs).

Signals description:

\[
\begin{align*}
+V_{\text{bat}} &= I & \text{Positive pin of external back up battery} \\
GND &= & \text{Negative pin of external back up battery}
\end{align*}
\]
CN1 - ABACO® I/O BUS CONNECTOR

CN1 is a 26 pins, male, vertical, low profile connector with 2.54 mm pitch. Through CN1 the card can be connected to external expansion modules developed by the user or to the numerous grifo® boards, both intelligent and not. All this connector signals are at TTL level and follows the ABACO® I/O BUS standard.

Signals description:

A0-A7 = O - Address BUS.
D0-D7 = I/O - Data BUS.
/INT BUS (/INT7) = I - Interrupt request connected to /INT7 microprocessor signal.
/IORQ = O - Input output request.
/RD = O - Read cycle status.
/WR = O - Write cycle status.
/RESET = O - Reset.
+5 Vdc = I/O - +5 Vdc power supply.
GND = - Ground signal.
N.C. = - Not connected.

Figure 4: CN1 - ABACO® I/O BUS Connector
CN3 - SERIAL LINE CONNECTOR

CN3 is a 6 pins, female PLUG connector for the on board serial line, that can be buffered as RS 232, RS 422, RS 485 or current loop. Placing of the signal has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the electric protocol follows the CCITT normative.

![CN3 Serial Line Connector Diagram](Image)

**Signals description:**

- **RX RS 232** = I - Receive Data for RS 232 serial line.
- **TX RS 232** = O - Transmit Data for RS 232 serial line.
- **RX- RS 422** = I - Receive Data Negative for RS 422 differential serial line.
- **RX+ RS 422** = I - Receive Data Positive for RS 422 differential serial line.
- **TX- RS 422** = O - Transmit Data Negative for RS 422 differential serial line.
- **TX+ RS 422** = O - Transmit Data Positive for RS 422 differential serial line.
- **RXTX- RS 485** = I/O - Receive Transmit Data Negative for RS 485 differential serial line.
- **RXTX+ RS 485** = I/O - Receive Transmit Data Positive for RS 485 differential serial line.
- **RX- C.L.** = I - Receive Data Negative for Current Loop bipolar serial line.
- **RX+ C.L.** = I - Receive Data Positive for Current Loop bipolar serial line.
- **TX- C.L.** = O - Transmit Data Negative for Current Loop bipolar serial line.
- **TXB+ C.L.** = O - Transmit Data Positive for Current Loop bipolar serial line.
- **+5 Vdc/GND** = I - +5 Vdc or ground signal.
- **GND** = - Ground signal.
FIGURE 6: SERIAL COMMUNICATION DIAGRAM
**FIGURE 7: RS 232 PIN OUT AND CONNECTION EXAMPLE**

**FIGURE 8: RS 422 PIN OUT AND POINT TO POINT CONNECTION EXAMPLE**

**FIGURE 9: RS 485 PIN OUT AND POINT TO POINT CONNECTION EXAMPLE**
FIGURE 10: RS 485 PIN OUT AND NETWORK CONNECTION EXAMPLE

An RS 485 network must be forced with a suitable pair of resistors connected on a single point and it must be terminated only at the beginning and at the end of the physical line, near the master unit and near the farther slave unit.

On GPC® AM4 there are both the force and termination circuits that can be connected or disconnected through two specific jumpers, as described in the following paragraphs.

Please remember to connect the termination resistor near the master unit, if only it is not already available inside the same unit. In fact many RS 422 and RS 485 interfaces have this feature.

For further informations please refer to "RS 422 and RS 485 Interface Circuits" data book, by TEXAS INSTRUMENTS.
**FIGURE 11:** 4 wires Current Loop point to point connection example

**FIGURE 12:** 2 wires Current Loop point to point connection example
**FIGURE 13: CONNECTORS, MEMORIES, TRIMMER, BATTERY, ETC. LOCATION**
CN5 - DIGITAL I/O LINES CONNECTOR

CN5 is a 20 pins, male, vertical, low profile connector with 2.54 mm pitch. CN5 connects 16 digital I/O lines (and pertinent internal peripheral devices), plus 2 of the 4 timer counter output lines, to the external field signals. All the signals follow the standard I/O ABACO® pin out and they are at TTL level except AC- and AC+ analog comparator signals that range from 0 to +5 V.

Signals description:

PB.n = I/O - Port B digital line n.
PD.n = I/O - Port D digital line n.
PE.n = I/O - Port E digital line n.
/INTn = I - Interrupt request n.
Tn = I - Timer counter n input signal.
IC1 = I - Input Capture timer 1.
OCn = O - Timer counter n output signal.
MOSI, MISO, /SS, SCK = I/O - Serial peripheral interface signals.
AC+, AC- = I - Comparator analog input positive and negative.
PWMn = O - Pulse Width Modulated signal n.
+5 Vdc = O - +5 Vdc power supply.
GND = - Ground signal.

Further informations on these internal devices can be found in "MULTIPLEXED PIN" paragraph and appendix B of the manual.
FIGURE 15: DIGITAL I/O SIGNALS CONNECTION DIAGRAM
CN6 - A/D CONVERTER CONNECTOR

CN6 is a 20 pins, male, vertical, low profile connector with 2.54 mm pitch. CN6 connects the 8 A/D converter lines plus 2 of the 4 timer counter output lines, to the external field signals. The A/D lines are voltage analog signals (0÷2.490 V) provided of filter capacitors, while the timer counter lines are at TTL level. Placing of the signal has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the pin outs follows the A/D ABACO® standard.

![Figure 16: CN6 - A/D Converter Connector](image)

Signals description:

- **ADCn** = I - A/D converter analog input n.  
- **AGND** =  - Analog ground signal.  
- **PB.n** = I/O - Port B digital line n.  
- **OCnm** = O - Timer counter n, section m, output signal.  
- **PWMnm** = O - Pulse Width Modulated signal n, section m.  
- **+5 Vdc** = O - +5 Vdc power supply.  
- **GND** =  - Ground signal.
FIGURE 17: A/D CONVERTER INPUTS DIAGRAM
CN9 - ISP INTERFACE CONNECTOR

CN9 is a 10 pins, male, vertical, low profile connector with 2.54 mm pitch. CN9 connects the in system programming interface to an external ISP programmers. All the signals are at TTL level and they follow the EQUINOX standard pin out. The Micro ISP-S3 and Micro ISP-S4 programmers can be directly connected to CN9 connector through the included 10 pins flat cable, with no requirements of particular adapters or cables.

**FIGURE 18: CN9 - ISP INTERFACE CONNECTOR**

Signals description:

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROG</td>
<td>I - ISP program selection signal.</td>
</tr>
<tr>
<td>MO</td>
<td>I - ISP master output signal.</td>
</tr>
<tr>
<td>MI</td>
<td>O - ISP master input signal.</td>
</tr>
<tr>
<td>SCK1</td>
<td>I - ISP serial clock 1 signal.</td>
</tr>
<tr>
<td>RESET</td>
<td>I - Target RESET control signal.</td>
</tr>
<tr>
<td>+5 Vdc</td>
<td>O - +5 Vdc power supply.</td>
</tr>
<tr>
<td>GND</td>
<td>- Ground signal.</td>
</tr>
<tr>
<td>N.C.</td>
<td>- Not connected.</td>
</tr>
</tbody>
</table>
TRIMMERS AND CALIBRATION

On GPC® AM4 is available a trimmer, named RV1, that calibrates the Vref voltage of the A/D converter section. The GPC® AM4 is subjected to a careful test that verifies and calibrates all the card sections. The calibration is executed in laboratory, with a controlled +20°C room temperature, following these steps:

- The A/D voltage reference (Vref) is calibrated through RV1 trimmer, by using a 5 digits precision multimeter, to a value of +2,4900 Vdc. The reference voltage is measured between the two pins of C18 capacitor.
- The correspondance between the analog input signal and the combination read from A/D is verified. This check is performed with a reference signal connected to A/D inputs and testing that the A/D combination and the theoric combination differ at maximum of the A/D section errors sum (± 1 point).
- The trimmer is blocked with paint.

The analog interfaces use high precision components that are selected during mounting phase to avoid complicate and long calibration procedures. After the calibration, the on board trimmer is blocked with paint to maintain calibration also in presence of mechanic stresses (vibrations, movements, delivery, etc.).

The reference voltage generation circuit defines the full scale value for all the 8 analog inputs, that is: 0÷2,490 V; this value can't be changed so when higher analog signals must be acquired, the user must reduce them externally. For example a simple and cheap resistors divider can be used.

To recognize trimmer location on GPC® AM4, please refer to figure 13, while to get further informations on A/D converter section and analog comparator section (that doesn't require any calibration procedures) please read appendix B.

I/O CONNECTION

To prevent possible connecting problems between GPC® AM4 and the external systems, the user has to read carefully the information of the previous paragraphs and he must follow these instructions:

- For RS 232, RS 422, RS 485 or current loop communication signals the user must follow the standard rules of these protocols.
- For all TTL signals the user must follow the rules of this electric standard. The connected digital signal must be always referred to card digital ground (GND). For TTL signals, the 0 Vdc level corresponds to logic state "0", while 5Vdc level corresponds to logic state "1".
- The analog inputs (A/D section) must be connected to voltage signals in the 0÷2,490 V range. Remember that the analog inputs available on CN6 are provided of filter capacitors that ensure an higher stability of the acquired signals, but reduce at the same time the bandwidth frequency.
- The analog inputs (comparator section) must be connected to voltage signals in the 0÷5,000 V range.
DIGITAL I/O INTERFACES

With CN5 (compatible with standard I/O ABACO® connector) the GPC® AM4 can be connected to some of the numerous grifo® boards modules that have the same pin out. The connection of all these external modules is really simple in fact only a 20 ways flat cable, crimped with two 20 pins connectors, is necessary. This flat cable connect the power supply too and the user can make it himself or he can order it with the order code FLT.20+20. About software the use of digital I/O interfaces is likewise easy in fact GPC® AM4 software tools include proper drivers, library, example, etc. Below there is a brief description of the supported interfaces:

- QTP 16P, QTP 24P, KDx x24, DEB 01, etc.: they are usefull local operator panels. These boards already have all the resources (alphanumeric displays, matrix keyboards, LEDs etc.) necessary to solve the common man machine communication problems at a short distance from GPC® AM4. For software the programmer can use the relative procedure contained in all the GPC® AM4 software tools. These procedures normally are software drivers added to the language and they use directly its console instructions (for example PRINT and INPUT for BASIC, PRINTF and SCANF for C etc.), so for the user is very simple to write on display and to get data from keyboard.

- MCI 64: it a large mass memory support that can directly manage the PCMCIA memory cards RAM, FLASH, ROM, etc.) in their available sizes. About software the developed drivers provide procedures to read and write data at a specified address, for the selected programming language.

- IAC 01, DEB 01: it is an interface for CENTRONICS parallel printer that can be connected with a standard printer cable. The printer is managed by software through the high level instructions of the selected programming language (LPRINT for BASIC, PRINTF for C, etc.).

- RBO xx, TBO xx, XBI xx, OBI xx: these are buffer interfaces for I/O TTL signals. With these modules the the TTL input signals are converted in NPN or PNP optoisolated inputs and the TTL output signals are converted in relays or transistor optoisolated outputs.

For further information about the digital I/O interfaces please read "EXTERNAL CARDS" chapter and the software tools documentation.

VISUAL SIGNALATIONS

GPC® AM4 board is provided with a LED in order to signal to the user some internal status conditions, as described in the following table:

<table>
<thead>
<tr>
<th>LED</th>
<th>COLOUR</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD1</td>
<td>Red</td>
<td>Indicates the in system programmation status: active when ISP is enabled and viceversa (CN9 PROG signal).</td>
</tr>
</tbody>
</table>

FIGURE 19: VISUAL SIGNALATIONS TABLE

The main purpose of this LED is to provide the user a visual indication of the board status, making easier the operations to verify the correct working of the system. To easily locate the LED on the board please see figure 13, while for a description about ISP programming refer to the proper paragraph.
JUMPERS

On GPC® AM4 there are 7 jumpers for card configuration, 6 of them are solder jumpers. Thanks to these jumpers, the user can define for example the serial line configuration, the peripheral devices functionality and so on. Below there is the jumpers list, location and function.

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>PIN N°</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>2</td>
<td>It selects the configuration input status (RUN or DEBUG).</td>
</tr>
<tr>
<td>JS1, JS2</td>
<td>2</td>
<td>They connect the termination and force circuit to RS 422 and RS 485 serial line.</td>
</tr>
<tr>
<td>JS3</td>
<td>3</td>
<td>It selects the connection for pin 1 of CN3.</td>
</tr>
<tr>
<td>JS14</td>
<td>2</td>
<td>It connects the on board battery BT1, to back up circuit.</td>
</tr>
<tr>
<td>JS15</td>
<td>3</td>
<td>It selects the serial line configuration between RS 422 and RS 485 standards.</td>
</tr>
<tr>
<td>JS23</td>
<td>3</td>
<td>It defines connection for PE.6, INT6 CPU signal.</td>
</tr>
</tbody>
</table>

**FIGURE 20: JUMPERS SUMMARIZING TABLE**

The following tables describe all the right connections of GPC® AM4 jumpers with their relative functions. To recognize these valid connections, please refer to the board printed diagram (serigraph) or to figure 26 of this manual, where the pins numeration is listed; for recognizing jumpers location, please refer to figure 23. The "*" used in the following tables, denotes the default connection, or on the other hand the connection set up at the end of testing phase, that is the configuration the user receives.

2 PINS JUMPERS

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>FUNCTION</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>not connected</td>
<td>It connects configuration input to +5Vdc, and it selects the RUN mode.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>It connects configuration input to GND, and it selects the DEBUG mode.</td>
<td></td>
</tr>
<tr>
<td>JS1, JS2</td>
<td>not connected</td>
<td>The termination and force circuit is not connected to RS 422 or RS 485 serial line.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>The termination and force circuit is connected to RS 422 or RS 485 serial line.</td>
<td></td>
</tr>
<tr>
<td>JS14</td>
<td>not connected</td>
<td>The on board battery BT1 is not connected to back up circuit.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>The on board battery BT1 is connected to back up circuit.</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 21: 2 PINS JUMPERS TABLE**
3 PINS JUMPERS

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>FUNCTION</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>JS3</td>
<td>position 1-2</td>
<td>It connects pin 1 of CN3 to GND.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It connects pin 1 of CN3 to +5 Vcc.</td>
<td>*</td>
</tr>
<tr>
<td>JS15</td>
<td>position 1-2</td>
<td>It configures the serial line for RS 485 (half duplex on 2 wires).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It configures the serial line for RS 422 (full duplex or half duplex on 4 wires).</td>
<td></td>
</tr>
<tr>
<td>JS23</td>
<td>position 1-2</td>
<td>It connects pin 8 of the CPU (PE.6, /INT6) to real time clock interrupt signal.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It connects pin 8 of the CPU (PE.6, /INT6) to RS 422 and RS 485 DIR signal.</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 22: 3 pins jumpers table**

SOLDER JUMPERS

The solder jumpers called JSxx are connected by a thin copper connection on the solder side. So, if one of their configurations has to be changed, the user must first cut this connection using a sharpened cutter, then make the new connection using a low power soldering tool and some non-corrosive tin.

INTERRUPTS

One of the most important GPC® AM4 features is the powerful interrupts management. Here is a short description of how the board’s hardware interrupt signals can be managed; a more complete description of the hardware interrupts can be found in the microprocessor data sheets or in appendix B of this manual.

- **ABACO® I/O BUS**  ->  It generates an /INT7 interrupt, by the /INT BUS signal of CN1 connector.
- **Real Time Clock**  ->  It generates an /INT6 interrupt, according with JS23 connection.
- **I/O Ports**  ->  It generates the /INT0÷/INT5 interrupts, by the homonymous signal of CN5.
- **CPU inside devices**  ->  Possible sources of internal interrupt events are: Timers Counters, SPI, UART, A/D converter, Analog comparator, EEPROM.

The board has an interrupt unit that enables, disables, masks and defines priority of all the interrupt sources. The addresses of the interrupt service routines can be software programmed by the user acting on a suitable vector table saved in the microprocessor memory. So the user program has always the possibility to react promptly to every external event, also when more sources are simultaneously active.
**Figure 23: Jumpers Location**
SERIAL COMMUNICATION SELECTION

The communication serial line can be buffered in RS 232, RS 422, RS 485 or current loop electric standard. By hardware can be selected which one of these electric standard is used, through jumpers connection (as described in the previous tables). By software the serial lines can be programmed to operate with 8, 9 bits per character, no parity, 1 stop bit at standard or no standard baud rates, through some CPU internal registers setting. Some components necessary for RS 422, RS 485 and current loop communication are not mounted and not tested on the default configuration card, so the first not standard configuration must always be executed by grifo® technician; then the user can change himself the configuration, following the below description:

- SERIAL LINE CONFIGURED IN RS 232 (default configuration)
  
<table>
<thead>
<tr>
<th>JS1, JS2</th>
<th>JS15</th>
<th>JS23</th>
<th>IC1</th>
<th>IC2</th>
<th>IC3</th>
<th>IC4</th>
<th>IC5</th>
</tr>
</thead>
<tbody>
<tr>
<td>not connected</td>
<td>not connected</td>
<td>position 1-2</td>
<td>MAX 202 driver</td>
<td>no component</td>
<td>no component</td>
<td>no component</td>
<td>no component</td>
</tr>
</tbody>
</table>

- SERIAL LINE CONFIGURED IN CURRENT LOOP (.CLOOP option)

<table>
<thead>
<tr>
<th>JS1, JS2</th>
<th>JS15</th>
<th>JS23</th>
<th>IC1</th>
<th>IC2</th>
<th>IC3</th>
<th>IC4</th>
<th>IC5</th>
</tr>
</thead>
<tbody>
<tr>
<td>not connected</td>
<td>not connected</td>
<td>position 1-2</td>
<td>no component</td>
<td>no component</td>
<td>no component</td>
<td>HP 4200 driver</td>
<td>HP 4100 driver</td>
</tr>
</tbody>
</table>

  The current loop serial line is a passive line, so during connection the user must provide an external power supply, as described in figures 11 and 12. The current loop interface allows either point to point or network connection with 4 or 2 wires.

- SERIAL LINE CONFIGURED IN RS 422 (.RS422 option)

<table>
<thead>
<tr>
<th>JS1, JS2</th>
<th>JS15</th>
<th>JS23</th>
<th>IC1</th>
<th>IC2</th>
<th>IC3</th>
<th>IC4</th>
<th>IC5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(*)</td>
<td>position 2-3</td>
<td>position 2-3</td>
<td>no component</td>
<td>SN 75176 driver</td>
<td>SN 75176 driver</td>
<td>no component</td>
<td>no component</td>
</tr>
</tbody>
</table>

  With PE.6=DIR signal (managed by software with PORT E registers) the user enables or disables the transmitter driver:

<table>
<thead>
<tr>
<th>DIR</th>
<th>logic state</th>
<th>transmitter driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>low level</td>
<td>0</td>
<td>enabled</td>
</tr>
<tr>
<td>high level</td>
<td>1</td>
<td>disabled</td>
</tr>
</tbody>
</table>

  allowing either point to point (driver can be maintained always enabled) or network (driver is enabled only when the unit can hold the line) connection.
Figure 24: Serial communication drivers location
- SERIAL LINE CONFIGURED IN RS 485 (.RS485 option)

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1</td>
<td>no component</td>
</tr>
<tr>
<td>IC2</td>
<td>SN 75176 driver</td>
</tr>
<tr>
<td>IC3</td>
<td>no component</td>
</tr>
<tr>
<td>IC4</td>
<td>no component</td>
</tr>
<tr>
<td>IC5</td>
<td>no component</td>
</tr>
</tbody>
</table>

JS1, JS2 = (*)
JS15 = position 1-2
JS23 = position 2-3

With PE.6=DIR signal (managed by software with PORT E registers) the user defines the RS 485 line direction:

<table>
<thead>
<tr>
<th>Direction</th>
<th>Logic State</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>low level</td>
<td>0</td>
<td>RS 485 line transmitting</td>
</tr>
<tr>
<td>high level</td>
<td>1</td>
<td>RS 485 line receiving</td>
</tr>
</tbody>
</table>

allowing network connection in a master multi slave system and multi master system. With RS 485 communication line, on CN3 the pins 4 and 5 have the double function of reception and transmission signals. All the transmitted characters are at the same time received when the user select RS 485 on GPC® AM4; in this way the line conflicts can be immediately recognized by simply testing the received character after each transmission.

(*) With jumper JS1 and JS2 the RS 422 receiving line or the RS 485 line can be terminated and forced with a suitable resistors circuit. The line termination must be added only at the beginning and at the end of the physical line, connecting both the jumpers. Normally these jumper must be connected in point to point network, or on the farther cards in multipoints network.

After reset or power on phase, the PE.6=DIR signal is forced to high level that maintain the RS 485 driver receiving and that disables the RS 422 transmitter driver; this condition eliminates any conflict on the communication line.

For further information about serial communication, please refer to connection examples described in figures 6÷12 and to appendix B.

BACK UP

GPC® AM4 has an on board lithium battery BT1 for the back up of external SRAM and RTC content when power supply is switched off. Jumper JS14 connects physically the battery so it can be disconnected to save its charge whenever back up is not needed (delivery, stockage, etc.). By CN2 connector it is possible to connect an external battery; configuration of jumper JS14 does not affect the working of this battery and it can replace BT1 completely.

Please refer to the paragraph “ELECTRIC FEATURES” to choose the type of the external back up battery, while to easily find the back up components location see figure 13.

CONFIGURATION INPUT

GPC® AM4 board is provided with one jumper (J1), tipically used for system configuration purposes, that can be software acquired by the user program. The mostly implemented applications for this feature are: working conditions setting, selection of some on board firmware parameters, used language selection, etc. Some software tools use this jumper for the selection between the RUN and DEBUG working modalities. For further informations please refer to paragraphs "I/O ADDRESSES" and "CONFIGURATION JUMPER INPUT", while to easily locate its position on the board please refer to figure 23.
MULTIPLEXED PINS

The microcontroller used on the GPC® AM4 includes six 8 bits ports equal to 48 general purpose input output lines. Many of these lines are internally multiplexed and they assume different functions according with the performed software programmation. In the below table are described all the signals that have multiplexed functions on the card plus their status during the power on and reset phase:

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>MPX FUNCTIONS</th>
<th>USE ON GPC® AM4</th>
<th>RESET STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA.0:PA.7</td>
<td>AD0:AD7</td>
<td>AD0:AD7</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.0</td>
<td>/SS</td>
<td>PB.0, /SS on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.1</td>
<td>SCK</td>
<td>PB.1, SCK on CN5; SCK1 on CN9</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.2</td>
<td>MOSI</td>
<td>PB.2, MOSI on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.3</td>
<td>MISO</td>
<td>PB.3, MISO on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.4</td>
<td>OC0, PWM0</td>
<td>PB.4, OC0, PWM0 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.5</td>
<td>OC1A, PWM1A</td>
<td>PB.5, OC1A, PWM1A on CN6</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.6</td>
<td>OC1B, PWM1B</td>
<td>PB.6, OC1B, PWM1B on CN6</td>
<td>Tri state</td>
</tr>
<tr>
<td>PB.7</td>
<td>OC2, PWM2</td>
<td>PB.7, PWM2 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PC.0:7</td>
<td>A8:A15</td>
<td>A8:A15</td>
<td>Output low (0 V)</td>
</tr>
<tr>
<td>PD.0</td>
<td>/INT0</td>
<td>PD.0, /INT0 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PD.1</td>
<td>/INT1</td>
<td>PD.1, /INT1 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PD.2</td>
<td>/INT2</td>
<td>PD.2, /INT2 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PD.3</td>
<td>/INT3</td>
<td>PD.3, /INT3 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PD.4</td>
<td>IC1</td>
<td>PD.4, IC1 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PD.5</td>
<td>-</td>
<td>PD.5 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PD.6</td>
<td>T1</td>
<td>PD.6, T1 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PD.7</td>
<td>T2</td>
<td>PD.7, T2 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.0</td>
<td>PDI, RXD</td>
<td>RX on CN3; MO on CN9</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.1</td>
<td>PDO, TXD</td>
<td>TX on CN3; MI on CN9</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.2</td>
<td>AC+</td>
<td>PE.2, AC+ on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.3</td>
<td>AC-</td>
<td>PE.3, AC- on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.4</td>
<td>INT4</td>
<td>PE.4, /INT4 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.5</td>
<td>INT5</td>
<td>PE.5, /INT5 on CN5</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.6</td>
<td>INT6</td>
<td>/INT RTC; DIR</td>
<td>Tri state</td>
</tr>
<tr>
<td>PE.7</td>
<td>INT7</td>
<td>/INT BUS on CN1</td>
<td>Tri state</td>
</tr>
<tr>
<td>PF.0:PF.7</td>
<td>ADC0:ADC7</td>
<td>ADC0:ADC7 on CN6</td>
<td>High impedance input</td>
</tr>
</tbody>
</table>

**Figure 25: Multiplexed pins table**
The user can arbitrarily decide the use of the signals that on **GPC® AM4** has more than one functions, but he must take care of the other signals setting. The software tools developed for the card provide the right configuration of each multiplexed pins, so the user can simply acquire the old setting, change the allowed signals functions and then set the new configuration.

### RESET AND WATCH DOG

The watch dog circuit of **GPC® AM4** is really efficient and provided of easy software management. In details the most important features of this circuit are:

- astable functionality;
- software programmable intervent time from 12 to 1600 msec;
- software enable;
- software protected disable;
- software retrigger;

With the astable mode when the intervent time elapses, the circuit becomes active, it stay active till the end of reset time (about 130 msec) and after it is deactivated. The watch dog retrigger, enable and disable operation are performed by software as described on appendix B.

After an activation and following deactivation of /RESET signal, the card resumes execution of the program saved on the CPU internal FLASH EPROM (at address 00000H) starting from a global reset status of all the on board peripheral devices.

The /RESET signal is connected to homonymous pin of CN1 connector to allow reset condition of the external card too. Moreover on **GPC® AM4** are available other reset sources as the power good circuit, the ISP interface and the contact P1. The two pins of P1 can be connected to a normally open contact (i.e. a push button) and when the contact is closed (shortcut of the two pins) the reset circuit is enabled. The main purpose of this contact is to come out of infinite loop conditions, useful especially during debug and develop phases, or to ensure a particular initial status. Please see figure 13 for an easy localization of this contact.

Please remember that during a reset caused by watch dog circuit the /RESET signal of CN1 is not enabled, so the possible peripheral cards connected to **ABACO® I/O BUS** connector are not reset. Anyway the user can recognize the just happened watch dog intervention by testing the CPU MCUSR status register, at the beginning of the application program; if reset is caused by watch dog, then the user can set properly the peripheral cards always through the application program software.
FIGURE 26: COMPONENTS MAPS (SOLDERING SIDE AND COMPONENTS SIDE)
SOFTWARE

A wide selection of software development tools can be obtained, allowing use of the module as a system for its own development, both in assembler and in other high level languages; in this way the user can easily develop all the requested application programs in a very short time. Generally all software packages available for the mounted microprocessor, or for the AVR family, can be used. Below is described a brief list of the available software tools:

**AVR BASIC**: is a powerful new integrated development environment for AVR microcontroller. The toolset incorporates an editor, optimising BASIC compiler, assembler and HEX creator. The BASIC compiler produces very tight AVR machine code by virtue of the fact it translates the BASIC source into actually run time assembly code which is optimised to run as fast as possible. The target AVR microcontroller therefore runs true assembly code rather than tokenised code which is found in many other BASIC compilers. The upshot of this is that AVR BASIC code should be almost as fast as assembler written code.

**ICC AVR**: cross compiler for C source program. It is a powerful software tool that includes editor, ANSI C compiler, assembler, linker, library management program and project manager included in an easy to use integrated development environment for Windows and other P.C. operating systems. Library sources, floating point, integration with AVR studio, on line help and ANSI terminal emulator for target communication are provided too.

**DDS MICRO C AVR**: low cost cross compiler for C source program. It is a powerful software tool that includes editor, C compiler (integer), assembler, optimizer, source linker and library in one easy to use integrated development environment. There are also included the library sources and many utilities programs. The default IDE can be replaced by a new one named **Micro IDE**, that is more powerful, for Windows operating system and provided of many utility functions.

**Micro ISP-S3, S4**: it is a complete In System Programmer for ATMEL microcontroller (89S family and AVR family). It must be connected to a spare P.C. COM serial port and to target card ISP interface through a provided 10 pins flat cable plus connector. It can be used to download and upload code and data from target microprocessor FLASH EPROM and EEPROM in very fast and comfortable modality. It includes two status LEDs and it is completely driven by Meridian programmer interface software that has the following features: Windows 3.1, 95, 98 and NT compatible; wide list of device commands (device check signature, blank check, erase, read, write, verify, security, auto program, etc.); supports reading and writing of fuse bits and lock bits; binary and HEX files format; SPI parameters setting (RESET polarity, RESET timing, SPI frequency, write times, etc.); context sensitive help.

**AVR Studio**: it is a development tool for AVR family of microcontroller that fully control execution of program on AVR in circuit emulator or on the built in AVR instruction set simulator. AVR Studio supports source level execution of assembly and C programs generated by external compilers and assemblers. The tool is based on a set of windows for source, watch, registers, memory, peripherals, message and processor that enable the user to have full control of the status of every element in the execution target.
INTRODUCTION

In this chapter are reported all informations about card use, related to software programmation of GPC® AM4. For example the registers addresses, the memories and peripheral devices allocation are described below.

ON BOARD DEVICES ADDRESSES

The on board devices addresses are managed from a specific control logic, realized with CMOS gates. This control logic allocates memory and peripheral devices with very low power consumption and simple software management.

The ATMega 103 microprocessor addresses 64K words of code memory and 64K bytes of data memory and the control logic provides on board memory and peripheral devices allocation inside these addresses spaces.

Summarizing the control logic allocates:

- **ABACO® I/O BUS**
- 32K of SRAM on IC 9
- Configuration input J1
- Real Time Clock
- Internal microprocessor memories
- Internal microprocessor peripheral devices

The maps management is hardware defined so the addresses of all these devices can't be changed with different values. If some different specific maps are required, please contact directly grifo®. Further informations of maps and addresses can be found in the following paragraphs and in appendix B.

ABACO® I/O BUS ADDRESSES

The GPC® AM4 control logic manages the ABACO® I/O BUS and it defines the allocation addresses of this BUS. As described in the following "I/O ADDRESSES" table, the addresses from 8000H to 80FFH are available for ABACO® I/O BUS. Any I/O operations at each one of these addresses enables the /IORQ signal and the other control signals of CN1 connector. Please remember that ABACO® I/O BUS has only 8 address bits and 8 data bits, so when the peripheral card is mapped by hardware, only the least significant byte of the 16 bits I/O address is meaningful.
### I/O ADDRESSES

The on board control logic allocates all the external peripheral devices registers in the upper 32K bytes of the microprocessor data memory space. Next table shows names, addresses, meanings and directions of peripheral device registers (excluding the internal microprocessor ones). For a detailed description of the registers function, please refer to next chapter "PERIPHERAL DEVICES SOFTWARE DESCRIPTION".

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>REG.</th>
<th>ADDRESS</th>
<th>R/W</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABACO® I/O BUS</td>
<td>I/OBUS</td>
<td>8000H-80FFH</td>
<td>R/W</td>
<td>ABACO® I/O BUS addresses</td>
</tr>
<tr>
<td>J1</td>
<td>RUNDEB</td>
<td>A000H</td>
<td>R</td>
<td>Register for configuration input acquisition</td>
</tr>
<tr>
<td>Real Time Clock</td>
<td>SEC1</td>
<td>A000H</td>
<td>R/W</td>
<td>Data register for seconds units</td>
</tr>
<tr>
<td></td>
<td>SEC10</td>
<td>A001H</td>
<td>R/W</td>
<td>Data register for seconds decines</td>
</tr>
<tr>
<td></td>
<td>MIN1</td>
<td>A002H</td>
<td>R/W</td>
<td>Data register for minutes units</td>
</tr>
<tr>
<td></td>
<td>MIN10</td>
<td>A003H</td>
<td>R/W</td>
<td>Data register for minutes decines</td>
</tr>
<tr>
<td></td>
<td>HOU1</td>
<td>A004H</td>
<td>R/W</td>
<td>Data register for hours units</td>
</tr>
<tr>
<td></td>
<td>HOU10</td>
<td>A005H</td>
<td>R/W</td>
<td>Data register for hours decines and AM/PM</td>
</tr>
<tr>
<td></td>
<td>DAY1</td>
<td>A006H</td>
<td>R/W</td>
<td>Data register for day units</td>
</tr>
<tr>
<td></td>
<td>DAY10</td>
<td>A007H</td>
<td>R/W</td>
<td>Data register for day decines</td>
</tr>
<tr>
<td></td>
<td>MON1</td>
<td>A008H</td>
<td>R/W</td>
<td>Data register for month units</td>
</tr>
<tr>
<td></td>
<td>MON10</td>
<td>A009H</td>
<td>R/W</td>
<td>Data register for month decines</td>
</tr>
<tr>
<td></td>
<td>YEA1</td>
<td>A00AH</td>
<td>R/W</td>
<td>Data register for year units</td>
</tr>
<tr>
<td></td>
<td>YEA10</td>
<td>A00BH</td>
<td>R/W</td>
<td>Data register for year decines</td>
</tr>
<tr>
<td></td>
<td>WEE</td>
<td>A00CH</td>
<td>R/W</td>
<td>Data register for week day</td>
</tr>
<tr>
<td></td>
<td>REGD</td>
<td>A00DH</td>
<td>R/W</td>
<td>Control register D</td>
</tr>
<tr>
<td></td>
<td>REGE</td>
<td>A00EH</td>
<td>R/W</td>
<td>Control register E</td>
</tr>
<tr>
<td></td>
<td>REGF</td>
<td>A00FH</td>
<td>R/W</td>
<td>Control register F</td>
</tr>
<tr>
<td>Disable</td>
<td>DIS</td>
<td>E000H</td>
<td>R/W</td>
<td>Register for control signals disabling</td>
</tr>
</tbody>
</table>

**Figure 27: I/O addresses table**

The previous table shows only the registers of the peripheral devices external to microprocessor; for the internal registers please refer to specific documentation of the manufacturing company or to appendix B of this manual. About I/O maps the control logic avoids every conflicts problems between CPU internal and external peripherals.

The DIS register can be used to disable the external control signals after each access to the other registers in fact the **GPC® AM4** microcontroller mantains latched these control signals also when the operation is terminated.
MEMORY ADDRESSES

The memory addresses on **GPC® AM4** are described in the following figure:

**Figure 28: Memory Allocation**

- **Code Memory**
  - 1FFFFH
  - 10000H
  - 08000H
  - 1FFFFH

- **Data Memory**
  - 10000H
  - 0D000H
  - 0C000H
  - 0A000H
  - 08000H
  - 01000H
  - 00060H
  - 00020H
  - 00000H

- **IC6 Internal SRAM**
- **IC9 32K SRAM**
  - 7FFFH
  - 1000H
  - NOT USED

- **Real Time Clock**
- **RUNDEB (J1)**
- **ABACO® I/O Bus**
- **I/O Registers**
- **CPU Registers**
- **Not Used**
- **Disable**
The maximum 164K bytes of memory, are allocated on the board as below described:

- 64K words or 128K bytes of internal FLASH EPROM allocated in code memory space.
- 4K bytes of internal SRAM allocated in data memory space.
- 4K bytes of internal EEPROM.
- 32K bytes of external SRAM allocated in data memory space: only the last 28K bytes of this device areaddressed by the CPU, the remaining 4K bytes are not used.

The prefix internal means that the device is inside the microcontroller, while code memory and data memory are the areas addressed by the CPU. In details the GPC® AM4 memories can be managed as described in the following table:

<table>
<thead>
<tr>
<th>MEMORY</th>
<th>READ OPERATION</th>
<th>WRITE OPERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERNAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLASH EPROM</td>
<td>ISP interface</td>
<td>ISP interface</td>
</tr>
<tr>
<td></td>
<td>Direct CPU instructions</td>
<td></td>
</tr>
<tr>
<td>INTERNAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REGISTERS</td>
<td>Direct CPU instructions</td>
<td>Direct CPU instructions</td>
</tr>
<tr>
<td>INTERNAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRAM</td>
<td>Direct CPU instructions</td>
<td>Direct CPU instructions</td>
</tr>
<tr>
<td>INTERNAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EEPROM</td>
<td>ISP interface</td>
<td>ISP interface</td>
</tr>
<tr>
<td></td>
<td>Indirect CPU instructions</td>
<td>Indirect CPU instructions</td>
</tr>
<tr>
<td>EXTERNAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRAM</td>
<td>Direct CPU instructions</td>
<td>Direct CPU instructions</td>
</tr>
<tr>
<td>EXTERNAL</td>
<td>I/O REGISTERS</td>
<td>Direct CPU instructions</td>
</tr>
</tbody>
</table>

**FIGURE 29: MEMORIES ACCESS**

The external SRAM and I/O registers can be used through the direct CPU instructions only if the external data access has been enabled by the proper MCUCR internal register.

After power on or reset phase, the card starts execution of code saved at address 00000H of internal FLASH EPROM, with external data memory acess disabled.

For further informations on memories, addressing modes, timings, etc., please refer to appendix B.
In the previous paragraphs are described the external registers addresses, while in this one there is a specific description of registers meaning and function (please refer to I/O addresses table, for the registers names and addresses values). In the following paragraphs the D7÷D0 indication denotes the eight bits of the combination used in read/write operations.

**CPU INTERNAL PERIPHERALS**

The descriptions of the registers that manages the CPU internal peripheral devices (I/O Port, Timer Counter, A/D converter, EEPROM, etc) is available in the appendix B. Whenever this informations are still insufficient, please refer to specific documentation of the manufacturing company.

**CONFIGURATION INPUT (J1)**

The J1 configuration jumper installed on the GPC® AM4 board can be acquired simply by performing a read operation from RUNDEB registers and masking bit D7. The value is in complemented logic, this means that the connected jumper gives a logic value '0' while if the jumper is not connected the acquired logic value will be "1". The mostly implemented applications for this feature are: working conditions setting, selection of some on board firmware parameters, language selection, etc.

This jumper switches between the RUN (not connected) or the DEBUG (connected) mode, a feature used by some grifo® software tools, as described in the manuals of the tools themselves.

**REAL TIME CLOCK**

This peripheral is allocated in 16 consecutives addresses, 3 of which correspond to status registres while the remaining 13 are for datas. Data registers are used both for read operations (of the current time and date) and write operations (to initialize the time and date) just like the status registers which are used in write operations (to program the operative mode) and in read operations (to acquire the RTC status). Here follows a list of the RTC data registers' meanings:

- SEC1 - Units of seconds - 4 least significant bits of SEC1.3÷SEC1.0
- SEC10 - Decines of seconds - 3 least significant bits of SEC10.2÷SEC10.0
- MIN1 - Units of minutes - 4 least significant bits of MIN1.3÷MIN1.0
- MIN10 - Decines of minutes - 3 least significant bits of MIN10.2÷MIN10.0
- HOU1 - Units of hours - 4 least significant bits of HOU1.3÷HOU1.0
- DAY1 - Units of day number - 4 least significant bits of DAY1.3÷DAY1.0
- DAY10 - Decines of day number - 2 least significant bits of DAY10.1÷DAY10.0
- MON1 - Units of month - 4 least significant bits of MON1.3÷MON1.0
- MON10 - Decines of month - 1 least significant bit of MON10.0
- YEA1 - Units of year - 4 least significant bits of YEA1.3÷YEA1.0
- YEA10 - Decines of year - 4 least significant bits of YEA10.3÷YEA10.0
- WEE - Day of the week - 3 least significant bits of WEE.2÷WEE.0
For this last register the three least significant bits mean:

<table>
<thead>
<tr>
<th>WEE.2</th>
<th>WEE.1</th>
<th>WEE.0</th>
<th>Day of the week</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Sunday</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Monday</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Tuesday</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Wednesday</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Thursday</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Friday</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Saturday</td>
</tr>
</tbody>
</table>

While, the meaning of the three control registers is:

**REG D = NU NU NU NU 30S IF B H**

Where:
- **NU** = Not used.
- **30S** = If high (1) it allows a 30 seconds correction of the time. Once set the RTC seconds are reset and the minutes increased, if the previous seconds was equal or greater than 30.
- **IF** = It manages the RTC interrupt status. When read it shows the current interrupt status (1 active and viceversa), when reset to 0 it disables the RTC interrupt signal if the interrupt mode is selected.
- **B** = Indicates whether R/W operations can be performed on the registers:
  - 1 -> operations are not allowed and viceversa.
- **H** = If high (1) it stores the written time and date.

**REG E = NU NU NU NU T1 T0 I M**

Where:
- **NU** = Not used.
- **T1 T0** = Determines the duration of the internal counters interrupt cycle.
  - 0 0 -> 1/64 second
  - 0 1 -> 1 second
  - 1 0 -> 1 minute
  - 1 1 -> 1 hour
- **I** = It defines the interrupt operating mode:
  - 1 -> it selects interrupt mode: when the selected duration elapses the interrupt is enabled and then disabled only with a reset of bit IF of control register D;
  - 0 -> it selects the standard mode: when the selected duration elapses the interrupt is enabled and then disabled only after 7,8 msec.
- **M** = It masks the interrupt status:
  - 1 -> interrupt masked: the RTC interrupt signal is always disabled;
  - 0 -> interrupt not masked: the RTC interrupt signal reflects interrupt status.

**REG F = NU NU NU NU T 24/12 S R**

Where:
NU = Not used.

T = It determines from which internal counter to take the counting signal:
   1 -> main counter (fast counter for test);
   0 -> 15th counter.

24/12 = It determines the hours counting mode:
       1 -> 0÷23;
       0 -> 1-12 with AM/PM.

S = If high (1) it stops the clock time counting until the next enabling (0).

R = If high (1) it resets all the internal counters.

DISABLE CONTROL SIGNALS

The DIS register can be used to disable the external control signals after each access to the other external registers, in fact the GPC® AM4 microcontroller maintains latched the address signals, used to generates this peripheral control signal, also when the operation is terminated. The control signals that remain enabled are only the chip selects, while the other read and write control signals are correctly disabled. If the user wants to maintain the control signal normally disabled, he can simply perform a read or write operation to the DIS register address after each access to any others external registers.

For the on board peripheral devices (Real Time Clock and configuration input) the disable operation it is not necessary, while it is probably indispensable after ABACO® I/O BUS access, especially when the connected electronics is developed by the user and this uses the /IORQ signal.
EXTERNAL CARDS

GPC® AM4 can be connected to a wide range of block modules and operator interface system produced by grifo®, or to many system of other companies. The onboard resources can be expanded with a simple connection to the numerous peripheral grifo® boards, both intelligent and not, thanks to its standard ABACO® I/O BUS connector. Even single EURO cards with BUS ABACO® can be connected, by using the proper mother boards.

Hereunder some of these cards are briefly described; ask the detailed information directly to grifo®, if required.

**KDL xxx - KDF xxx**

Keyboard Display interface - LCD or Fluorescent

Interface with Fluorescent or LCD display, LEDs backlit, 20x2 or 20x4 characters; up to 24 keys matrix keyboard connector. It is directly driven by 16 TTL I/O lines; High level languages supported.

**QTP 24 - QTP 24P**

Quick Terminal Panel 24 keys - Parallel interface

Intelligent user panel equipped with Fluorescent or LCD display, LEDs backlit, 20x2 or 20x4 characters; RS 232, RS 422, RS 485 or current loop serial line; serial E² for set up and message. Possibility of re-naming keys, LEDs and panel name by inserting label with new name into the proper slot; 24 Keys and 16 LEDs with blinking attribute and buzzer manageable by software; built in power supply; RTC option, reader of magnetic badge and relay. The QTP 24P is low cost no intelligent (passive) version. It is directly driven from 16 TTL I/O lines; high level languages supported.

**QTP G28**

Quick Terminal Panel - LCD Graphic, 28 keys

Intelligent user panel equipped with graphic LCD display 240x128 pixel, CFL backlit; optocoupled RS 232 line and additional RS 232, RS 422, RS 485 or current loop serial line. Independent optional CAN line controller; serial E² for set up; RTC and RAM Lithium backed; primary graphic objects; possibility of renaming keys, LEDs and panel name by inserting label with new name into the proper slot; 28 Keys and 16 LEDs with blinking attribute and buzzer manageable by software; built in power supply; reader of magnetic badge, smart card and relay option.

**MCI 64**

Memory Cards Interfaces 64 MBytes

Interfacing card for managing 68 pins PCMCIA memory cards, it is directly driven from any ABACO® I/O standard connector; availability of high level drivers for programming languages.

**IBC 01**

Interface Block Communication

Conversion card for serial communication, 2 RS 232 lines; 1 RS 422–485 line; 1 optical fibre line; selectable DTE/DCE interface; quick connection for DIN 46277-1 and 3 rails.

**ADC 812**

Analog to Digital Converter, 12 bits, multi range

DAS (Data Acquisition System) multi range 8 channels 12 bit A/D conversion lines; track and hold; 6µs conversion time; range ±10, ±5, +10, +5Vdc or 0÷20, 4÷20mA; analog inputs connections through quick terminal screw connectors; ABACO® I/O BUS interface; direct mounting for DIN 247277-1 and 3 rails; 4 type dimension.
**Figure 30: Available Connections Diagram**

- **PC like or Macintosh**
- **PLC**
- **QTP G28**
- **Micro ISP**
- **PC**

**External Lithium Battery 3.6 V** for Back up

- **Serial Line RS 232, RS 422, RS 485, Current loop**

- **ABACO® I/O Bus**
  - **ZBx series**
  - **ABB 03 or ABB 05, etc.**

- **ABACO® BUS**
  - **ANY I/O TYPE**
  - **C/I R16-T16, etc.**
  - **IPC 52, UAR 24, etc.**

- **10 Bits ANALOG INPUT. VOLTAGE:**
  - **0±2.490 V**

- **3 COUNTER TIMER PWM, SPI, 6 INT, ANALOG COMPARATOR**

- **DIGITAL I/O INTERFACES:**
  - **QTP xxP**
  - **PRINTER MEMORY CARD**
  - **ENCODERS MOTORS**

- **DIGITAL TTL INPUT/OUTPUT**
  - **to XBI-01, OBI-01, RBO-08 etc.**
  - **RELAY TRANS. OPTO COUPLED**
**DAC 212**
Digital to Analog Converter 12 bits, multi range
Digital to Analog converter; multi range 2 channels 12 bits ± 10, +10 Vdc output; analog outputs connections through quick terminal screw connectors; **ABACO®** I/O BUS interface; direct mounting for DIN 247277-1 and 3 rails; 4 type dimension.

**CAN 14**
Control Area Network, 1 channel, galvanically insulated
UART CAN SJA1000; 1 serial channels galvanically insulated; **ABACO®** I/O BUS interface; 4 type dimension; support of CAN 2.0B protocol; transfer rate up to 1M bit/sec; direct mounting for DIN 247277-1 and 3 rails.

**ETI 324**
Encoder Timer I/O, 3 counters, 24 I/O
Three timers counters driven by 82C54; bidirectional optocoupled encoder input; direction identifier; phases multiplier; 24 digital lines driven by 82C55 on two standard I/O **ABACO®** connectors; **ABACO®** I/O BUS interface; direct mounting for DIN 247277-1 and 3 rails.

**OBI 01 - OBI 02**
Opto BLOCK Input NPN-PNP
Interface between 16 NPN, PNP optocoupled and displayed input lines, with screw terminal and **ABACO®** standard I/O 20 pins connector; power supply section; connection for DIN Ω rails.

**OBI N8 - OBI P8**
Opto BLOCK Input NPN-PNP
Interface between 8 NPN, PNP optocoupled and displayed input lines, with screw terminal and **ABACO®** standard I/O 20 pins connector; power supply section; connection for DIN Ω rails.

**TBO 01 - TBO 08**
Transistor BLOCK Output
Interface for **ABACO®** standard I/O 20 pins connector; 16 or 8 transistor output lines 45 Vdc 3 A open collector; screw terminal; optocoupled and displayed lines; connection for DIN 247277-1 and 3 rails.

**RBO 01**
Relé BLOCK Output
Interface for **ABACO®** standard I/O 20 pins connector; 8 displayed 5A or 10A relays; screw terminal; connection for DIN Ω rails.

**RBO 08 - RBO 16**
Relé BLOCK Output
Interface for **ABACO®** standard I/O 20 pins connector; 8 or 16 displayed Relays 3A with MOV; screw terminal; connection for DIN Ctype and Ω rails.

**XBI 01**
miXed BLOCK Input Output
Interface for **ABACO®** standard I/O 20 pins connector; 8 transistor output lines 45 Vdc 3A; 8 input lines; screw terminal; optocoupled and displayed I/O lines; connection for DIN 247277-1 and 3 rails.
**XBI R4 - XBI T4**

miXed BLOCK Input-Output

Interface for **ABACO**® standard I/O 20 pins connector; 4 Relays 3A with MOV or 4 optocoupled Transistors 3A open collectors; 4 input lines optocoupled; screw terminal; connection for DIN C type and Ω rails.

**FBC xxx**

Flat Block Contactxxx pins

This interconnection system "wire to board" allows the connection to many type of flat cable connectors to terminal for external connections. Connection for DIN Ω rails.

**ABB 05**

**ABACO**® Block BUS 5 slots

5 slots **ABACO**® mother board with power supply. Double power supply built in; 5Vdc 2,5A section for powering the on board logic; second section at 24Vdc 400mA galvanically coupled, for the optocoupled input lines. Auxiliary connector for **ABACO**® I/O BUS. Connection for DIN Ω rails.

**ABB 03**

**ABACO**® Block BUS 3 slots

3 slots **ABACO**® mother board; 4 TE pitch connectors; **ABACO**® I/O BUS connector; screw terminal for power supply; connection for DIN C type and Ω rails.

**ZBR xxx**

Zipped BLOCK Relays xx Input + xx Output

Peripheral cards family, relays outputs, equipped with housing for Ω rails mounting. Double power supply built in; 5Vdc section for powering the on board logic and an external CPU cards; second section, galvanically coupled, for the optocoupled input lines. All I/O lines are displayed by LEDs. Relays contacts are 3A and are MOV protected. All I/O connections are availables on quick terminal connectors. 1 connector interface to **ABACO**® I/O BUS. The ZBR serie represent the ideal complement for cards 3 and 4 type. The ZBR cards can be directly driven, through the PC parallel port, by using the PCC A26 low cost interface card. ZBR 324 -> 32 opto in, 24 relays; ZBR 246 -> 24 opto in, 16 relays; ZBR 168 -> 16 opto in, 8 relays; ZBR 84 -> 8 opto in, 4 relays.

**ZBT xxx**

Zipped BLOCK Transistors xx Input + xx Output

Peripheral cards family having optocoupled outputs and 3A transistor in open collector. Cards are equipped with housing for Ω rails mounting. Double power supply built in; 5Vdc section for powering the on board logic and an external CPU cards; second section, galvanically coupled, for the optocoupled input lines. All I/O lines are displayed by LEDs. All output transistors are equipped with protection against inductive loads. All I/O connections are availables on easy quick terminal connectors. Connector interface to **ABACO**® I/O BUS. The ZBT serie represent the ideal complement for cards 3 and 4 type. The ZBT cards can be directly driven, through the PC parallel port, by using the PCC A26 low cost interface card. ZBT 324 -> 32 opto in, 24 transistors; ZBT 246 -> 24 opto in, 16 transistors; ZBT 168 -> 16 opto in, 8 transistors; ZBT 84 -> 8 opto in, 4 transistors.
In this chapter there is a complete list of technical books, where the user can find all the necessary documentations on the components mounted on **GPC® AM4**.

Data book Manuale TEXAS INSTRUMENTS: *The TTL Data Book - SN54/74 Families*
Data book Manuale TEXAS INSTRUMENTS: *High Speed CMOS Logic Data Book*
Data book Manuale TEXAS INSTRUMENTS: *RS-422 and RS-485 Interface Circuits*
Data book Manuale TEXAS INSTRUMENTS: *Linear Circuits Voltage Regulators and Supervisor*

Data book NEC: *Memory Products*

Data book HEWLETT PACKARD: *Optoelectronics Designer's Catalog*

Data book MAXIM: *New Releases Data Book - Volume IV*

Data book NATIONAL SEMICONDUCTOR: *Linear Data Book - Volume 1*

Data sheets SEIKO EPSON: *REAL TIME CLOCK MODULE RTC-72421 Application manual*

Data book ATMEL: *AVR enhanced RISC microcontroller data book*

For further information and upgrades please refer also to specific internet web pages of the manufacturing companies.
APPENDIX A: CARD MECHANICAL MOUNTING

The GPC® AM4 can be physically mounted in two different manner. The first is the piggy back mounting (stack trough mode) that use the three connectors CN1, CN5 and CN6 for the interface with a user developed board. This connectors lead out of 7 mm on solder side and the user board must have proper female strip connectors (2.54 mm pitch) where the card can be plugged in, obtaining a single system.

The second mode expect a mounting inside a proper plastic container for a direct mounting on DIN 2477277-1 and 3 Ω rails; if the card is used with some other peripheral cards (i.e. ZBR xxx or ZBT xxx), a single longer container can be used obtaining a single module. The described plastic container code is 414487 type RS/100 by Weidmuller and it can be ordered to grifo® as BLOCK.Ell options, where lll is the required lenght in mm, or as BLOCK.4T options when a container is already supplied and only a 4 type card extension is required. By selecting this mounting the electric connection between GPC® AM4 and other peripheral cards is performed with a flat cable that must be really short or order it as FLT 26+26 I/O option.

In the following figures are described the module dimensions with the connector positions and some images that illustrate the connection modes.

**Figure A1: Module dimension for piggy back mounting**
**Figure A2: Piggy Back Mounting**

**Figure A3: Weidmuller Rail Mounting**
APPENDIX B: ON BOARD DEVICES DESCRIPTION

Features

• Utilizes the AVR RISC Architecture
• 8-bit High Performance and Low Power RISC Architecture
• Execution of Powerful Instructions in One Clock Cycle
• 32 x 8 General Purpose Working Registers + Peripheral Control Registers
• Up to 6 MIPS Throughput at 6 MHz

Data and Nonvolatile Program Memory

• 64K/128K Bytes of In-System Programmable Flash
  - Endurance: 1,000 Write/Erase Cycles
• 4K Bytes Internal SRAM
• 2K/4K Bytes of In-System Programmable EEPROM
  - Endurance: 100,000 Write/Erase Cycles
• Programming Lock for Flash Program and EEPROM Data Security
• SPI Interface for In-System Programming

Peripheral Features

• On-chip Analog Comparator
• Programmable Watchdog Timer with On-chip Oscillator
• Programmable Serial UART
• Master/Slave SPI Serial Interface
• Real Time Counter (RTC) with Separate Oscillator
• Two 8-bit Timer/Counters with Separate Prescaler and PWM
• Expanded 16-bit Timer/Counter system, with Separate Prescaler, Compare, Capture Modes and Dual 8-, 9-, or 10-bit PWM
• Programmable Watchdog Timer with On-chip Oscillator
• 8-channel, 10-bit ADC

Special Microcontroller Features

• Low-power Idle, Power Save and Power Down Modes
• Software Selectable Clock Frequency
• External and Internal Interrupt Sources

Specifications

• Low-power, High-speed CMOS Process Technology
• Fully Static Operation

Power Consumption at 4 MHz, 3V, 25°C

• Active: 5.5 mA
• Idle Mode: 1.6 mA
• Power Down Mode: < 1 µA

I/O and Packages

• 32 Programmable I/O Lines, 8 Output Lines, 8 Input Lines
• 64-pin TQFP

Operating Voltages

• 2.7 - 3.6V (ATmega603L and ATmega103L)
• 4.0 - 5.5V (ATmega603 and ATmega103)

Speed Grades

• 0 - 4 MHz (ATmega603L and ATmega103L)
• 0 - 6 MHz (ATmega603 and ATmega103)

The ATmega603/103 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega603/103 achieves throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega603/103 is supported with a full suite of program and system development tools including C compiler, macro assemblers, program solution tools, circuit simulation, and evaluation kits.
The ATmega603/103 AVR RISC Architecture

The ATmega603/103 AVR uses a Harvard architecture concept - with separate memories and buses for program and data. The program memory is accessed with a single level pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory in-system programmable Flash memory. With a few exceptions, AVR instructions have a single 16-bit word form meaning that every program memory address contains a single 16-bit instruction.

During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack effectively allocates in the general data SRAM, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 16-bit stack pointer SP is read/write accessible in the I/O space.

The 4000 bytes data SRAM can be easily accessed through the five different addressing modes supported in the AVR architecture.

**Architectural Overview**

**Figure 4.** The ATmega603/103 AVR RISC Architecture

**Figure 1.** The ATmega603/103 Block Diagram
A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The memory spaces in the AVR architecture are all linear and regular memory maps.

**General Purpose Register File**

Figure 5 shows the structure of the 32 general purpose working registers in the CPU.

**Figure 5. AVR CPU General Purpose Working Registers**

All the register operating instructions in the instruction set have direct and single cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and any register. ALU instructions like ADD, SUB, MUL, DIV and all other operations between two registers or on a single register apply to the entire register file.

As shown in Figure 5, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X, Y and Z registers can be set to index any register in the file.

### SRAM Data Memory

The ATmega603/103 supports two different configurations for the SRAM data memory, as listed in the following table:

**Table 2. Memory Configurations**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Internal SRAM Data Memory</th>
<th>External SRAM Data Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>40K</td>
<td>None</td>
</tr>
<tr>
<td>B</td>
<td>40K</td>
<td>up to 64K</td>
</tr>
</tbody>
</table>

Note: When using 64K of External SRAM, 60K will be available.
Figure 7. Memory Configurations

Memory Configuration A

Program Memory

Data Memory

$0000 - $001F

32 Registers

$0020 - $005F

64 I/O Registers

Internal SRAM

(4000 x 8)

$0060

$7FFF/$FFFF

$1000

$0FFF

Program Flash

(32K/64K x 16)

Memory Configuration B

Program Memory

Data Memory

$0000 - $001F

32 Registers

$0000 - $001F

64 I/O Registers

$0020 - $005F

Internal SRAM

(4000 x 8)

$0060

$7FFF/$FFFF

$1000

$0FFF

Program Flash

(32K/64K x 16)

The 4096 first Data Memory locations address both the Register file, the I/O Memory and the internal data SRAM. The first 96 locations address the register file and I/O memory, and the next 4000 locations address the internal data SRAM. An optional external data SRAM can be used with the ATmega603/103. This SRAM will occupy an area in the remain ing address locations in the 64K address space. This area starts at the address following the internal SRAM. If a 64K external SRAM is used, 4K of the external memory is lost as the addresses are occupied by internal memory.

When the addresses accessing the SRAM memory space exceed the internal data memory locations, the external data SRAM is accessed using the same instructions as for the internal data memory access. When the internal data memory is accessed, the read and write strobe pins (RD and WR) are inactive during the whole access cycle. External SRAM operation is enabled by setting the SRE bit in the MCLUCR register.

Accessing external SRAM takes one additional clock cycle per byte compared to access of the internal SRAM. This means that the commands LD, ST, LDS, STS, PUSH and POP take one additional clock cycle. If the stack is placed in external memory, interrupts, subroutine calls and returns take two clock cycles extra because the two-byte program counter is pushed and popped. When external SRAM interface is used with wait state, two additional clock cycles are used per byte.

This has the following effect: Data transfer instructions take two extra clock cycles, whereas interrupt, subroutine calls and returns will need four clock cycles more than specified in the instruction set manual.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-Decrement and Indirect with Pre-Increment. In the register file, registers R26 to R31 feature the indirect address pointers.

The Indirect with Displacement mode features a 63 address locations reach from the base address given by the Y register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers Y and Z are decremented and incremented.

The entire data address space including the 32 general purpose working registers and the 64 I/O registers are all accessible through these addressing modes. See the next section for a detailed description of the different addressing modes.

Program and Data Addressing Modes

The ATmega603/103 AVR RISC microcontroller supports powerful and efficient addressing modes for access to program memory (Flash) and data memory (SRAM, Register File and I/O Memory). This section describes the different addressing modes supported by the AVR architecture. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

Register Direct, Single Register Rd

Figure 8. Direct Single Register Addressing

The operand is contained in register d (Rd).
Register Direct, Two Registers Rd and Rr

Figure 9. Direct Register Addressing, Two Registers

Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O Direct

Figure 10. I/O Direct Addressing

Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

Data Direct

Figure 11. Direct Data Addressing

A 16-bit Data Address is contained in the 16 LSBs of a two-word instruction. Rd/Rr specify the destination or source register.

Data Indirect with Displacement

Figure 12. Data Indirect with Displacement

Operand address is the result of the Y or Z-register contents added to the address contained in 6 bits of the instruction word.
Data Indirect

Figure 13. Data Indirect Addressing

Operand address is the contents of the X, Y or the Z-register.

Data Indirect With Pre-Decrement

Figure 14. Data Indirect Addressing with Pre-Decrement

The X, Y or the Z-register is decremented before the operation. Operand address is the decremented contents of the X, Y or the Z-register.

Data Indirect With Post-Increment

Figure 15. Data Indirect Addressing with Post-Increment

The X, Y or the Z-register is incremented after the operation. Operand address is the content of the X, Y or the Z-register prior to incrementing.

Constant Addressing Using the LPM and ELPM Instructions

Figure 16. Code Memory Constant Addressing

Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 32K), LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1). If ELPM is used, LSB of the RAM Page Z register - RAMPZ is used to select low or high memory page (RAMPZ0 = 0: Low Page, RAMPZ0 = 1: High Page). ELPM does not apply to ATmega603.
Direct Program Address, JMP and CALL

Figure 17. Direct Program Memory Addressing

Program execution continues at the address immediate in the instruction words.

Indirect Program Addressing, IJMP and ICALL

Figure 18. Indirect Program Memory Addressing

Program execution continues at address contained by the Z-register (i.e. the PC is loaded with the contents of the Z-register).

Relative Program Addressing, RJMP and RCALL

Figure 19. Relative Program Memory Addressing

Program execution continues at address PC + k, where k is an offset of -2048 to 2047.

EEPROM Data Memory

The EEPROM memory is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 52, specifying the EEPROM address register, the EEPROM data register, and the EEPROM control register.

Memory Access Times and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access. The AVR CPU is driven by the System Clock Ø, directly generated from the external crystal for the chip. No internal clock division is used.

Figure 20 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 20. The Parallel Instruction Fetches and Instruction Executions

Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.
ATmega603/103

I/O Memory

The I/O space definition of the ATmega603/103 is shown in the following table:

<table>
<thead>
<tr>
<th>I/O Address (SRAM Address)</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3F ($5F)</td>
<td>SREG</td>
<td>Status Register</td>
</tr>
<tr>
<td>$3E ($5E)</td>
<td>SPH</td>
<td>Stack Pointer High</td>
</tr>
<tr>
<td>$3D ($5D)</td>
<td>SPL</td>
<td>Stack Pointer Low</td>
</tr>
<tr>
<td>$3C ($5C)</td>
<td>XDIV</td>
<td>XTAL Divide Control Register</td>
</tr>
<tr>
<td>$3B ($5B)</td>
<td>RAMPZ</td>
<td>RAM Page Z Select Register</td>
</tr>
<tr>
<td>$3A ($5A)</td>
<td>EICR</td>
<td>External Interrupt Control Register</td>
</tr>
<tr>
<td>$39 ($59)</td>
<td>EMSK</td>
<td>External Interrupt Mask Register</td>
</tr>
<tr>
<td>$38 ($58)</td>
<td>EIFR</td>
<td>External Interrupt Flag Register</td>
</tr>
<tr>
<td>$37 ($57)</td>
<td>TIMSK</td>
<td>Timer/Counter Interrupt Mask</td>
</tr>
<tr>
<td>$36 ($56)</td>
<td>TIFR</td>
<td>Timer/Counter Interrupt Flag</td>
</tr>
<tr>
<td>$35 ($55)</td>
<td>MUCR</td>
<td>MCU General Control Register</td>
</tr>
<tr>
<td>$34 ($54)</td>
<td>MCSR</td>
<td>MCU Status Register</td>
</tr>
<tr>
<td>$33 ($53)</td>
<td>TCCR0</td>
<td>Timer/Counter0 Control Register</td>
</tr>
<tr>
<td>$32 ($52)</td>
<td>TCNT0</td>
<td>Timer/Counter0 (8-bit)</td>
</tr>
<tr>
<td>$31 ($51)</td>
<td>OCR0</td>
<td>Timer/Counter0 Output Compare Register</td>
</tr>
<tr>
<td>$30 ($50)</td>
<td>ASSR</td>
<td>Asynchronous Mode Status Register</td>
</tr>
<tr>
<td>$3F ($4F)</td>
<td>TCCR1A</td>
<td>Timer/Counter1 Control Register A</td>
</tr>
<tr>
<td>$2E ($4E)</td>
<td>TCCR1B</td>
<td>Timer/Counter1 Control Register B</td>
</tr>
<tr>
<td>$2D ($4D)</td>
<td>TCNT1H</td>
<td>Timer/Counter1 High Byte</td>
</tr>
<tr>
<td>$2C ($4C)</td>
<td>TCNT1L</td>
<td>Timer/Counter1 Low Byte</td>
</tr>
<tr>
<td>$2B ($4B)</td>
<td>OCR1AH</td>
<td>Timer/Counter1 Output Compare Register A High Byte</td>
</tr>
<tr>
<td>$2A ($4A)</td>
<td>OCR1AL</td>
<td>Timer/Counter1 Output Compare Register A Low Byte</td>
</tr>
<tr>
<td>$29 ($49)</td>
<td>OCR1BH</td>
<td>Timer/Counter1 Output Compare Register B High Byte</td>
</tr>
<tr>
<td>$28 ($48)</td>
<td>OCR1BL</td>
<td>Timer/Counter1 Output Compare Register B Low Byte</td>
</tr>
<tr>
<td>$27 ($47)</td>
<td>ICR1H</td>
<td>Timer/Counter1 Input Capture Register High Byte</td>
</tr>
<tr>
<td>$26 ($46)</td>
<td>ICR1L</td>
<td>Timer/Counter1 Input Capture Register Low Byte</td>
</tr>
<tr>
<td>$25 ($45)</td>
<td>TCCR2</td>
<td>Timer/Counter2 Control Register</td>
</tr>
<tr>
<td>$24 ($44)</td>
<td>TCNT2</td>
<td>Timer/Counter2 (8-bit)</td>
</tr>
<tr>
<td>$23 ($43)</td>
<td>OCR2</td>
<td>Timer/Counter2 Output Compare Register</td>
</tr>
<tr>
<td>$22 ($42)</td>
<td>WDTCCR</td>
<td>Watchdog Timer Control Register</td>
</tr>
<tr>
<td>$1F ($3F)</td>
<td>EEsARH</td>
<td>EEPROM Address Register High</td>
</tr>
<tr>
<td>$1E ($3E)</td>
<td>EEsARL</td>
<td>EEPROM Address Register Low</td>
</tr>
<tr>
<td>$1D ($3D)</td>
<td>EE2DR</td>
<td>EEPROM Data Register</td>
</tr>
<tr>
<td>$1C ($3C)</td>
<td>EE2CR</td>
<td>EEPROM Control Register</td>
</tr>
</tbody>
</table>

Note: Reserved and unused locations are not shown in the table.

All the different ATmega603/103 I/Os and peripherals are placed in the I/O space. The I/O locations are direct accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I space. I/O registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. These registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set chapter for more details. When using the I/O specific instructions IN, OUT, the I/O register address $00 - $3F are used. When addressing I/O registers as RAM, $20 must be added to this address. All I/O register addresses throughout this document are shown with the SRAM address in parentheses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instruction work with registers $00 to $1F only.

Table 3. ATmega603/103 I/O Space (Continued)
The different I/O and peripherals control registers are explained in the following sections.

### Status Register - SREG
The AVR status register - SREG - at I/O space location $3F ($5F) is defined as:

<table>
<thead>
<tr>
<th>Bit</th>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3F ($5F)</td>
<td>T</td>
</tr>
</tbody>
</table>

#### Bit 7 - I: Global Interrupt Enable
The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is performed in separate control registers. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. The bit is cleared by hardware after an interrupt occurs, and is set by the RETI instruction to enable subsequent interrupts.

#### Bit 6 - T: Bit Copy Storage
The bit copy instructions BLD (Bit Load) and BST (Bit Store) use the T bit as source and destination for the operated bit.

#### Bit 5 - H: Half Carry Flag
The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

#### Bit 4 - S: Sign Bit, S = N & V
The S-bit is always an exclusive or between the negative flag N and the two’s complement overflow flag V. See the Instruction Set Description for detailed information.

#### Bit 3 - V: Two’s Complement Overflow Flag
The two’s complement overflow flag V supports two’s complement arithmetics. See the Instruction Set Description for detailed information.

#### Bit 2 - N: Negative Flag
The negative flag N indicates a negative result from an arithmetical or logical operation. See the Instruction Set Description for detailed information.

#### Bit 1 - Z: Zero Flag
The zero flag Z indicates a zero result from an arithmetical or logical operation. See the Instruction Set Description for detailed information.

#### Bit 0 - C: Carry Flag
The carry flag C indicates a carry in an arithmetical or logical operation. See the Instruction Set Description for detailed information.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

### Stack Pointer - SP
The general AVR 16-bit Stack Pointer is effectively built up of two 8-bit registers in the I/O space locations $3E ($5E) and $3D ($5D). As the ATmega603/103 supports up to 64 kB memory, all 16 bits are used.

<table>
<thead>
<tr>
<th>Bit</th>
<th>15 14 13 12 11 10 9 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3E ($5E)</td>
<td>SPH</td>
</tr>
</tbody>
</table>

#### Stack Pointer contents
- **$3E ($5E)**: SPH
- **$3D ($5D)**: SPF

Read/Write: R/W
Initial value: 0 0 0 0 0 0 0 0

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The stack pointer must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when an address is popped from the Stack with return from subroutine RET return from interrupt RETI.

### RAM Page Z Select Register - RAMPZ
The RAMPZ register is normally used to select which 64K RAM Page is accessed by the Z pointer. As the ATmega603/103 does not support more than 64K of SRAM memory, this register is used only to select which page in the program memory is accessed when the ELPM instruction is used. The different settings of the RAMPZ0 bit have the following effects:

- **RAMPZ0 = 0**: Program memory address $0000-$7FFF (lower 64K bytes) is accessed by ELPM
- **RAMPZ0 = 1**: Program memory address $8000-$FFFF (higher 64K bytes) is accessed by ELPM

Note that LPM is not affected by the RAMPZ setting.

### MCU Control Register - MCUCR
The MCU Control Register contains control bits for general MCU functions.

<table>
<thead>
<tr>
<th>Bit</th>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 ($5)</td>
<td>SRE</td>
</tr>
</tbody>
</table>

Read/Write: R/W
Initial value: 0 0 0 0 0 0 0 0

- **Bit 7 - SRE: External SRAM Enable**
  - When the SRE bit is set (one), the external data SRAM is enabled, and the pin functions A7-D7 (Port A), and A8-A15 (Port C) are activated as the alternate pin functions. Then the SRE bit overrides any pin direction settings in the respective direction registers. When the SRE bit is cleared (zero), the external data SRAM is disabled, and the normal pin direction settings are used.

- **Bit 6 - SRIW: External SRAM Wait State**
  - When the SRIW bit is set (one), a one cycle wait state is inserted in the external data SRAM access cycle. When the SF bit is cleared (zero), the external data SRAM access is executed with a three-cycle scheme. See Figure 51. External SRAM Access Cycle without wait states and Figure 52. External SRAM Access Cycle with wait state.

- **Bit 5 - SE: Sleep Enable**
  - The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit before the execution of the SLEEP instruction.
Bits 2..0 - Res: Reserved bits
These bits are reserved bits in the ATmega603/103 and always read zero.

XTAL Divide Control Register - XDIV
The XTAL Divide Control Register is used to divide the XTAL clock frequency by a number in the range 1 - 129. This feature can be used to decrease power consumption when the requirement for processing power is low.

- Bit 7 - XDIVEN: XTAL Divide Enable
  This bit sets (one) the clock frequency of the CPU and all peripherals is divided by the factor defined by the setting of XDIV6 - XDIV0. This bit can be set and cleared run-time to vary the clock frequency as suitable to the application.

- Bits 6..0 - XDIV6..XDIV0: XTAL Divide Select Bits 6 - 0
  These bits define the division factor that applies when the XDIVEN bit is set (one). The value of these bits is defined by the following formula for the resulting CPU clock frequency $f_{CLK}$:

$$f_{CLK} = \frac{f_{XTAL}}{129 - d}$$

The value of these bits can only be changed when XDIVEN is zero. When XDIVEN is set to one, the value written simultaneously into XDIV6..XDIV0 is taken as the division factor. When XDIVEN is cleared to zero, the value written simultaneously into XDIV6..XDIV0 is rejected. As the divider divides the master clock input to the MCU, the speed of peripherals is reduced when a division factor is used.

Table 4. Sleep Mode Select

<table>
<thead>
<tr>
<th>SM1</th>
<th>SM0</th>
<th>Sleep Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Idle Mode</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Power Down</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Power Save</td>
</tr>
</tbody>
</table>

Table 5. Reset and Interrupt Vectors

<table>
<thead>
<tr>
<th>Vector No.</th>
<th>Program Address</th>
<th>Source</th>
<th>Interrupt Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0000</td>
<td>RESET</td>
<td>Hardware Pin, Power-on Reset and Watchdog Reset</td>
</tr>
<tr>
<td>2</td>
<td>$0002</td>
<td>INT0</td>
<td>External Interrupt Request 0</td>
</tr>
<tr>
<td>3</td>
<td>$0004</td>
<td>INT1</td>
<td>External Interrupt Request 1</td>
</tr>
<tr>
<td>4</td>
<td>$0006</td>
<td>INT2</td>
<td>External Interrupt Request 2</td>
</tr>
<tr>
<td>5</td>
<td>$0008</td>
<td>INT3</td>
<td>External Interrupt Request 3</td>
</tr>
<tr>
<td>6</td>
<td>$000A</td>
<td>INT4</td>
<td>External Interrupt Request 4</td>
</tr>
<tr>
<td>7</td>
<td>$000C</td>
<td>INT5</td>
<td>External Interrupt Request 5</td>
</tr>
<tr>
<td>8</td>
<td>$000E</td>
<td>INT6</td>
<td>External Interrupt Request 6</td>
</tr>
<tr>
<td>9</td>
<td>$0010</td>
<td>INT7</td>
<td>External Interrupt Request 7</td>
</tr>
<tr>
<td>10</td>
<td>$0012</td>
<td>TIMER2 COMP</td>
<td>Timer/Counter2 Compare Match</td>
</tr>
<tr>
<td>11</td>
<td>$0014</td>
<td>TIMER2 OVF</td>
<td>Timer/Counter2 Overflow</td>
</tr>
<tr>
<td>12</td>
<td>$0016</td>
<td>TIMER1 CAPT</td>
<td>Timer/Counter1 Capture Event</td>
</tr>
<tr>
<td>13</td>
<td>$0018</td>
<td>TIMER1 COMPA</td>
<td>Timer/Counter1 Compare Match A</td>
</tr>
<tr>
<td>14</td>
<td>$001A</td>
<td>TIMER1 COMPB</td>
<td>Timer/Counter1 Compare Match B</td>
</tr>
<tr>
<td>15</td>
<td>$001C</td>
<td>TIMER1 OVF</td>
<td>Timer/Counter1 Overflow</td>
</tr>
<tr>
<td>16</td>
<td>$001E</td>
<td>TIMER0 COMP</td>
<td>Timer/Counter0 Compare Match</td>
</tr>
<tr>
<td>17</td>
<td>$0020</td>
<td>TIMER0 OVF</td>
<td>Timer/Counter0 Overflow</td>
</tr>
<tr>
<td>18</td>
<td>$0022</td>
<td>SPI, STC</td>
<td>SPI Serial Transfer Complete</td>
</tr>
<tr>
<td>19</td>
<td>$0024</td>
<td>UART, RX</td>
<td>UART, Rx Complete</td>
</tr>
<tr>
<td>20</td>
<td>$0026</td>
<td>UART, UDRE</td>
<td>UART Data Register Empty</td>
</tr>
<tr>
<td>21</td>
<td>$0028</td>
<td>UART, TX</td>
<td>UART, Tx Complete</td>
</tr>
<tr>
<td>22</td>
<td>$002A</td>
<td>ADC</td>
<td>ADC Conversion Complete</td>
</tr>
<tr>
<td>23</td>
<td>$002C</td>
<td>EE READY</td>
<td>EEPROM Ready</td>
</tr>
<tr>
<td>24</td>
<td>$002E</td>
<td>ANALOG COMP</td>
<td>Analog Comparator</td>
</tr>
</tbody>
</table>

Reset and Interrupt Handling

The ATmega603/103 provides 23 different interrupt sources. These interrupts and the separate reset vector each have separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be set (one) together with the I-bit in the status register in order to enable the interrupt.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. A complete list of vectors is shown in Table 5. The list also determines the priority levels of the different interrupts. The list shows the highest priority level as the lowest number. RESET has the highest priority, and next is INT0 - the External Interrupt Request 0.
The most typical program setup for the Reset and Interrupt Vector Addresses are:

<table>
<thead>
<tr>
<th>Address Labels</th>
<th>Code</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0000</td>
<td>jmp</td>
<td>Reset Handler</td>
</tr>
<tr>
<td>$0002</td>
<td>jmp</td>
<td>EXT_INT0 ; IRQ0 Handler</td>
</tr>
<tr>
<td>$0004</td>
<td>jmp</td>
<td>EXT_INT1 ; IRQ1 Handler</td>
</tr>
<tr>
<td>$0006</td>
<td>jmp</td>
<td>EXT_INT2 ; IRQ2 Handler</td>
</tr>
<tr>
<td>$0008</td>
<td>jmp</td>
<td>EXT_INT3 ; IRQ3 Handler</td>
</tr>
<tr>
<td>$000A</td>
<td>jmp</td>
<td>EXT_INT4 ; IRQ4 Handler</td>
</tr>
<tr>
<td>$000C</td>
<td>jmp</td>
<td>EXT_INT5 ; IRQ5 Handler</td>
</tr>
<tr>
<td>$000E</td>
<td>jmp</td>
<td>EXT_INT6 ; IRQ6 Handler</td>
</tr>
<tr>
<td>$0010</td>
<td>jmp</td>
<td>EXT_INT7 ; IRQ7 Handler</td>
</tr>
<tr>
<td>$0012</td>
<td>jmp</td>
<td>TIME_COMP ; Timer2 Compare Handler</td>
</tr>
<tr>
<td>$0014</td>
<td>jmp</td>
<td>TIME_OVF ; Timer2 Overflow Handler</td>
</tr>
<tr>
<td>$0016</td>
<td>jmp</td>
<td>TIM2_COMP ; Timer2 Compare Handler</td>
</tr>
<tr>
<td>$0018</td>
<td>jmp</td>
<td>TIM2_OVF ; Timer2 Overflow Handler</td>
</tr>
<tr>
<td>$001A</td>
<td>jmp</td>
<td>TIM1_COMPA ; Timer1 CompareA Handler</td>
</tr>
<tr>
<td>$001C</td>
<td>jmp</td>
<td>TIM1_OVF ; Timer1 Overflow Handler</td>
</tr>
<tr>
<td>$001E</td>
<td>jmp</td>
<td>TIM1_COMPB ; Timer1 CompareB Handler</td>
</tr>
<tr>
<td>$0020</td>
<td>jmp</td>
<td>TIM1_OVF ; Timer1 Overflow Handler</td>
</tr>
<tr>
<td>$0022</td>
<td>jmp</td>
<td>TIM0_COMP ; Timer0 Compare Handler</td>
</tr>
<tr>
<td>$0024</td>
<td>jmp</td>
<td>TIM0_OVF ; Timer0 Overflow Handler</td>
</tr>
<tr>
<td>$0026</td>
<td>jmp</td>
<td>SPI_STC ; SPI Transfer Complete Handler</td>
</tr>
<tr>
<td>$0028</td>
<td>jmp</td>
<td>UART_TXC ; UART TX Complete Handler</td>
</tr>
<tr>
<td>$002A</td>
<td>jmp</td>
<td>ADC ; ADC Converter Complete Handler</td>
</tr>
<tr>
<td>$002C</td>
<td>jmp</td>
<td>EE_RDY ; EEPROM Ready Handler</td>
</tr>
<tr>
<td>$0030</td>
<td>ldi</td>
<td>r16, high(RAMEND); Main program start</td>
</tr>
<tr>
<td>$0031</td>
<td>out</td>
<td>SPI, r16</td>
</tr>
<tr>
<td>$0032</td>
<td>ldi</td>
<td>r16, low(RAMEND)</td>
</tr>
<tr>
<td>$0033</td>
<td>out</td>
<td>SPI, r16</td>
</tr>
<tr>
<td>$0034</td>
<td></td>
<td>&lt;instr&gt; xxx</td>
</tr>
</tbody>
</table>

Reset Sources

The ATmega603/103 has three sources of reset:

- **Power-On Reset**: The MCU is reset when the supply voltage is below the power-on reset threshold ($V_{POT}$).
- **External Reset**: The MCU is reset when a low level is present on the RESET pin for more than 50 ns.
- **Watchdog Reset**: The MCU is reset when the Watchdog timer period expires and the Watchdog is enabled.

During reset, all I/O registers except the MCU Status register are then set to their initial values, and the program starts execution from address $0000$. The instruction placed in address $0000$ must be a JMP - absolute jump instruction to the reinitialization routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 23 shows the reset logic. Table 6 defines the timing and electrical parameters of the reset circuitry.

### Table 6. Timing Parameters

<table>
<thead>
<tr>
<th>Source</th>
<th>PORF</th>
<th>EXTRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power-On Reset</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>External Reset</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Watchdog Reset</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

To make use of these bits to identify a reset condition, the user software should direct both the PORF and EXTRF bits early as possible in the program. Checking the PORF and EXTRF values is done before the bit is cleared. If the bit cleared before an external or watchdog reset occurs, the source of reset can be found by using the following truth table:

### Table 9. Reset Source Identification

<table>
<thead>
<tr>
<th>EXTRF</th>
<th>PORF</th>
<th>Reset Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Watchdog Reset</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Power-on Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>External Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Power-on Reset</td>
</tr>
</tbody>
</table>

**MCU Status Register - MCUSR**

The MCU Status Register provides information on which reset source caused an MCU reset.

- **Bit 7 - Res**: Reserved Bits
  These bits are reserved bits in the ATmega603/103 and always read as zero.

- **Bit 6 - EXTRF**: Power-on Reset Flag
  After a power-on reset, this bit is undefined (X). It will be set by an external reset. A watchdog reset will leave this bit unchanged.

- **Bit 5 - PORF**: Power-on Reset Flag
  This bit is set by a power-on reset. A watchdog reset or an external reset will leave this bit unchanged.

To summarize, the following table shows the value of these two bits after the three modes of reset:

### Table 8. PORF and EXTRF Values after Reset

<table>
<thead>
<tr>
<th>Source</th>
<th>EXTRF</th>
<th>PORF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power-On Reset</td>
<td>undefined</td>
<td>0</td>
</tr>
<tr>
<td>External Reset</td>
<td>1</td>
<td>unchanged</td>
</tr>
<tr>
<td>Watchdog Reset</td>
<td>unchanged</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 27. Watchdog Reset During Operation
Interrupt Handling

The ATmega603/103 has two dedicated 8-bit Interrupt Mask control registers; EIMSK - External Interrupt Mask register and TIMSK - Timer/Counter Interrupt Mask register. In addition, other enable and mask bits can be found in the pin change control registers.

When an interrupt occurs, the Global Interrupt Enable bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction - RETI - is executed.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hardware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing logic one to the flag position(s) to be cleared.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priority. Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition is active.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

External Interrupt Flag Register - EIFR

<table>
<thead>
<tr>
<th>Bit 6..0 - INTF7 - INTF4: External Interrupt 7-4 Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read/Write R/W R/W R/W R/W R/W R/W R/W R/W R/W</td>
</tr>
<tr>
<td>Initial value 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

- **Bit 7 - TOIE2:** Timer/Counter2 Overflow Interrupt Enable
- **Bit 6 - TOIE1:** Timer/Counter1 Overflow Interrupt Enable
- **Bit 5 - TICIE1:** Timer/Counter1 Input Capture/Interrupt Enable
- **Bit 4 - OCIE2:** Timer/Counter2 Compare Match Interrupt Enable
- **Bit 3 - OCIE1A:** Timer/Counter1 Compare Match A Interrupt Enable
- **Bit 2 - OCIE1B:** Timer/Counter1 Compare Match B Interrupt Enable
- **Bit 1 - OCIE0:** Timer/Counter0 Compare Match Interrupt Enable
- **Bit 0 - OCIE1:** Timer/Counter1 Input Capture/Interrupt Enable

When the TOIE2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt (at vector $0002$) is executed if a Compare Match in Timer/Counter2 occurs, i.e., when the OCF2 bit is set in the Timer/Counter2 Overflow Flag Register - TOV2F.

When the TOIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $0004$) is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set in the Timer/Counter1 Overflow Flag Register - TOV1F.

When the TICIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt (at vector $0016$) is executed if a capture-triggering event occurs on T29, PDI4(C1), i.e., when the ICF1 bit is set in the Timer/Counter1 Input Capture Flag Register - ICF1F.

When the OCIE1A bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the Timer/Counter1 Compare Match A Event Interrupt is enabled. The corresponding interrupt (at vector $0018$) is executed if a Compare Match in Timer/Counter1 occurs, i.e., when the OCF1A bit is set in the Timer/Counter1 Output Compare Flag Register - OCF1AF.
• Bit 3 - OCIE1B: Timer/Counter1 Output Compare B Match Interrupt Enable
When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare B Match interrupt is enabled. The corresponding interrupt (at vector $001A$) is executed if a Compare match in Timer/Counter occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

• Bit 2 - TOIE1: Timer/Counter1 Overflow Interrupt Enable
When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $001C$) is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set in the Timer/Counter1 Interrupt Flag Register - TIFR.

• Bit 1 - OCF0: Output Compare Flag 0
When the OCF0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt (at vector $001E$) is executed if a Compare match in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter0 Interrupt Flag Register - TIFR.

• Bit 0 - TOV0: Timer/Counter0 Overflow Flag
When the TOV0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $0020$) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter0 Interrupt Flag Register - TIFR.

Timer/Counter Interrupt Flag Register - TIFR

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S/3: (S3)</td>
<td>OCF2</td>
<td>TOV2</td>
<td>ICF1</td>
<td>OCF1A</td>
<td>OCF1B</td>
<td>TOV1</td>
<td>OCF0</td>
<td>TOV0</td>
</tr>
</tbody>
</table>

Initial value: 0

• Bit 7 - OCIF2: Output Compare Flag 2
When the TOV2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare match interrupt is enabled. The corresponding interrupt (at vector $0024$) is executed if a Compare match in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter2 Interrupt Flag Register - TIFR.

• Bit 6 - TOV2: Timer/Counter2 Overflow Flag
When the TOV2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt (at vector $0026$) is executed if an overflow in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter2 Interrupt Flag Register - TIFR.

• Bit 5 - ICF1: Input Capture Flag 1
When the TOV2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture interrupt is enabled. The corresponding interrupt (at vector $0028$) is executed if an input capture occurs in Timer/Counter1, i.e., when the ICF1 bit is set in the Timer/Counter1 Input Capture Flag Register - TIFR.

• Bit 4 - OCF1A: Output Compare Flag 1A
When the TOV2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Output Compare A interrupt is enabled. The corresponding interrupt (at vector $002A$) is executed if an output compare occurs in Timer/Counter1, i.e., when the OCF1A bit is set in the Timer/Counter1 Output Compare Flag Register - TIFR.

• Bit 3 - OCIF1B: Output Compare Flag 1B
When the TOV2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Output Compare B interrupt is enabled. The corresponding interrupt (at vector $002C$) is executed if an output compare occurs in Timer/Counter1, i.e., when the OCIF1B bit is set in the Timer/Counter1 Output Compare Flag Register - TIFR.

Interrupt Response Time
The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. 4 clock cycles after the interrupt has been set, the program vector address for the actual interrupt handling routine is executed. During this clock cycle period, the program counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented. The vector is normally a jump to the interrupt routine, and this jump takes 3 clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. A return from an interrupt handling routine (same as for a subroutine call routine) takes 4 clock cycles. During these 4 clock cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2. When an interrupt occurs, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Sleep Modes
To enter any of the three sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. The SM1 and SM0 bits in the MCUCR register select which sleep mode (Idle, Power Down, or Power Save) will be activated by the SLEEP instruction, see Table 4.

Idle Mode
When the SM1/SM0 bits are set to 00, the SLEEP instruction makes the MCU enter the Idle Mode, stopping the CPU from executing instructions. This enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow or a UART Receive Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the analog comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status register - ACSR. This reduces power consumption in Idle Mode. When the MCU wakes up from idle mode, the CPU starts program execution immediately.
**Power Down Mode**

When the SM1/SM0 bits are set to 10, the SLEEP instruction makes the MCU enter the Power Down Mode. In this mode the external oscillator is stopped, while the external interrupts and the Watchdog (if enabled) continue operating. Only external reset, a watchdog reset (if enabled), or an external level interrupt can wake up the MCU.

Note that if a level triggered interrupt is used for wake-up from Power Down Mode, the changed level must be held some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the watchdog oscillator clock, and if the input has the required level during this time, the MCU will wake up. The period of the watchdog oscillator is 1 us (nominal) at 5.0V and 25°C. The frequency of the watchdog oscillator is voltage dependent, as shown in section “Typical characteristics” on page 110.

When waking up from Power Down Mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined as the time that elapses from the SLEEP instruction until the clock has become stable.

If the wake-up condition disappears before the MCU wakes up and starts to execute, e.g. a low level on PD is not held long enough, the interrupt causing the wake-up will not be executed.

**Power Save Mode**

When the SM1/SM0 bits are 11, the SLEEP instruction makes the MCU enter the Power Save Mode. This mode is identical to Power Down, with one exception:

- If Timer/Counter0 is clocked asynchronously, i.e. the AS0 bit in ASSR is set, Timer/Counter0 will run during sleep. In addition to the Power Down wake-up sources, the device can also wake up from either Timer Overflow or Output Compare event from Timer/Counter0 if the corresponding Timer/Counter0 interrupt enable bits are set in TCR1.

To ensure that the interrupt routine is executed when waking up, the global interrupt enable bit in SREG must be set.

When waking up from Power Save Mode by an external interrupt, 2 instruction cycles are executed before the interrupt flags are updated. When waking up by the asynchronous timer, 3 instruction cycles are executed before the flags are updated. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and the interrupt routine has not started yet.

**Timer/Counters**

The ATmega603/103 provides three general purpose Timer/Counters - two 8-bit T/Cs and one 16-bit T/C. Timer/Counters can optionally be asynchronously clocked from an external oscillator. This oscillator is optimized for use with a 32.768 k Hz crystal, enabling use of Timer/Counter0 as a Real Time Clock (RTC). Timer/Counter0 has its own prescaler.

Timer/Counters 1 and 2 have individual prescaling selection from the same 10-bit prescaling timer. These Timer/Counters can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.
The clock source for Timer/Counter0 prescaler is named PCK0. PCK0 is by default connected to the main system clock CK. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. By setting the AS0 bit in ASSR, Timer/Counter 0 prescaler is asynchronously clocked from the TOSC1 pin. This enables use of Timer/Counter0 as a Real Time Clock (RTC). A crystal can be connected between the TOSC1 and TOSC2 pins to serve as an independent clock source for Timer/Counter0. This oscillator is optimized for use with a 32.768 kHz crystal.

8-bit Timer/Counters TC0 and TC2

Figure 30 shows the block diagram for Timer/Counter0.

Figure 30. Timer/Counter0 Block Diagram

Note: Figure 31 shows the block diagram for Timer/Counter2.

The 8-bit Timer/Counter0 can select clock source from PCK0 or prescaled PCK0. The 8-bit Timer/Counter2 can select clock source from CK, prescaled CK, or an external pin. Both Timer/Counters can be stopped as described in the specification for the Timer/Counter Control Registers - TCCR0 and TCCR2.

The different status flags (overflow, compare match and capture event) are found in the Timer/Counter Interrupt Flag Register - TIFR. Control signals are found in the Timer/Counter Control Registers - TCCR0 and TCCR2. The interrupt enable/disable settings are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter2 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To ensure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counters feature a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make these units useful for lower speed functions or exact timing functions with infrequent actions.

Both Timer/Counters support two Output Compare functions using the Output Compare Registers - OCR0 and OCR2. The data source to be compared to the Timer/Counter contents. The Output Compare function includes optional clearing the counter on compare match, and action on the Output Compare Pins - PB4(OC0/PWM0) and PB6(OC2/PWM2) on compare match.

Timer/Counter0 and 2 can also be used as 8-bit Pulse Width Modulators. In this mode the Timer/Counter and the output compare register serve as a glitch-free, stand-alone PWM with centered pulses. Refer to page 40 for a detailed description of this function.
**Timer/Counter0 Control Register - TCCR0**

<table>
<thead>
<tr>
<th>Bit 7-0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>PWM0</td>
</tr>
<tr>
<td>6</td>
<td>COM01</td>
</tr>
<tr>
<td>5</td>
<td>COM00</td>
</tr>
<tr>
<td>4</td>
<td>CTC0</td>
</tr>
<tr>
<td>3</td>
<td>CS02</td>
</tr>
<tr>
<td>2</td>
<td>CS01</td>
</tr>
<tr>
<td>1</td>
<td>CS00</td>
</tr>
<tr>
<td>0</td>
<td>TCCR0</td>
</tr>
</tbody>
</table>

**Initial value:** 0 0 0 0 0 0 0 0

**Timer/Counter2 Control Register - TCCR2**

<table>
<thead>
<tr>
<th>Bit 7-0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>PWM2</td>
</tr>
<tr>
<td>6</td>
<td>COM21</td>
</tr>
<tr>
<td>5</td>
<td>COM20</td>
</tr>
<tr>
<td>4</td>
<td>CTC2</td>
</tr>
<tr>
<td>3</td>
<td>CS22</td>
</tr>
<tr>
<td>2</td>
<td>CS21</td>
</tr>
<tr>
<td>1</td>
<td>CS20</td>
</tr>
<tr>
<td>0</td>
<td>TCCR2</td>
</tr>
</tbody>
</table>

**Initial value:** 0 0 0 0 0 0 0 0

**Note:** n = 0 or 2

In PWM mode, these bits have a different function. Refer to Table 14 for a detailed description.

**Bit 7 - Res: Reserved Bit**

This bit is a reserved bit in the ATmega603/103 and always reads as zero.

**Bit 6 - PWM0 / PWM2: Pulse Width Modulator Enable**

When set (one) this bit enables PWM mode for Timer/Counter0 or Timer/Counter2. This mode is described on page 40.

**Bits 5,4 - COM01, COM00 / COM21, COM20: Compare Output Mode, bits 1 and 0**

The COMn1 and COMn0 control bits determine any output pin action following a compare match in Timer/Counter. Any output pin actions affect pins PB4(OC0/PWM0) or PB7(OC2/PWM2). Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 11.

**Table 11. Compare Mode Select**

<table>
<thead>
<tr>
<th>COMn1</th>
<th>COMn0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Timer/Counter disconnected from output pin OCn/PWMn.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Toggle the OCn/PWMn output line.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Clear the OCn/PWMn output line (to zero).</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Set the OCn/PWMn output line (to one).</td>
</tr>
</tbody>
</table>

**Table 12. Timer/Counter0 Prescale Select**

<table>
<thead>
<tr>
<th>CS02</th>
<th>CS01</th>
<th>CS00</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Timer/Counter0 is stopped.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>PCO0/32</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>PCO0/8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>PCO0/32/64</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>PCO0/128</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>PCO0/256</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>PCO0/1024</td>
</tr>
</tbody>
</table>

**Table 13. Timer/Counter2 Prescale Select**

<table>
<thead>
<tr>
<th>CS22</th>
<th>CS21</th>
<th>CS20</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Timer/Counter2 is stopped.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>CK</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>CK/8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>CK/64</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>CK/256</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>CK/1024</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>External Pin PD7(T2), falling edge</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>External Pin PD7(T2), rising edge</td>
</tr>
</tbody>
</table>

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CPU clock. If the external pin modes are used for Timer/Counter2, transitions on PD7(T2) will clock the counter even if pin is configured as an output. This feature can give the user SW control of the counting.

**Timer/Counter0 - TCNT0**

<table>
<thead>
<tr>
<th>Bit 7-0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>MSB</td>
</tr>
<tr>
<td>6</td>
<td>MSB</td>
</tr>
<tr>
<td>5-0</td>
<td>LSB</td>
</tr>
</tbody>
</table>

**Initial value:** 0 0 0 0 0 0 0

**Timer/Counter2 - TCNT2**

<table>
<thead>
<tr>
<th>Bit 7-0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>MSB</td>
</tr>
<tr>
<td>6</td>
<td>MSB</td>
</tr>
<tr>
<td>5-0</td>
<td>LSB</td>
</tr>
</tbody>
</table>

**Initial value:** 0 0 0 0 0 0

These 8-bit registers contain the value of the Timer/Counters.

Both Timer/Counters are realized as up or up/down (in PWM mode) counters with read and write access. If Timer/Counter is written to and a clock source is selected, it continues counting in the timer clock cycle after it is preset w the written value.
The output compare registers are 8-bit read/write registers. The Timer/Counter Output Compare Registers contain the data to be continuously compared with the Timer/Counter counts to the OCR value. A software write that sets the Timer/Counter and Output Compare Register to the same value does not generate a compare match. A compare match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Timer/Counter 0 and 2 in PWM mode
When the PWM mode is selected, the Timer/Counter and the Output Compare Register - OCR0 or OCR2 form an 8-bit free-running, glitch-free and phase correct PWM with outputs on the PB4(OC0/PWM0) or PB7(OC2/PWM2) pin. Timer/Counter acts as an up-down counter, counting up from $00$ to $FF$, where it turns and counts down again to $00$ before the cycle is repeated. When the counter value matches the contents of the Output Compare register, either PB4(OC0/PWM0) or PB7(OC2/PWM2) pin is set or cleared according to the settings of the COM01/COM00 and COM21/COM20 bits in the Timer/Counter Control Registers TCCR0 and TCCR2. Refer to Table 14 for details.

### Table 14. Compare Mode Select in PWM Mode

<table>
<thead>
<tr>
<th>COM01</th>
<th>COM00</th>
<th>Effect on Compare/PWM Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>Not connected</td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>Not connected</td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).</td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).</td>
<td></td>
</tr>
</tbody>
</table>

Note: n = 0 or 2

In PWM mode, the Output Compare register is transferred to a temporary location when written. The value latched when the Timer/Counter reaches $FF$. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR0 or OCR2 write. See Figure 32 for an example.

### Figure 32. Effects on Unsynchronized OCR Latching

- **Synchronized OCR Latch**
- **Unsynchronized OCR Latch**
- **Glitch**

During the time between the write and the latch operation, a read from OCR0 or OCR2 will read the contents of the temporary location. This means that the most recently written value always will read out of OCR0/2. When the OCR register (not the temporary register) is updated to $00$ or $FF$, the PWM output changes to low or high immediately according to the settings of COM21/COM20 or COM11/COM10. This is shown in Table 15:

### Table 15. PWM Outputs OCRn = $00$ or $FF$

<table>
<thead>
<tr>
<th>COMn1</th>
<th>COMn0</th>
<th>OCRn</th>
<th>Output PWMn</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>0</td>
<td>$00$</td>
<td>L</td>
</tr>
<tr>
<td>1 0</td>
<td>1</td>
<td>$FF$</td>
<td>H</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>$00$</td>
<td>H</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
<td>$FF$</td>
<td>L</td>
</tr>
</tbody>
</table>

Note: n = 0 or 2

In PWM mode, the Timer Overflow Flag, TOV0 or TOV2, is set when the counter advances from $00$. Timer Overflow Interrupt0 and 2 operates exactly as in normal Timer/Counter mode, i.e., it is executed when TOV0 or TOV2 is set, but Timer Overflow Interrupt and global interrupts are enabled. This does also apply to the Timer Output Compare flags interrupts.

The frequency of the PWM will be Timer Clock Frequency divided by 510.
Asynchronous Operation of Timer/Counter0

When Timer/Counter0 operates synchronously, all operations and timing are identical to Timer/Counter2. During asynchronous operation, however, some considerations must be taken:

- **Bit 1 - OCR0UB: Output Compare Register0 Update Busy**
  - When Timer/Counter0 operates asynchronously and OCR0 is written, this bit becomes set (one). When OCR0 has been updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates that OCR0 is ready to be updated with a new value.
  - When Timer/Counter0 operates asynchronously and TCCR0 is written, this bit becomes set (one). When TCCR0 has been updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates that TCCR0 is ready to be updated with a new value.
  - If a write is performed to any of the three Timer/Counter0 registers while its update busy flag is set (one), the updated value might get corrupted and cause an unintended interrupt to occur.

- **Bit 0 - TCR0UB: Timer/Counter Control Register0 Update Busy**
  - During asynchronous operation, the synchronization of the interrupt flags for the asynchronous timer takes 3 processes plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value. When reading TCNT0, OCR0, and TCCR0, there is a difference in result. When reading TCNT0, the actual timer value read. When reading OCR0 or TCCR0, the value in the temporary storage register is read.

**Asynchronous Operation of Timer/Counter0**

- When Timer/Counter0 operates synchronously, all operations and timing are identical to Timer/Counter2. During asynchronous operation, however, some considerations must be taken:

  1. Disable the timer 0 interrupts OCIE0 and TCIE0.
  2. Select clock source by setting ASO as appropriate.
  3. When writing to the register, TCNT0, OCR0, or TCCR0, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means that e.g., writing to TCNT0 does not disturb an OCR0 write in progress. To detect that a transfer to the destination register has taken place, a Asynchronous Status Register - ASSR has been implemented.
  4. When writing to any of the registers TCNT0, OCR0, or TCCR0, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means that e.g., writing to TCNT0 does not disturb an OCR0 write in progress. To detect that a transfer to the destination register has taken place, a Asynchronous Status Register - ASSR has been implemented.
  5. When entering Power Save mode after having written to TCNT0, OCR0 or TCCR0, the user must wait until the write register has been updated before using Timer/Counter0. Otherwise, the MCU will go to sleep before the changes have had any effect. This is extremely important if the output compare interrupt is used to wake up the device. Output compare is disabled during write to OCR0 or TCNT0. If the write cycle is not finished (i.e., the user goes sleep before the OCR0UB bit returns to zero), the device will never get a compare match and the MCU will not wake up.

- **When using Timer/Counter0 for wake-up from Power Save mode**: The interrupt logic needs one TOSC1 cycle to get reset. If the time between wake up and entering Power Save mode is less than one TOSC1 cycle, the interrupt will not occur and the device will fail to wake up. The user should wait until the corresponding Update Busy flag in ASSR returns to zero.

**16-bit Timer/Counter1**

Figure 33 shows the block diagram for Timer/Counter1.

- **Description of wake up from power save mode**: When the timer is clocked asynchronously, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. To execute the corresponding Timer/Counter0 interrupt routine, the global interrupt bit in SREG must have been set. Otherwise, the timer will still wake up from power down, continues to execute the sleep command. The interrupt flags are updated 3 processor cycles after the processor clock has started. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and the interrupt routine has not started yet.

- **Synchronization of the interrupt flags for the asynchronous timer takes 3 processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the interrupt flag. The output compare pin is changed on the timer clock, and is not synchronized to the processor clock.**
Figure 33. Timer/Counter1 Block Diagram

Timer/Counter1 can also be used as an 8, 9 or 10-bit Pulse Width Modulator. In this mode the counter and the OCR1A/OCR1B registers serve as a dual glitch-free stand-alone PWM with centered pulses. Refer to page 49 for detailed description on this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Capture Register - ICR1, triggered by an external event on the Input Capture Pin - PD4/(IC1). The actual capture event setting is defined by the Timer/Counter1 Control Register - TCCR1B. In addition, the Analog Comparator can be set to trigger the Input Capture. Refer to the paragraph, "The Analog Comparator", for details on this. The ICP pin logic is shown in Figure 34.

Table 16. Compare 1 Mode Select

<table>
<thead>
<tr>
<th>COMX1</th>
<th>COMX0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td></td>
<td>Timer/Counter1 disconnected from output pin OC1X</td>
</tr>
<tr>
<td>0 1</td>
<td></td>
<td>Toggle the OC1X output line</td>
</tr>
<tr>
<td>1 0</td>
<td></td>
<td>Set the OC1X output line to zero</td>
</tr>
<tr>
<td>1 1</td>
<td></td>
<td>Set the OC1X output line to one</td>
</tr>
</tbody>
</table>

Note: X = A or B. In PWM mode, these bits have a different function. Refer to Table 17 for a detailed description.

Table 17. PWM Mode Select

<table>
<thead>
<tr>
<th>PWM11</th>
<th>PWM10</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td></td>
<td>PWM operation of Timer/Counter1 is disabled</td>
</tr>
<tr>
<td>0 1</td>
<td></td>
<td>Timer/Counter1 is an 8-bit PWM</td>
</tr>
<tr>
<td>1 0</td>
<td></td>
<td>Timer/Counter1 is a 9-bit PWM</td>
</tr>
<tr>
<td>1 1</td>
<td></td>
<td>Timer/Counter1 is a 10-bit PWM</td>
</tr>
</tbody>
</table>

If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples, all 4 must be equal 1 to activate the capture flag.

Timer/Counter1 Control Register A - TCCR1A

<table>
<thead>
<tr>
<th>Bit</th>
<th>COM1A1</th>
<th>COM1A0</th>
<th>COM1B1</th>
<th>COM1B0</th>
<th>PWM11</th>
<th>PWM10</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: X = A or B.

In PWM mode, these bits have a different function. Refer to Table 17 for a detailed description.
### Timer/Counter1 Control Register B - TCCR1B

<table>
<thead>
<tr>
<th>Bit 7 (ICNC1)</th>
<th>Bit 6 (ICES1)</th>
<th>Bit 5, 4 (Res)</th>
<th>Bit 3 (CTC1)</th>
<th>Bit 2, 1, 0 (CS12, CS11, CS10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Bit 7 - ICNC1:** Input Capture1 Noise Canceler (4 CKs)
  - When the ICNC1 bit is cleared (zero), the input capture trigger noise canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the input capture pin PD4(IC1) as specified. When the ICNC1 bit is one, four successive samples are measures on PD4(IC1), and all samples must be high/low according to the input capture trigger specification in the ICES1 bit. The actual sampling frequency is XTAL clock frequency.

- **Bit 6 - ICES1:** Input Capture1 Edge Select
  - While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register - ICR1 - on the falling edge of the input capture pin - PD4(IC1). While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the Input Capture Register - ICR1 - on the rising edge of the input capture pin - PD4(IC1).

- **Bits 5, 4 - Res:** Reserved bits
  - These bits are reserved bits in the ATmega603/103 and always read zero.

- **Bit 3 - CTC1:** Clear Timer/Counter1 on Compare Match
  - When the CTC1 control bit is set (one), the Timer/Counter1 is reset to $0000$ in the clock cycle after a compare match occurs. Timer/Counter1 continues counting and is unaffected by a complete match. The compare match is detected in the CPU clock cycle following the match, this function will behave differently when a prescaler higher than 1 is used for the timer. When a prescaling of 1 is used and the compareA register is set to C, the timer counts as follows if CTC1 is set:
  - $0 | C-2 | C-1 | C | 0 | 1 | ...$
  - When the prescaler is set to divide by 8, the timer will count like this:
  - $0 | C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0 | ...$

- **Bits 2, 1, 0 - CS12, CS11, CS10:** Clock Select, bit 2, 1 and 0
  - The clock select bits 2, 1 and 0 define the prescaling source of Timer/Counter1.

#### Table 18. Clock 1 Prescale Select

<table>
<thead>
<tr>
<th>CS12</th>
<th>CS11</th>
<th>CS10</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Stop, the Timer/Counter1 is stopped.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>CK</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>CK/8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>CK/64</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>CK/256</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>CK/1024</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>External Pin T1, falling edge</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>External Pin T1, rising edge</td>
</tr>
</tbody>
</table>

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CPU clock. If the external pins are used for Timer/Counter1, transitions on PD6/(T1) will clock the counter even if it is configured as an input. This feature can activate the user SW control of the counting.

---

### Timer/Counter1 - TCNT1H and TCNT1L

<table>
<thead>
<tr>
<th>Bit 7 (S2D)</th>
<th>Bit 6 (S2C)</th>
<th>Bit 5 (MSB)</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low bytes are read and written simultaneously, when the CPU accesses these registers, the access is performed using an 8-bit or 16-bit memory operation. This temporary register is also used when accessing OCR1A, OCR1B and ICR1. If the microcontroller and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

- **TCNT1 Timer/Counter1 Write:**
  - When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the byte in the TEMP register, and all 16 bits written to the TCNT1 Timer/Counter1 register simultaneously. Consequently, the high byte of TCNT1H must be accessed first for a full 16-bit register write operation. When using Timer/Counter1 as an 8-bit timer, it is sufficient to write the 8-bit only.

- **TCNT1 Timer/Counter1 Read:**
  - When the CPU reads the low byte TCNT1L, the data of TCNT1L is sent to the CPU and the data of the high byte of TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register read operation. When using Timer/Counter1 as an 8-bit timer, it is sufficient to read the low byte only.

The Timer/Counter1 is realized as an up or down (in PWM mode) counter with read access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the clock cycle after it is preset with the written value.
Timer/Counter1 Output Compare Register - OCR1AH and OCR1AL

<table>
<thead>
<tr>
<th>Bit</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$28$</td>
<td>MSB</td>
<td>OCR1AH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$29$</td>
<td>MSB</td>
<td>OCR1AL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Read/Write: R/W
Initial value: 00000000

The output compare registers are 16-bit read-write registers.

The Timer/Counter1 Output Compare Registers contain the data to be continuously compared with Timer/Counter Actions on compare matches are specified in the Timer/Counter1 Control and Status register. A compare match does not occur if Timer/Counter1 counts to the OCR value. A software write that sets TCNT1 and OCR1A or OCR1B to the same value does not generate a compare match.

A compare match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Since the Output Compare Registers - OCR1A and OCR1B - are 16-bit registers, a temporary register TEMP is used when OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1A or OCR1B, the data is temporarily stored in the TEMP register. The high byte OCR1AH or OCR1BH is written first for a full 16-bit register write operation.

The TEMP register is also used when accessing TCNT1, OCR1A and OCR1B. If the main program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program.

Timer/Counter1 Input Capture Register - ICR1H and ICR1L

<table>
<thead>
<tr>
<th>Bit</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$27$ ($37$)</td>
<td>MSB</td>
<td>ICR1H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$26$ ($36$)</td>
<td>MSB</td>
<td>ICR1L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Read/Write: R/W
Initial value: 00000000

The input capture register is a 16-bit read-only register.

When the rising or falling edge (according to the input capture edge setting - ICES1) of the signal at the input capture pin (PB4/IC1) is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register - ICR1. At the same time, the input capture flag - ICF1 - is set (one).
During the time between the write and the latch operation, a read from OCR1A or OCR1B will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1A/B when the OCR1A/OCR1B contains $0000 or TOP, the output OC1A/OC1B is updated to low or high on the next compare match according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 21.

Table 21. PWM Outputs OCR1X = $0000 Or TOP

<table>
<thead>
<tr>
<th>COM1X1</th>
<th>COM1X0</th>
<th>OCR1X</th>
<th>Output OC1X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>$0000</td>
<td>L</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>TOP</td>
<td>H</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$0000</td>
<td>H</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>TOP</td>
<td>L</td>
</tr>
</tbody>
</table>

Note: X = A or B

In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter advances from $0000. Timer Overflow Interrupt and global interrupts are enabled. This does also apply to the Timer Output Compare flags and interrupts.

Watchdog Timer

The Watchdog Timer is locked from a separate on-chip oscillator. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted as shown in Table 22. See characterization data for typical values at other Vcc levels. The WDR - Watchdog Reset - instruction resets the Watchdog Timer. From the Watchdog is reset, eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the ATmega603/103 resets and executes from the reset vector. For timing details on the Watchdog reset, refer to page 28.

To prevent unintentional disabling of the watchdog, a special turn-off procedure must be followed when the watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details.

Table 22. Watchdog Timer Control Register - WDTCSR

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>WDTCSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$21 (S41)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>WDTOE</td>
<td>WDE</td>
<td>WDP2</td>
<td>WDP1</td>
<td>WDP0</td>
<td>WDTCR</td>
</tr>
</tbody>
</table>

Read/Write: R R R R/R W/R W/R W/R W/R

- Bits 7-5 - Reserved bits
- Bit 4 - WDTOE: Watch Dog Turn Off Enable
- Bit 3 - WDE: Watch Dog Enable
- Bit 2-0 - WDP2, WDP1, WDP0: Watch Dog Timer Prescaler 2, 1 and 0

The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Time-out Periods are shown in Table 22.

Figure 35. Effects on Unsynchronized OCR1 Latching

Figure 36. Watchdog Timer
**Table 22. Watch Dog Timer Prescale Select**

<table>
<thead>
<tr>
<th>WDP2</th>
<th>WDP1</th>
<th>WDP0</th>
<th>Number of IWDOS Oscillator cycles</th>
<th>Typical time-out at Vcc = 3.0V</th>
<th>Typical time-out at Vcc = 5.0V</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16K cycles</td>
<td>47 ms</td>
<td>15 ms</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>32K cycles</td>
<td>94 ms</td>
<td>30 ms</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>64K cycles</td>
<td>0.19 s</td>
<td>0.06 s</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>128K cycles</td>
<td>0.38 s</td>
<td>0.12 s</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>256K cycles</td>
<td>0.75 s</td>
<td>0.24 s</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>512K cycles</td>
<td>1.5 s</td>
<td>0.49 s</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1,024K cycles</td>
<td>3.0 s</td>
<td>0.97 s</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2,048K cycles</td>
<td>6.0 s</td>
<td>1.9 s</td>
</tr>
</tbody>
</table>

**Note:** The frequency of the watchdog oscillator is voltage dependent as shown in the Electrical Characteristics section. The WDR - Watchdog Reset - instruction should always be executed before the Watchdog Timer is enabled. This ensures that the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without reset, the watchdog timer may not start counting from zero.

**EEPROM Read/Write Access**

The EEPROM access registers are accessible in the I/O space.

- The access time is in the range of 2.5 - 4ms, depending on the Vcc voltages. A self-timing function lets the user software detect when the next byte can be written. A special EEPROM Ready interrupt can be set to trigger when the EEPROM is ready to accept new data.

In order to prevent un-intentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM control register for details on this.

When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed. When read, the CPU is halted for 4 clock cycles.

**EEPROM Address Register - EEARH, EEARL**

- **Bits 7..0 - EEARH: EEPROM Address:**
  - For the EEPROM write operation, the EEARH register contains the address to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEARH contains the data read out from the EEPROM at the address given by EEAR.

**EEPROM Data Register - EEDR**

- **Bits 7..0 - EEDR: EEPROM Data:**
  - For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

**EEPROM Control Register - EECR**

- **Bits 7..4 - Res: Reserved bits**
  - These bits are reserved bits in the ATmega603/103 and will always be read as zero.

- **Bit 3 - EERIE: EEPROM Ready Interrupt Enable**
  - When the bit in SREG and EERIE are set (one), the EEPROM Ready Interrupt is enabled. When cleared (zero), the interrupt is disabled. The EEPROM Ready Interrupt constantly generates an interrupt request when EWEW is cleared (zero).

- **Bit 2 - EEMWE: EEPROM Master Write Enable**
  - The EEMWE bit determines whether setting EWEW to one causes the EEPROM to be written. When EEMWE is set, setting EWEW will write data to the EEPROM at the selected address. When EEMWE is zero, setting EWEW will have no effect.

- **Bit 1 - EEWE: EEPROM Write Enable**
  - The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. The EEWE bit must be set when the logical one is written to EEAR, otherwise no EEPROM write takes place. The following procedure should be followed when writing to EEPROM (the order of steps 2 and 3 is unessential):

1. Wait until EEWE becomes zero.
2. Write new EEPROM address to EEAR (optional).
3. Write new EEPROM data to EEDR (optional).
4. Write a logical one to the EEMWE bit in EECR.
5. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

Caution: An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR and EEDR registers will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt cleared during the last steps to avoid these problems.

When the write access time (typically 2.5ms at Vcc = 5V or 4ms at Vcc = 2.7V) has elapsed, the EEEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE is set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 - EERE: EEPROM Read Enable**
  - The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in EEAR register, the EEERE bit must be set. When the EEERE bit is cleared (zero) by hardware, requested data is found in EEAR register. The EEPROM read access takes one instruction and there is no need to poll the EERE bit. When EEERE is been set, the CPU is halted for four cycles before the next instruction is executed.

**EEPROM Data Register - EEDR**

- **Bits 7..0 - EEDR: EEPROM Data:**
  - For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

**EEPROM Control Register - EECR**

- **Bits 7..4 - Res: Reserved bits**
  - These bits are reserved bits in the ATmega603/103 and will always be read as zero.

- **Bit 3 - EERIE: EEPROM Ready Interrupt Enable**
  - When the bit in SREG and EERIE are set (one), the EEPROM Ready Interrupt is enabled. When cleared (zero), the interrupt is disabled. The EEPROM Ready Interrupt constantly generates an interrupt request when EWEW is cleared (zero).

- **Bit 2 - EEMWE: EEPROM Master Write Enable**
  - The EEMWE bit determines whether setting EWEW to one causes the EEPROM to be written. When EEMWE is set, setting EWEW will write data to the EEPROM at the selected address. When EEMWE is zero, setting EWEW will have no effect. When EEMWE has been set (one) by software, hardware clears the bit to zero after four clock cycles. See the description of the EEW bit for a EEPROM write procedure.

- **Bit 1 - EEWE: EEPROM Write Enable**
  - The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. The EEWE bit must be set when the logical one is written to EEAR, otherwise no EEPROM write takes place. The following procedure should be followed when writing to EEPROM (the order of steps 2 and 3 is unessential):

1. Wait until EEWE becomes zero.
2. Write new EEPROM address to EEAR (optional).
3. Write new EEPROM data to EEDR (optional).
4. Write a logical one to the EEMWE bit in EECR.
5. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

Caution: An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR and EEDR registers will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt cleared during the last steps to avoid these problems.

When the write access time (typically 2.5ms at Vcc = 5V or 4ms at Vcc = 2.7V) has elapsed, the EEEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE is set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 - EERE: EEPROM Read Enable**
  - The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in EEAR register, the EEERE bit must be set. When the EEERE bit is cleared (zero) by hardware, requested data is found in EEAR register. The EEPROM read access takes one instruction and there is no need to poll the EERE bit. When EEERE is set, the CPU is halted for four cycles before the next instruction is executed.
The user should poll the EEWE bit before starting the read operation. If a write operation is in progress when new data is written to the EEPROM I/O registers, the write operation will be interrupted, and the result is undefined.

**Prevent EEPROM Corruption**

During periods of low V_{CC}, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using the EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly if the supply voltage for executing instructions is too low.

EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This is best done by an external low V_{CC} protection circuit, often referred to as a Brown-Out Detector (BOD). Please refer to application note AVR 180 for design considerations regarding power-on reset and low voltage detection.
2. Keep the AVR core in Power Down Sleep Mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM registers from unintentional writes.
3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory can not be updated by the CPU, and will not be subject to corruption.

**Serial Peripheral Interface - SPI**

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega603/103 and peripheral devices or between several AVR devices. The ATmega603/103 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode (Slave Mode Only)

The interconnection between master and slave CPUs with SPI is shown in Figure 38. The PB1(SCK) pin is the clock output in the master mode and the clock input in the slave mode. Writing to the SPI data register of the master CPU starts a SPI clock generator, and the data written shifts out of the PB2(MOSI) pin and into the PB1(MISO) pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If the SPI Interrupt Enable bit (SPIE) in the SPCR register is set, an interrupt is requested. The Slave Select input, PB0(SS), is low to select individual slave SPI device. The two shift registers in the Master and the Slave can be considered as one distributed 16-bit shift register. This is shown in Figure 38. When data is shifted from the master to the slave, data is also shifted in the opposite direction, simultaneously. This means that during one shift cycle, data in the master and slave is interchanged.
Figure 38. SPI Master-Slave Interconnection

The system is single buffered in the transmit direction and double buffered in the receive direction. This means that characters to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received byte must be read from the SPI Data Register before the next byte has been completely shifted. Otherwise, the first byte is lost.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK, and SS pins is overridden according to the following Table:

Table 23. SPI Pin Overrides

<table>
<thead>
<tr>
<th>PIN</th>
<th>Direction, Master SPI</th>
<th>Direction, Slave SPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSI</td>
<td>User Defined</td>
<td>Input</td>
</tr>
<tr>
<td>MISO</td>
<td>Input</td>
<td>User Defined</td>
</tr>
<tr>
<td>SCK</td>
<td>User Defined</td>
<td>Input</td>
</tr>
<tr>
<td>SS</td>
<td>User Defined</td>
<td>Input</td>
</tr>
</tbody>
</table>

Note: See “Alternate Functions of Port B” on page 78 for a detailed description and how to define the direction of the user defined SPI pins.

SS Pin Functionality

When the SPI is configured as a master (MSTR in SPCR is set), the user can determine the direction of the SS pin. If SS is configured as an output, the pin is a general output pin which does not affect the SPI system. If SS is configured as input, it must be held high to ensure Master SPI operation. If the SS pin is driven low by peripheral circuitry when the SPI is configured as master with the SS pin defined as an input, the SPI system interprets this as another master selecting the SPI as a slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a slave. As a result of the SPI becoming a slave, the MOSI and SCK pins become inputs.
2. The SPIF flag in SPSR is set, and if the SPI interrupt is enabled and the I-bit in SREG is set, the interrupt routine will be executed.

Thus, when interrupt-driven SPI transmit is used in master mode, and there exists a possibility that SS is driven low, the interrupt routine should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable SPI master mode.

When the SPI is configured as a slave, the SS pin is always input. When SS is held low, the SPI is activated and M1 becomes an input. All other pins are inputs. When SS is driven high, all pins are inputs, and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the SS pin is brought high. If the SS pin is brought high during a transmission, the SPI will stop sending and receiving immediately and both data received and data sent must be considered as lost.

Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 39 and Figure 40.

Figure 39. SPI Transfer Format with CPHA = 0 and DORD = 0

Figure 40. SPI Transfer Format with CPHA = 1 and DORD = 0

Table 24. SPI Control Register - SPCR

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPCR</td>
<td>SPIE</td>
<td>SPE</td>
<td>DORD</td>
<td>MSTR</td>
<td>CPOL</td>
<td>CPHA</td>
<td>SPR1</td>
<td>SPR0</td>
</tr>
<tr>
<td>Read/Write</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
</tr>
<tr>
<td>Initial value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Bit 7 - SPIE: SPI Interrupt Enable
- Bit 6 - SPE: SPI Enable
- Bit 5 - DORD: Data Order
- Bit 4 - MSTR: Master or Slave
- Bit 3 - CPOL: Clock polarity
- Bit 2 - CPHA: Clock phase
- Bit 1 - SPR1: Slave select polarity
- Bit 0 - SPR0: Slave select polarity

* Not defined but normally LSB of previously transmitted character.
**Bit 4 - MSTR: Master/Slave Select**

This bit selects Master SPI mode when set (one), and Slave SPI mode when cleared (zero). It is configured as an input and is driven low when MSTR is set. MSTR will be cleared, and SPIF in SPSR will become set. The user will then have set MSTR to re-enable SPI master mode.

- **Bit 3 - CPOL: Clock Polarity**
  
  When this bit is set (one), SCK is high when idle. When CPOL is cleared (zero), SCK is low when idle. Refer to Figure 39 and Figure 40 for additional information.

- **Bit 2 - CPHA: Clock Phase**
  
  Refer to Figure 39 or Figure 40 for the functionality of this bit.

- **Bits 1,0 - SPR1, SPR0: SPI Clock Rate Select 1 and 0**
  
  These two bits control the SCK rate of the device configured as a master. SPR1 and SPR0 have no effect on the slave.

<table>
<thead>
<tr>
<th>SPRI</th>
<th>SPR0</th>
<th>SCK Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>fcl / 4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>fcl / 16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>fcl / 64</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>fcl / 128</td>
</tr>
</tbody>
</table>

Table 24. Relationship Between SCK and the Oscillator Frequency

Note: Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled.

**SPI Status Register - SPSR**

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPIF</td>
<td>WCOL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Read/Write: R/R/R/R/R/R/R/Initial value: 0/0/0/0/0/0/0/0

- **Bit 7 - SPIF: SPI Interrupt Flag**
  
  When a serial transfer is complete, the SPIF bit is set (one) and an interrupt is generated if SPIE in SPCR is set (one) or if a global interrupts are enabled. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI status register with SPIF set (one), then accessing the SPSR. The clearing of the SPIF bit is not dependent on the state of the SPI Slave Select (MSTR) bit.

- **Bit 6 - WCOL: Write Collision flag**
  
  The WCOL bit is set if the SPI data register (SPDR) is written during a data transfer. The WCOL bit is cleared (zero) by first reading the SPI Status Register with WCOL set (one), then accessing the SPI Data Register.

- **Bit 5, 0 - Res: Reserved bits**
  
  These bits are reserved bits in the ATmega603/103 and will always read as zero.

**SPI Data Register - SPDR**

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB</td>
<td>LSB</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Read/Write: R/W/R/W/R/W/R/W/Initial value: x/x/x/x/x/x/x/Undefined

The SPI Data Register is a read/write register used for data transfer between the register file and the SPI Shift register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.
Figure 41. UART Transmitter

On the Baud Rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin, followed by the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has been written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to set when the stop bit is shifted out, the UDRE flag will remain set. In this case, after the stop bit has been present on TXD for one bit length, the TX Complete Flag, TXC, in USR is set.

The TXEN bit in UCR enables the UART transmitter when set (one). When this bit is cleared (zero), the PE1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to PE1, which is forced to be an output pin regardless of the setting of the DDE1 bit in DDRE.

Figure 42. UART Receiver

The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical zero will be interpreted as the falling edge of a start bit, and the start bit detection sequence initiated. Let sample 1 denote the first zero sample. Following the 1 to 0 transition, the receiver samples the RXD pin samples 8, 9, and 10. If two or more of these three samples are found to be logical ones, the start bit is rejected as a noise spike and the receiver starts looking for the next 1 to 0 transition.

If, however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are sampled at samples 8, 9, and 10. The logical value found in at least two of these three samples is taken as the bit value. Bits are shifted into the transmitter shift register as they are sampled. Sampling of an incoming character is shown in Fig. 43.

Figure 43. Sampling Received Data
When the stop bit enters the receiver, the majority of the three samples must be one to accept the stop bit. If two or more samples are logic zeros, the Framing Error (FE) flag in the UART Status Register (USR) is set when the received data is transferred to UDR. Before reading the UDR register, the user should always check the FE bit to detect Framing Errors. It is cleared when UDR is read.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received data. When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register is accessed. If 9-bit word data is selected (the CHR9 bit in the UART Control Register, UCR, is set), the RXB8 bit in UCR loaded with bit 9 of the Transmit Shift register when data is transferred to UDR.

If, after receiving a character, the UDR register has not been accessed since the last receive, the OverRun (OR) flag in USR is set. This means that the new data transferred from the shift register could not be transferred to UDR and is lost. The OR bit is buffered, and is available when the valid data byte in UDR has been read. The user should always check the OR flag after reading from the UDR register in order to detect any overruns if the baud rate is high or CPU load is high.

When the RXEN bit in the UCR register is cleared (zero), the receiver is disabled. This means that the PE0 pin can be used as a general I/O pin. When RXEN is set, the UART Receiver will be connected to PE0, which is forced to be an input regardless of the setting of the DDE0 bit in DDRE. When PE0 is forced to input by the UART, the PORTE0 bit can still be used to control the pull-up resistor on the pin.

When the CH9 bit in the UCR register is set, transmitted and received characters are 9-bit long plus start and stop bit. The 9th data bit to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR register. The 9th byte data to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR register. The 9th byte data to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR register.

UART Control Register - UCR

Bit 7 - RXCIE: RX Complete Interrupt Enable
When this bit is set (one), a setting of the RXC bit in USR will cause the Receive Complete interrupt routine to be executed provided that global interrupts are enabled.

Bit 6 - TXCIE: TX Complete Interrupt Enable
When this bit is set (one), a setting of the TXC bit in USR will cause the Transmit Complete interrupt routine to be executed provided that global interrupts are enabled.

Bit 5 - UDRIE: UART Data Register Empty Interrupt Enable
When this bit is set (one), a setting of the UDRE bit in USR will cause the UART Data Register Empty interrupt routine to be executed provided that global interrupts are enabled.

Bit 4 - RXEN: Receiver Enable

When the RXEN bit in UCR is set, the UART Receive Complete Interrupt routine is executed when the RXC bit in USR is set. This enable the UART receiver when set (one). When the receiver is disabled, the RXC, OR and FE status flags cannot be set. If these flags are set, turning off RXEN does not cause them to be cleared.

Bit 3 - TXEN: Transmitter Enable

When the TXEN bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by writing a logical one to the bit.

Bit 5 - UDRE: UART Data Register Empty

When this bit is set (one) when a character written to UDR is transferred to the Transmit Shift register. Setting of this bit indicates that the transmitter is ready to receive a new character for transmission.

Bit 4 - FE: Framing Error

This bit is set if a Framing Error condition is detected, i.e. when the stop bit of an incoming character is zero. The FE bit is cleared when the stop bit of received data is zero.

Bit 3 - OR: OverRun

This bit is set if an OverRun condition is detected, i.e. when a character already present in the UDR register is not read before the next character is transferred from the Receiver Shift register. The OR bit is buffered, which means that it is cleared when the valid data byte in UDR is read.

Bit 2 - RXC: UART Receive Complete

This bit enables the UART receiver when set (one). When the receiver is disabled, the RXC, OR and FE status flags cannot be set. If these flags are set, turning off RXEN does not cause them to be cleared.

Bit 1 - TXC: UART Transmit Complete

This bit enables the UART transmitter when set (one). When disabling the transmitter while transmitting a character, the transmitter is not disabled before the character in the shift register plus any following character in UDR has been completely transmitted.

Bit 0 - CHR9: 9 Bit Characters

When this bit is set (one), transmitted and received characters are 9-bit long plus start and stop bits. The 9th bit data can be used as an extra stop bit or a parity bit.

When CH9 is set (one), RXB8 is the 9th data bit of the received character.
Bit 7 - ACD: Analog Comparator Disable

When this bit is set (one), the power to the analog comparator is switched off. This bit can be set at any time to turn off the analog comparator. This will reduce power consumption in active and idle mode. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

Bit 6 - Res: Reserved bit

This bit is a reserved bit in the ATmega603/103 and will always read as zero.

Table 25. UBRR Settings at Various CPU Frequencies

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBRR settings in Table 25. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. Using baud rates that have more than 1% error is not recommended. High error ratings give less noise resistance.

Analog Comparator

The analog comparator compares the input values on the positive input PE2 (AC+) and negative input PE3 (AC-). When the voltage on the positive input PE2 (AC+) is higher than the voltage on the negative input PE3 (AC-), the Analog Comparator Output, ACO is set (one). The output of the comparator can be used to generate an interrupt via the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select interrupt triggering on comparator output rise, fall, or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 44.

Figure 44. Analog Comparator Block Diagram
Bit 4 - ACI: Analog Comparator Interrupt Flag

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in the Status Register is set (one). ACI is cleared (zero) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated when cleared (zero), the interrupt is disabled.

Bit 3 - ACIE: Analog Comparator Interrupt Enable

When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated. When cleared (zero), the interrupt is disabled.

Bit 2 - ACIC: Analog Comparator Input Capture enable

When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the analog comparator. The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator trigger the Input Capture function. When cleared (zero), no connection between the analog comparator and the Input Capture function is given. The comparator trigger is connected to the Timer Interrupt Mask Register (TIMSK).

Bits 1,0 - ACIS1, ACIS0: Analog Comparator Interrupt Mode Select

These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in Table 26.

Table 26. ACIS1/ACIS0 Settings

<table>
<thead>
<tr>
<th>ACIS1</th>
<th>ACIS0</th>
<th>Interrupt Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Comparator Interrupt on Output Toggle</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Comparator Interrupt on Rising Output Edge</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Comparator Interrupt on Falling Output Edge</td>
</tr>
</tbody>
</table>

When changing the ACIS1/ACIS0 bits, the Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed. Caution: Using the SBI or CBI instruction on other bits than ACI in this register, will write a one back into ACI if it is read set, thus clearing the flag.

Analog to Digital Converter

Feature list:
- 10-bit Resolution
- ±12 LSB absolute accuracy
- 0.5 LSB Integral Non-linearity
- 70 - 290 µs conversion time
- Up to 14 ksPS
- 8 Multiplexed Input Channels
- Interrupt on ADC conversion complete
- Sleep Mode Noise Canceller

The ATmega603/103 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Mux terminal which allows any of Port F to be used as an input for the ADC. The ADC contains a Sample and Hold Amplifier which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the AI is shown in Figure 45.

The ADC has two separate analog supply voltage pins, AVcc and AGND. AGND must be connected to GND, and the voltage on AVcc must not differ more than ±0.3 V from Vcc. See the section “ADC Noise Cancelling Techniques” on page 66 on how to connect these pins.

An external reference voltage must be applied to the AREF pin. This voltage must be in the range AGND - AVCC.

Figure 45. Analog to Digital Converter Block Schematic
The ADC contains a prescaler, which divides the system clock to an acceptable ADC clock frequency. The ADC accepts input clock frequencies in the range 50 - 200 kHz. Applying a higher input frequency will result in a poorer accuracy, see "ADC DC Characteristics" on page 72.

The ADPS0 - ADPS2 bits in ADCSR are used to generate a proper ADC clock input frequency from any XTAL frequency above 100 kHz. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSR. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a conversion by setting the ADSC bit in ADCSR, the conversion starts at the following falling edge of the ADC clock cycle. The actual sample-and-hold takes place one ADC clock cycle after the start of the conversion. The results are written into the result registers ADCH and ADCL. The ADEN bit must be set high in this period, the ADC will start the new conversion immediately. For a summary conversion times, see Table 27.

**ADC Noise Canceler Function**

The ADC features a noise canceler that enables conversion during idle mode to reduce noise induced from the CPU core or other analog circuitry. To make use of this feature, the following procedure should be used:

1. Turn off the ADC by clearing ADEN.
2. Turn on the ADC and simultaneously start a conversion by setting ADEN and ADSC. This starts a dummy conversion that will be followed by a valid conversion.
3. Within 14 ADC clock cycles, enter idle mode.
4. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the MCU.

**ADC Multiplexer Select Register - ADMUX**

- **Bits 7..3 - Res: Reserved Bits**
  - These bits are reserved bits in the ATmega603/103 and always read as zero.
- **Bits 2..0 - MUX2..MUX0: Analog Channel Select Bits 2-0**
  - The value of these three bits selects which analog input 7-0 is connected to the ADC.

**Table 27. ADC Conversion Time**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Sample Cycle Number</th>
<th>Result Ready (cycle number)</th>
<th>Total Conversion Time (cycles)</th>
<th>Total Conversion Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Conversion</td>
<td>16</td>
<td>14</td>
<td>14</td>
<td>140 - 560</td>
</tr>
<tr>
<td>2nd Conversion</td>
<td>16</td>
<td>13</td>
<td>13</td>
<td>140 - 560</td>
</tr>
<tr>
<td>Single Conversion</td>
<td>16</td>
<td>12</td>
<td>12</td>
<td>140 - 560</td>
</tr>
<tr>
<td>Continuous Conversion</td>
<td>16</td>
<td>11</td>
<td>11</td>
<td>140 - 560</td>
</tr>
</tbody>
</table>

The value of these three bits selects which analog input 7-0 is connected to the ADC.
ADC Control and Status Register - ADCSR

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$06$</td>
<td>[ADEN]</td>
<td>[ADSC]</td>
<td>-</td>
<td>[ADIF]</td>
<td>[ADIE]</td>
<td>[ADPS2]</td>
<td>[ADPS1]</td>
<td>[ADPS0]</td>
</tr>
<tr>
<td>Read/Write</td>
<td>R/W</td>
<td>R/W</td>
<td>R</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>Initial value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Bit 7 - ADEN: ADC Enable
  Writing a logical '1' to this bit enables the ADC. By clearing this bit to zero, the ADC is turned off. Turning the ADC off when a conversion is in progress, will terminate this conversion.

- Bit 6 - ADSC: ADC Start Conversion
  A logical '1' must be written to this bit to start each conversion. The first time ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, a dummy conversion will precede the initialize conversion. This dummy conversion performs initialization of the ADC.
  ADCSC remains high during the conversion. ADCSC goes low after the conversion is complete, but before the result is written to the ADC Data Registers. This allows a new conversion to be initiated before the current conversion is complete. The result conversion will then start immediately after the current conversion completes. When a dummy conversion precedes a real conversion, ADCSC will stay high until the real conversion completes.
  Writing a zero to this bit has no effect.

- Bit 5 - Res: Reserved Bit
  This bit is reserved in the ATmega603/103.
  Warning: When writing ADCSR, a logical "0" must be written to this bit.

- Bit 4 - ADIF: ADC Interrupt Flag
  This bit is set (one) when an ADC conversion is complete and the result is written to the ADC Data Registers.

ADC Noise Canceling Techniques

Digital circuitry inside and outside the ATmega603/103 generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. The analog part of the ATmega603/103 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.
2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
3. The AVCC pin on the ATmega603/103 should have its own decoupling capacitor as shown in Figure 49.
4. Use the ADC noise canceler function to reduce induced noise from the CPU.
5. If some Port F pins are used as digital inputs, it is essential that these do not switch while a conversion is in progress.

Table 28. ADC Prescaler Selections

<table>
<thead>
<tr>
<th>ADPS2</th>
<th>ADPS1</th>
<th>ADPS0</th>
<th>Division Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Invalid</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>128</td>
</tr>
</tbody>
</table>

ADC Data Register - ADCL and ADCH

<table>
<thead>
<tr>
<th>Bit</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$05$</td>
<td>[ADCH]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Read/Write</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>Initial value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

When an ADC conversion is complete, the result is found in these two registers. It is essential that both registers are read and that ADCL is read before ADCH.

ADC Power Connections

![ADC Power Connections diagram](image-url)
I/O-Ports

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and C instructions. The same applies for changing drive value (if configured as output) or enabling/disabling of pull-up resistors configured as input.

Port A

Port A is an 8-bit bi-directional I/O port with internal pull-ups.

Three I/O memory address locations are allocated for Port A, one each for the Data Register - PORTA, $1B($3B), Direction Register - DDRA, $1A($3A) and the Port A Input Pins - PINA, $19($39). The Port A Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port A output buffers can sink 20mA and thus drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port A pins have alternate functions related to the optional external data SRAM. Port A can be configured to be a multiplexed low-order address/data bus during accesses to the byte.

When Port A is set to the alternate function by the SRE - External SRAM Enable bit in the MCUCR - MCU Control Register, the alternate settings override the data direction register.

Table 29. DDAn Effects on Port A Pins

<table>
<thead>
<tr>
<th>DDAn</th>
<th>PORTAn</th>
<th>I/O</th>
<th>Pull up</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Input</td>
<td>No</td>
<td>Tri-state (Hi-Z)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Input</td>
<td>Yes</td>
<td>PAN will source current if ext. pulled low.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Output</td>
<td>No</td>
<td>Push-Pull Zero Output</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Output</td>
<td>No</td>
<td>Push-Pull One Output</td>
</tr>
</tbody>
</table>

Note: n: 7, 6...0, pin number
Port B

Port B is an 8-bit bi-directional I/O port with internal pull-ups.

Three I/O memory address locations are allocated for Port B, one each for the Data Register - PORTB, $18($38), Direction Register - DDRB, $17($37) and the Port B Input Pins - PINB, $16($36). The Port B Input Pins address is readonly, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port B output buffers can sink 20mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port B pins with alternate functions are shown in the following table:

### Table 30. Port B Pins Alternate Functions

<table>
<thead>
<tr>
<th>Port Pin</th>
<th>Alternate Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB0</td>
<td>SS (SPI Slave Select input)</td>
</tr>
<tr>
<td>PB1</td>
<td>SCK (SPI Bus Serial Clock)</td>
</tr>
<tr>
<td>PB2</td>
<td>MOSI (SPI Bus Master Output/Slave Input)</td>
</tr>
<tr>
<td>PB3</td>
<td>MISO (SPI Bus Master Input/Slave Output)</td>
</tr>
<tr>
<td>PB4</td>
<td>OC0/PWM0 (Output Compare and PWM Output for Timer/Counter0)</td>
</tr>
<tr>
<td>PB5</td>
<td>OC1A/PWM1A (Output Compare and PWM Output A for Timer/Counter1)</td>
</tr>
<tr>
<td>PB6</td>
<td>OC1B/PWM1B (Output Compare and PWM Output B for Timer/Counter1)</td>
</tr>
<tr>
<td>PB7</td>
<td>OC2/PWM2 (Output Compare and PWM Output for Timer/Counter2)</td>
</tr>
</tbody>
</table>

When the pins are used for the alternate function the DDRB and PORTB register have to be set according to the alternate function description.

### Port B Data Register - PORTB

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S18</td>
<td>$18($38)</td>
<td>PORTB7</td>
<td>PORTB6</td>
<td>PORTB5</td>
<td>PORTB4</td>
<td>PORTB3</td>
<td>PORTB2</td>
<td>PORTB1</td>
</tr>
<tr>
<td>Read/Write</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>Initial value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Port B Data Direction Register - DDRB

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S17</td>
<td>$17($37)</td>
<td>DDB7</td>
<td>DDB6</td>
<td>DDB5</td>
<td>DDB4</td>
<td>DDB3</td>
<td>DDB2</td>
<td>DDB1</td>
</tr>
<tr>
<td>Read/Write</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>Initial value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Port B Input Pins Address - PINB

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S16</td>
<td>$16($36)</td>
<td>PINB7</td>
<td>PINB6</td>
<td>PINB5</td>
<td>PINB4</td>
<td>PINB3</td>
<td>PINB2</td>
<td>PINB1</td>
</tr>
<tr>
<td>Read/Write</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
</tbody>
</table>

The Port B Input Pins address - PINB - is not a register, and this address enables access to the physical value on each Port B pin. When reading PORTB, the Port B Data Latch is read, and when reading PINB, the logical values present on the pins are read.

### Port B as General Digital I/O

All 8 pins in port B have equal functionality when used as digital I/O pins.

PBn, General I/O pin: The DDBn bit in the DDRB register selects the direction of this pin. If DDBn is set (one), PBn is configured as an input pin. If DDBn is cleared (zero), PBn is configured as an output pin. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.
Alternate Functions of Port B

The alternate pin configuration is as follows:

- **OC2/PWM2, Bit 7**
  OC2/PWM2, Output Compare output for Timer/Counter2 or PWM output when Timer/Counter2 is in PWM Mode. The pin has to be configured as an output to serve this function.

- **OC1B/PWM1B, Bit 6**
  OC1B/PWM1B, Output Compare output B for Timer/Counter1 or PWM output B when Timer/Counter1 is in PWM Mode. The pin has to be configured as an output to serve this function.

- **OC1A/PWM1A, Bit 5**
  OC1A/PWM1A, Output Compare output A for Timer/Counter1 or PWM output A when Timer/Counter1 is in PWM Mode. The pin has to be configured as an output to serve this function.

- **OC0/PWM0, Bit 4**
  OC0/PWM0, Output Compare output for Timer/Counter0 or PWM output when Timer/Counter0 is in PWM Mode. The pin has to be configured as an output to serve this function.

- **MISO - Port B, Bit 3**
  MISO: Master data input, slave data output pin for SPI channel. When the SPI is enabled as a master, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a slave, the data direction of this pin is controlled by DDB3. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB3 bit. See the description of the SPI port for further details.

- **MOSI - Port B, Bit 2**
  MOSI: SPI Master data output, slave data input for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB2. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB2. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB2 bit. See the description of the SPI port for further details.

- **SCK - Port B, Bit 1**
  SCK: Master clock output, slave clock input pin for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB1. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB1. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB1 bit. See the description of the SPI port for further details.

- **SS - Port B, Bit 0**
  SS: Slave port select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB0. As a slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB0. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB0 bit. See the description of the SPI port for further details.

### Table 31. DDBn Effects on Port B Pins

<table>
<thead>
<tr>
<th>DDBn</th>
<th>PORTBn</th>
<th>I/O</th>
<th>Pull up</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Input</td>
<td>No</td>
<td>Tri-state (H-Z)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Input</td>
<td>Yes</td>
<td>PBn will source current if ext, pulled low</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Output</td>
<td>No</td>
<td>Push-Pull Zero Output</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Output</td>
<td>No</td>
<td>Push-Pull One Output</td>
</tr>
</tbody>
</table>

Note: n: 7...0, pin number

**Port B Schematics**

Note that all port pins are synchronized. The synchronization latches are however, not shown in the figures.

**Figure 54. Port B Schematic Diagram (Pin PB0)**

**Figure 55. Port B Schematic Diagram (Pin PB1)**
Figure 56. Port B Schematic Diagram (Pin PB2)

Figure 57. Port B Schematic Diagram (Pin PB3)

Figure 58. Port B Schematic Diagram (Pin PB4)

Figure 59. Port B Schematic Diagram (Pins PB5 and PB6)
Port C
PORT C is an 8-bit Output port.
The Port C pins have alternate functions related to the optional external data SRAM. When using the device with external SRAM, Port C outputs the high-order address byte during accesses to external data memory. When a reset condition becomes active, the port pins are not tristated, but the pins will assume their initial value after two stable clock cycles.

The Port C Data Register - PORTC

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$15$ ($35)</td>
<td>PORTC7</td>
<td>PORTC6</td>
<td>PORTC5</td>
<td>PORTC4</td>
<td>PORTC3</td>
<td>PORTC2</td>
<td>PORTC1</td>
<td>PORTC0</td>
</tr>
</tbody>
</table>

Read/Write

Initial value 0 0 0 0 0 0 0 0

Port D
Port D is an 8 bit bi-directional I/O port with internal pull-up resistors.
Three I/O memory address locations are allocated for the Port D, one each for the Data Register - PORTD, $12$ ($32$), Direction Register - DDRD, $11$ ($31$) and the Port D Input Pins - PIND, $10$ ($30$). The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.
The Port D output buffers can sink $20$ mA. As inputs, Port D pins that are externally pulled low will source current if the pullup resistors are activated.

Some Port D pins have alternate functions as shown in the following table:

<table>
<thead>
<tr>
<th>Port Pin</th>
<th>Alternate Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD0</td>
<td>INT0 (External Interrupt0 Input)</td>
</tr>
<tr>
<td>PD1</td>
<td>INT1 (External Interrupt1 Input)</td>
</tr>
<tr>
<td>PD2</td>
<td>INT2 (External Interrupt2 Input)</td>
</tr>
<tr>
<td>PD3</td>
<td>INT3 (External Interrupt3 Input)</td>
</tr>
<tr>
<td>PD4</td>
<td>IC1 (Timer/Counter1 Input Capture Trigger)</td>
</tr>
<tr>
<td>PD6</td>
<td>T1 (Timer/Counter1 Clock Input)</td>
</tr>
<tr>
<td>PD7</td>
<td>T2 (Timer/Counter2 Clock Input)</td>
</tr>
</tbody>
</table>

When the pins are used for the alternate function the DDRD and PORTD register has to be set according to the alternate function description.

Port D Data Register - PORTD

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$12$</td>
<td>PORTD7</td>
<td>PORTD6</td>
<td>PORTD5</td>
<td>PORTD4</td>
<td>PORTD3</td>
<td>PORTD2</td>
<td>PORTD1</td>
<td>PORTD0</td>
</tr>
</tbody>
</table>

Read/Write

Initial value 0 0 0 0 0 0 0 0
Port D Input Pins Address - PIND

The Port D Input Pins address - PIND - is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the Port D Data Latch is read, and when reading PIND, the logical values present on the pins are read.

Port D as general digital I/O

PDn, General I/O pin: The DDDn bit in the DDRD register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when configured as an output pin, the value on pin is written to the output latch. When configured as an input pin, the value on the pin is read and latched into the input register. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Note: n: 7,6...0, pin number

Alternate Functions of Port D

INT0..INT3 - Port D, Bits 0..3

External interrupt sources 0 - 3. The PD0 - PD3 pins can serve as external active low interrupt sources to the MCU. The internal pull up MOS resistors can be activated as described above. See the interrupt description for further details, and how to enable the sources.

IC1 - Port D, Bit 4

IC1 - Input Capture pin for Timer/Counter1. When a positive or negative (selectable) edge is applied to this pin, the contents of Timer/Counter1 is transferred to the Timer/Counter1 Input Capture Register. The pin has to be configured as an input pin. The internal pull up MOS resistor can be activated as described above. See the timer description on how to operate this function.

T1 - Port D, Bit 6

T1, Timer/Counter1 counter source. See the timer description for further details.

T2 - Port D, Bit 7

T2, Timer/Counter2 counter source. See the timer description for further details.

Table 33: DDDn Bits on Port D Pins

<table>
<thead>
<tr>
<th>DDDn</th>
<th>PORTDn</th>
<th>I/O</th>
<th>Pullup</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>No</td>
<td>No</td>
<td>To state (h-Z)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Yes</td>
<td>No</td>
<td>Pull out current is pulled low</td>
</tr>
</tbody>
</table>

Figure 62: Port D Schematic Diagram (Port D0, D1, D2 and D3)

Figure 63: Port D Schematic Diagram (Port D4)
Port E

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors.

Three I/O memory address locations are allocated for the Port E, one each for the Data Register - PORTE, $03(023), Dir Direction Register - DDRE, $02(022) and the Port E Input Pins - PINE, $01(021). The Port E Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

The Port E output buffers can sink 20 mA. As inputs, Port E pins that are externally pulled low will source current if the pull up resistors are activated.

All Port E pins have alternate functions as shown in the following table:

<table>
<thead>
<tr>
<th>Port E Pin Alternate Function</th>
<th>PE0</th>
<th>PE1</th>
<th>PE2</th>
<th>PE3</th>
<th>PE4</th>
<th>PE5</th>
<th>PE6</th>
<th>PE7</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD0/RXD (Programming Data Input or UART Receive Pin)</td>
<td>PDO/TXD (Programming Data Output or UART Transmit Pin)</td>
<td>AC+ (Analog Comparator Positive Input)</td>
<td>AC- (Analog Comparator Negative Input)</td>
<td>INT4 (External Interrupt4 Input)</td>
<td>INT5 (External Interrupt5 Input)</td>
<td>INT6 (External Interrupt6 Input)</td>
<td>INT7 (External Interrupt7 Input)</td>
<td></td>
</tr>
</tbody>
</table>

When the pins are used for the alternate function the DDRE and PORTE register has to be set according to the alternate function description.

Port E Data Register - PORTE

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S03</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>Read/Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Port E Data Direction Register - DDRE

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S02</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>Read/Write</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Port E Input Pins Address - PINE

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>S01</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>Read/Write</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
</tr>
</tbody>
</table>

The Port E Input Pins address - PINE - is not a register, and this address enables access to the physical value on each Port E pin. When reading PORTE, the Port E Data Latch is read, and when reading PINE, the logical values present on the pins are read.
Port E as general digital I/O

Port E is a general digital I/O port. The DDEN bit in the DDRE register selects the direction of this pin. If DDEN is set (one), PEEn is configured as an output pin. If DDEN is cleared (zero), PEEn is configured as an input pin. When configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, PEEn must be cleared (zero) or the pin must be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

<table>
<thead>
<tr>
<th>DDEn</th>
<th>PORTEn</th>
<th>I/O</th>
<th>Pull up</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Input</td>
<td>No</td>
<td>Tri-state (Hi-Z)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Input</td>
<td>Yes</td>
<td>PDn will source current if ext. pulled low.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Output</td>
<td>No</td>
<td>Push-Pull Zero Output</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Output</td>
<td>No</td>
<td>Push-Pull One Output</td>
</tr>
</tbody>
</table>

Table 35. DDEn Bits on Port E Pins

Note: n: 7..1..0, pin number

Alternate Functions Of Port E

PDI/RXD - Port E, Bit 0
PDI, Serial Programming Data Input. During Serial Program Downloading, this pin is used as data input line for ATmega603/103.
RXD, UART Receive Pin. Receive Data (Data input pin for the UART). When the UART receiver is enabled this pin is configured as an input regardless of the value of DDRD0. When the UART forces this pin to be an input, a logical one on PORTD0 will turn on the internal pull-up.

PDO/TXD - Port E, Bit 1
PDO, Serial Programming Data Output. During Serial Program Downloading, this pin is used as data output line for ATmega603/103.
TXD, UART Transmit Pin.

AC+ - Port E, Bit 2
AC+ - Analog Comparator Positive Input. This pin is directly connected to the positive input of the analog comparator.

AC- - Port E, Bit 3
AC- - Analog Comparator Negative Input. This pin is directly connected to the negative input of the analog comparator.

INT4 ... INT7 - Port E, Bit 4-7
INT4 ... INT7 - External Interrupt sources 4 - 7. The PE4 - PE7 pins can serve as external interrupt sources to the MCU. Interrupts can be triggered by low level or positive or negative edge on these pins. The internal pull-up MOS resisters can be activated as described above. See the interrupt description for further details, and how to enable the sources.

Port E Schematics

Note that all port pins are synchronized. The synchronization latches are however, not shown in the figures.
Figure 68. Port E Schematic Diagram (Pin PE2)

Figure 69. Port E Schematic Diagram (Pin PE3)

Port F
Port F is an 8-bit input port.
One I/O memory location is allocated for Port F, the Port F Input Pins - PINF, $00 ($20).
All Port F pins are connected to the analog multiplexer which further is connected to the A/D converter. The digital inq function of Port F can be used together with the A/D converter, allowing the user to use some pins of Port F and digit inputs and other as analog inputs, at the same time.

Port F Input Pins Address - PINF

<table>
<thead>
<tr>
<th>Bit</th>
<th>$00 ($20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>PINF7 R</td>
</tr>
<tr>
<td>6</td>
<td>PINF6 R</td>
</tr>
<tr>
<td>5</td>
<td>PINF5 R</td>
</tr>
<tr>
<td>4</td>
<td>PINF4 R</td>
</tr>
<tr>
<td>3</td>
<td>PINF3 R</td>
</tr>
<tr>
<td>2</td>
<td>PINF2 R</td>
</tr>
<tr>
<td>1</td>
<td>PINF1 R</td>
</tr>
<tr>
<td>0</td>
<td>PINF0 R</td>
</tr>
</tbody>
</table>


The Port F Input Pins address - PINF - is not a register, and this address enables access to the physical value on each Port F pin.
Memory Programming

Program and Data Memory Lock Bits

The ATmega603/103 MCU provides two Lock bits which can be left unprogrammed ('1') or can be programmed ('0') to obtain the additional features listed in Table 36. The Lock bits can only be enabled for parallel operation. Program the Fuse bits before programming the Lock bits.

Table 36. Lock Bit Protection Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>LB1</th>
<th>LB2</th>
<th>Protection Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>No memory lock features enabled.</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>Further programming of the Flash and EEPROM is disabled.</td>
</tr>
</tbody>
</table>

Note: In parallel mode, programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The ATmega603/103 has four Fuse bits, SPIEN, SUT1, 0, and EESAVE:

- When the SPIEN Fuse is programmed ('1'), Serial Program and Data Downloading is enabled. Default value is unprogrammed ('0'). The SPIEN Fuse is not accessible in serial mode.
- When EESAVE is programmed ('1'), the EEPROM memory is preserved through the Chip Erase cycle. Default value is unprogrammed ('1'). The EESAVE Fuse bit can not be programmed if any of the Lock bits are programmed.
- SUT1,0 Fuses: Determine the MCU start-up time. See Table 6 on page 26 for further details. Default value is unprogrammed ('11'), which gives a nominal start up time of 16 ms.

The status of the Fuse bits is not affected by Chip Erase.

Register Summary

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit7</th>
<th>Bit6</th>
<th>Bit5</th>
<th>Bit4</th>
<th>Bit3</th>
<th>Bit2</th>
<th>Bit1</th>
<th>Bit0</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3F</td>
<td>SREG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>$3E</td>
<td>SPH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>$3D</td>
<td>SPL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>$3C</td>
<td>XDIV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>$3B</td>
<td>RAMPZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>$3A</td>
<td>EICR</td>
<td>ISC7</td>
<td>ISC6</td>
<td>ISC5</td>
<td>ISC4</td>
<td>ISC3</td>
<td>ISC2</td>
<td>ISC1</td>
<td>ISC0</td>
<td>31</td>
</tr>
<tr>
<td>$39</td>
<td>EIMSK</td>
<td>INT7</td>
<td>INT6</td>
<td>INT5</td>
<td>INT4</td>
<td>INT3</td>
<td>INT2</td>
<td>INT1</td>
<td>INT0</td>
<td>30</td>
</tr>
<tr>
<td>$38</td>
<td>EIFR</td>
<td>INTF7</td>
<td>INTF6</td>
<td>INTF5</td>
<td>INTF4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>$37</td>
<td>TIMSK</td>
<td>OCIE2</td>
<td>TOIE2</td>
<td>TICIE1</td>
<td>OCIE1A</td>
<td>OCIE1B</td>
<td>TOIE1</td>
<td>OCIE0</td>
<td>TOIE0</td>
<td>31</td>
</tr>
<tr>
<td>$36</td>
<td>TIFR</td>
<td>OCF2</td>
<td>TOV2</td>
<td>ICF1</td>
<td>OCF1A</td>
<td>OCF1B</td>
<td>TOV1</td>
<td>OCF0</td>
<td>TOV0</td>
<td>32</td>
</tr>
<tr>
<td>$35</td>
<td>MCUCR</td>
<td>SRE</td>
<td>SRW</td>
<td>SE</td>
<td>SM1</td>
<td>SM0</td>
<td></td>
<td></td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>$34</td>
<td>MCUSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>$33</td>
<td>TCCR0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>39</td>
</tr>
<tr>
<td>$32</td>
<td>TCNT0</td>
<td>Timer/Counter0 (8 Bit)</td>
<td>page 39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$31</td>
<td>OCR0</td>
<td>Timer/Counter0 Output Compare Register</td>
<td>page 40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$30</td>
<td>ASSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>$2E</td>
<td>TCCR1B</td>
<td>ICNC1</td>
<td>ICES1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>47</td>
</tr>
<tr>
<td>$2D</td>
<td>TCNT1H</td>
<td>Timer/Counter1 - Counter Register High Byte</td>
<td>page 47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2C</td>
<td>TCNT1L</td>
<td>Timer/Counter1 - Counter Register Low Byte</td>
<td>page 47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2B</td>
<td>OCR1AH</td>
<td>Timer/Counter1 - Output Compare Register A High Byte</td>
<td>page 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2A</td>
<td>OCR1AL</td>
<td>Timer/Counter1 - Output Compare Register A Low Byte</td>
<td>page 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$29</td>
<td>OCR1BH</td>
<td>Timer/Counter1 - Output Compare Register B High Byte</td>
<td>page 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$28</td>
<td>OCR1BL</td>
<td>Timer/Counter1 - Output Compare Register B Low Byte</td>
<td>page 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$27</td>
<td>ICR1H</td>
<td>Timer/Counter1 - Input Capture Register High Byte</td>
<td>page 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$26</td>
<td>ICR1L</td>
<td>Timer/Counter1 - Input Capture Register Low Byte</td>
<td>page 48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$25</td>
<td>TCCR2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>$24</td>
<td>OCR2</td>
<td>Timer/Counter2 Output Compare Register</td>
<td>page 40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$23</td>
<td>DDRB</td>
<td>DDB7</td>
<td>DDB6</td>
<td>DDB5</td>
<td>DDB4</td>
<td>DDB3</td>
<td>DDB2</td>
<td>DDB1</td>
<td>DDB0</td>
<td>77</td>
</tr>
<tr>
<td>$22</td>
<td>PORTB</td>
<td>PORTB7</td>
<td>PORTB6</td>
<td>PORTB5</td>
<td>PORTB4</td>
<td>PORTB3</td>
<td>PORTB2</td>
<td>PORTB1</td>
<td>PORTB0</td>
<td>77</td>
</tr>
<tr>
<td>$21</td>
<td>USR</td>
<td>RXC</td>
<td>TXC</td>
<td>UDRE</td>
<td>FE</td>
<td>OR</td>
<td>GPC®</td>
<td>AM4</td>
<td>Rel. 5.00</td>
<td></td>
</tr>
<tr>
<td>$20</td>
<td>UBRR</td>
<td>UART Baud Rate Register</td>
<td>page 65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1F</td>
<td>SPDR</td>
<td>SPI Data Register</td>
<td>page 58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1E</td>
<td>USR</td>
<td>RXCIE</td>
<td>TXCIE</td>
<td>UDRIE</td>
<td>RXEN</td>
<td>TXEN</td>
<td>CHR9</td>
<td>RXB8</td>
<td>TXB8</td>
<td>63</td>
</tr>
<tr>
<td>$1D</td>
<td>EEDR</td>
<td>EEPROM Data Register</td>
<td>page 53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1C</td>
<td>EECR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>53</td>
</tr>
<tr>
<td>$1B</td>
<td>PORTA</td>
<td>PORTA7</td>
<td>PORTA6</td>
<td>PORTA5</td>
<td>PORTA4</td>
<td>PORTA3</td>
<td>PORTA2</td>
<td>PORTA1</td>
<td>PORTA0</td>
<td>75</td>
</tr>
<tr>
<td>$1A</td>
<td>DDRA</td>
<td>DDA7</td>
<td>DDA6</td>
<td>DDA5</td>
<td>DDA4</td>
<td>DDA3</td>
<td>DDA2</td>
<td>DDA1</td>
<td>DDA0</td>
<td>75</td>
</tr>
<tr>
<td>$19</td>
<td>PINA</td>
<td>PINA7</td>
<td>PINA6</td>
<td>PINA5</td>
<td>PINA4</td>
<td>PINA3</td>
<td>PINA2</td>
<td>PINA1</td>
<td>PINA0</td>
<td>75</td>
</tr>
<tr>
<td>$18</td>
<td>PORTC</td>
<td>PORTC7</td>
<td>PORTC6</td>
<td>PORTC5</td>
<td>PORTC4</td>
<td>PORTC3</td>
<td>PORTC2</td>
<td>PORTC1</td>
<td>PORTC0</td>
<td>75</td>
</tr>
<tr>
<td>$17</td>
<td>DDRC</td>
<td>DDC7</td>
<td>DDC6</td>
<td>DDC5</td>
<td>DDC4</td>
<td>DDC3</td>
<td>DDC2</td>
<td>DDC1</td>
<td>DDC0</td>
<td>77</td>
</tr>
<tr>
<td>$16</td>
<td>PINC</td>
<td>PINC7</td>
<td>PINC6</td>
<td>PINC5</td>
<td>PINC4</td>
<td>PINC3</td>
<td>PINC2</td>
<td>PINC1</td>
<td>PINC0</td>
<td>77</td>
</tr>
<tr>
<td>$15</td>
<td>PORTD</td>
<td>PORTD7</td>
<td>PORTD6</td>
<td>PORTD5</td>
<td>PORTD4</td>
<td>PORTD3</td>
<td>PORTD2</td>
<td>PORTD1</td>
<td>PORTD0</td>
<td>83</td>
</tr>
<tr>
<td>$14</td>
<td>DDRE</td>
<td>DDE7</td>
<td>DDE6</td>
<td>DDE5</td>
<td>DDE4</td>
<td>DDE3</td>
<td>DDE2</td>
<td>DDE1</td>
<td>DDE0</td>
<td>84</td>
</tr>
<tr>
<td>$13</td>
<td>PIND</td>
<td>PIND7</td>
<td>PIND6</td>
<td>PIND5</td>
<td>PIND4</td>
<td>PIND3</td>
<td>PIND2</td>
<td>PIND1</td>
<td>PIND0</td>
<td>84</td>
</tr>
<tr>
<td>$12</td>
<td>PORTE</td>
<td>PORTE7</td>
<td>PORTE6</td>
<td>PORTE5</td>
<td>PORTE4</td>
<td>PORTE3</td>
<td>PORTE2</td>
<td>PORTE1</td>
<td>PORTE0</td>
<td>87</td>
</tr>
<tr>
<td>$11</td>
<td>DDRE</td>
<td>DDE7</td>
<td>DDE6</td>
<td>DDE5</td>
<td>DDE4</td>
<td>DDE3</td>
<td>DDE2</td>
<td>DDE1</td>
<td>DDE0</td>
<td>84</td>
</tr>
<tr>
<td>$10</td>
<td>PINF</td>
<td>PINF7</td>
<td>PINF6</td>
<td>PINF5</td>
<td>PINF4</td>
<td>PINF3</td>
<td>PINF2</td>
<td>PINF1</td>
<td>PINF0</td>
<td>91</td>
</tr>
<tr>
<td>$0F</td>
<td>SPDR</td>
<td>SPI Data Register</td>
<td>page 58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0E</td>
<td>USR</td>
<td>RXCIE</td>
<td>TXCIE</td>
<td>UDRIE</td>
<td>RXEN</td>
<td>TXEN</td>
<td>CHR9</td>
<td>RXB8</td>
<td>TXB8</td>
<td>63</td>
</tr>
<tr>
<td>$0D</td>
<td>EEDR</td>
<td>EEPROM Data Register</td>
<td>page 53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0C</td>
<td>EECR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>53</td>
</tr>
<tr>
<td>$0B</td>
<td>PORTA</td>
<td>PORTA7</td>
<td>PORTA6</td>
<td>PORTA5</td>
<td>PORTA4</td>
<td>PORTA3</td>
<td>PORTA2</td>
<td>PORTA1</td>
<td>PORTA0</td>
<td>75</td>
</tr>
<tr>
<td>$0A</td>
<td>DDRA</td>
<td>DDA7</td>
<td>DDA6</td>
<td>DDA5</td>
<td>DDA4</td>
<td>DDA3</td>
<td>DDA2</td>
<td>DDA1</td>
<td>DDA0</td>
<td>75</td>
</tr>
<tr>
<td>$09</td>
<td>PINA</td>
<td>PINA7</td>
<td>PINA6</td>
<td>PINA5</td>
<td>PINA4</td>
<td>PINA3</td>
<td>PINA2</td>
<td>PINA1</td>
<td>PINA0</td>
<td>75</td>
</tr>
<tr>
<td>$08</td>
<td>PORTC</td>
<td>PORTC7</td>
<td>PORTC6</td>
<td>PORTC5</td>
<td>PORTC4</td>
<td>PORTC3</td>
<td>PORTC2</td>
<td>PORTC1</td>
<td>PORTC0</td>
<td>75</td>
</tr>
<tr>
<td>$07</td>
<td>DDRC</td>
<td>DDC7</td>
<td>DDC6</td>
<td>DDC5</td>
<td>DDC4</td>
<td>DDC3</td>
<td>DDC2</td>
<td>DDC1</td>
<td>DDC0</td>
<td>77</td>
</tr>
<tr>
<td>$06</td>
<td>PINC</td>
<td>PINC7</td>
<td>PINC6</td>
<td>PINC5</td>
<td>PINC4</td>
<td>PINC3</td>
<td>PINC2</td>
<td>PINC1</td>
<td>PINC0</td>
<td>77</td>
</tr>
<tr>
<td>$05</td>
<td>PORTD</td>
<td>PORTD7</td>
<td>PORTD6</td>
<td>PORTD5</td>
<td>PORTD4</td>
<td>PORTD3</td>
<td>PORTD2</td>
<td>PORTD1</td>
<td>PORTD0</td>
<td>83</td>
</tr>
<tr>
<td>$04</td>
<td>DDRE</td>
<td>DDE7</td>
<td>DDE6</td>
<td>DDE5</td>
<td>DDE4</td>
<td>DDE3</td>
<td>DDE2</td>
<td>DDE1</td>
<td>DDE0</td>
<td>84</td>
</tr>
<tr>
<td>$03</td>
<td>PIND</td>
<td>PIND7</td>
<td>PIND6</td>
<td>PIND5</td>
<td>PIND4</td>
<td>PIND3</td>
<td>PIND2</td>
<td>PIND1</td>
<td>PIND0</td>
<td>84</td>
</tr>
<tr>
<td>$02</td>
<td>PORTE</td>
<td>PORTE7</td>
<td>PORTE6</td>
<td>PORTE5</td>
<td>PORTE4</td>
<td>PORTE3</td>
<td>PORTE2</td>
<td>PORTE1</td>
<td>PORTE0</td>
<td>87</td>
</tr>
<tr>
<td>$01</td>
<td>DDRE</td>
<td>DDE7</td>
<td>DDE6</td>
<td>DDE5</td>
<td>DDE4</td>
<td>DDE3</td>
<td>DDE2</td>
<td>DDE1</td>
<td>DDE0</td>
<td>84</td>
</tr>
<tr>
<td>$00</td>
<td>PINF</td>
<td>PINF7</td>
<td>PINF6</td>
<td>PINF5</td>
<td>PINF4</td>
<td>PINF3</td>
<td>PINF2</td>
<td>PINF1</td>
<td>PINF0</td>
<td>91</td>
</tr>
</tbody>
</table>

Note: For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI an SBI instructions work with registers $00 to $1F only.
### Instruction Set Summary (Continued)

#### DATA TRANSFER INSTRUCTIONS

<table>
<thead>
<tr>
<th>Mnemonics</th>
<th>Operands</th>
<th>Description</th>
<th>Operation</th>
<th>Flags</th>
<th>#Clocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>←</td>
<td>Rd, K</td>
<td>Load Immediate Rd</td>
<td>(Z+RAMPZ)</td>
<td>None</td>
<td>3</td>
</tr>
<tr>
<td>←</td>
<td>(X)</td>
<td>Load Indirect and Post-Inc. Rd</td>
<td>X - 1, Rd</td>
<td>(X)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, -X</td>
<td>Load Indirect and Pre-Dec. X</td>
<td>Rd - K</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Y</td>
<td>Load Indirect Rd</td>
<td>Rd - K - C</td>
<td>(Z + q)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Y+q</td>
<td>Load Indirect with Displacement Rd</td>
<td>Rd - K - C</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, -Z</td>
<td>Load Indirect and Pre-Dec. Z</td>
<td>Z - 1, Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, K</td>
<td>Logical AND Register and Constant Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Y+</td>
<td>Load Indirect and Post-Inc. Rd</td>
<td>Y - 1, Rd</td>
<td>(Y)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Y</td>
<td>Load Indirect Rd</td>
<td>Rd - K</td>
<td>(Y)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Y</td>
<td>Load Indirect Rd</td>
<td>Rd - K - C</td>
<td>(Y + q)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Y, Y</td>
<td>Load Indirect Rd</td>
<td>Rd - K - C</td>
<td>(Y)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Rr</td>
<td>Subtract with Carry two Registers Rd</td>
<td>Rd - K - C</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd</td>
<td>One's Complement Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd</td>
<td>Logical AND Registers Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Rr</td>
<td>Logical OR Registers Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Rr</td>
<td>Logical OR Registers Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Rr</td>
<td>Logical OR Registers Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Rr</td>
<td>Logical OR Registers Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
<tr>
<td>←</td>
<td>Rd, Rr</td>
<td>Logical OR Registers Rd</td>
<td>Rd</td>
<td>(Z)</td>
<td>2</td>
</tr>
</tbody>
</table>

### BRANCH INSTRUCTIONS

<table>
<thead>
<tr>
<th>Mnemonics</th>
<th>Operands</th>
<th>Description</th>
<th>Operation</th>
<th>Flags</th>
<th>#Clocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPM</td>
<td>R0</td>
<td>Load Program Memory R0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RJMP</td>
<td>k</td>
<td>Relative Jump PC</td>
<td>PC + k + 1</td>
<td>None</td>
<td>2</td>
</tr>
<tr>
<td>IJMP</td>
<td>(Z)</td>
<td>Indirect Jump to (Z) PC</td>
<td>PC + 2 or 3</td>
<td>None</td>
<td>1 / 2 / 3</td>
</tr>
<tr>
<td>RCALL</td>
<td>k</td>
<td>Relative Subroutine Call PC</td>
<td>PC + k + 1</td>
<td>None</td>
<td>3</td>
</tr>
<tr>
<td>ICALL</td>
<td>(Z)</td>
<td>Indirect Call to (Z) PC</td>
<td>PC + 2 or 3</td>
<td>None</td>
<td>1 / 2 / 3</td>
</tr>
<tr>
<td>RET</td>
<td></td>
<td>Subroutine Return PC</td>
<td>ST ACK</td>
<td>None</td>
<td>4</td>
</tr>
<tr>
<td>RETI</td>
<td></td>
<td>Interrupt Return PC</td>
<td>ST (Rd(n), Rd(0))</td>
<td>None</td>
<td>2</td>
</tr>
</tbody>
</table>

### CONDITION CODES

<table>
<thead>
<tr>
<th>Condition Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCLR</td>
<td>Flag Clear SREG(s)</td>
</tr>
<tr>
<td>BSET</td>
<td>Flag Set SREG(s)</td>
</tr>
<tr>
<td>BREQ</td>
<td>Branch if Equal</td>
</tr>
<tr>
<td>BRCC</td>
<td>Branch if Carry Cleared</td>
</tr>
<tr>
<td>BRCH</td>
<td>Branch if Carry Set</td>
</tr>
<tr>
<td>BRGE</td>
<td>Branch if Greater or Equal, Signed</td>
</tr>
<tr>
<td>BRHS</td>
<td>Branch if Half Carry Flag Set</td>
</tr>
<tr>
<td>BRHC</td>
<td>Branch if Half Carry Flag Cleared</td>
</tr>
<tr>
<td>BRLO</td>
<td>Branch if Lower</td>
</tr>
<tr>
<td>BRNE</td>
<td>Branch if Not Equal</td>
</tr>
<tr>
<td>BRRC</td>
<td>Branch if Relative</td>
</tr>
<tr>
<td>BRSH</td>
<td>Branch if Same or Higher</td>
</tr>
<tr>
<td>BRVS</td>
<td>Branch if Overflow Flag is Set</td>
</tr>
<tr>
<td>BRIE</td>
<td>Branch if Interrupt Enabled</td>
</tr>
<tr>
<td>BRID</td>
<td>Branch if Interrupt Disabled</td>
</tr>
</tbody>
</table>

Note: 1. Not in ATmega603.
In this appendix are available some electric diagrams of the most frequently used GPC® AM4 interfaces. All these interfaces can be yourself produced and some of them are standard grifo® cards and, if required, they can be directly ordered.

**Figure C1: PPI Expansion Electric Diagram**
Figure C2: SPA 03 Electric Diagram
**Title:** QTP 16P  
**Date:** 22-07-1998  
**Rel.:** 1.2  
**Page:** 1 of 1

---

**FIGURE C3: QTP 16P ELECTRIC Diagram**
FIGURE C4: QTP 24P ELECTRIC DIAGRAM (1 OF 2)
FIGURE C5: QTP 24P ELECTRIC DIAGRAM (2 OF 2)
Figure C6: ABACO® I/O BUS INPUT OUTPUT ELECTRIC DIAGRAM
FIGURE C7: BUS INTERFACE ELECTRIC DIAGRAM
FIGURE C8: IAC 01 electric diagram
APPENDIX D: ALPHABETICAL INDEX

A
A/D
conversion time 8
diagram 21
impedance 9
inputs 9, 23
resolution 8
A/D converter 6, 8, 20, 23
ABACO® I/O BUS 3, 11, 26, 35, C-6
Address BUS 11, 35
Addresses 35
Assistance 1

B
Back up 4, 9, 10, 30
Battery 4, 9, 10, 17, 25, 30
Baud rate 28
Bibliography 46
Block diagram 5

C
Calibration 23
Clock 3, 8
Comparator 9, 18, 23
Component maps 33
Configuration input 25, 30, 39
Connections 23, 42
Connectors 8, 10, 17, A-1
CN1 11
CN2 10, 30
CN3 12, 26, 30
CN5 18
CN6 20
CN9 22
Consumption 3, 9
Container 1, 8, A-1
Control logic 35
CPU 4, 8, 26, 39, B-1
Crystal 3, 8
Current 9
Current loop 3, 12, 16, 23, 28
D
Data BUS 11, 35
Devices 8, 35, 36, 39
Dimension A-1
DIR signal 26, 28, 30
Disable control signals 36, 41

E
EEPROM 4, 8, 38
Electric diagrams C-1
Expansion 3, 6, 42
External cards 42

F
Features
electric 9
general 2, 8
physical 8
technical 8
FLASH EPROM 4, 8, 37
Frequency 3, 8

H
Humidity 9

I
I/O addresses 36
I/O interfaces 24
I/O lines 6, 18, 24, 26, 44, C-1, C-7
I/O signals 19
Installation 1, 10
Interrupts 26
Intervent time 8, 32
ISP 7, 22, 24, 38
Instructions 4

J
Jumpers 25
2 pins 25
3 pins 26
Jumpers location 27

L
LED 17, 24
M
Maps 35
Memory 4, 17, 37
Memory cards 24, 42
Micro ISP 7, 22, 34
Mounting A-1
Multiplexed pins 19, 31

N
Network 15

O
Operator panels 24, 42, C-3, C-4
Options 28, A-1

P
Photo 7
Piggy back mounting A-2
Ports 18, 31
Power supply 3, 9
PPI expansion C-1
Printer 24, C-8
Programming 7, 24
PWM 6, 18, 20

R
Rail mounting A-2
RAM 4, 8, 37
Real Time Clock 6, 26, 36, 39
Registers 6, 36
Release 1
Reset 22, 32
RS 232 3, 12, 14, 23, 28
RS 422 3, 9, 12, 14, 23, 25, 26, 28
RS 485 3, 9, 12, 14, 23, 25, 26, 28

S
Serial communication 13
Serial drivers 29
Serial line 3, 12, 23, 28
Size 8
Software 34
Solder jumpers 26
SPI 18
T
Temperature 9
Termination circuit 9, 15, 30
Timer counters 6, 18, 20
TransZorb™ 3
Trimmer 17, 23
TTL 23

V
Version 1
Voltage reference 21

W
Warranty 1
Watch dog 6, 8, 32
Weight 8