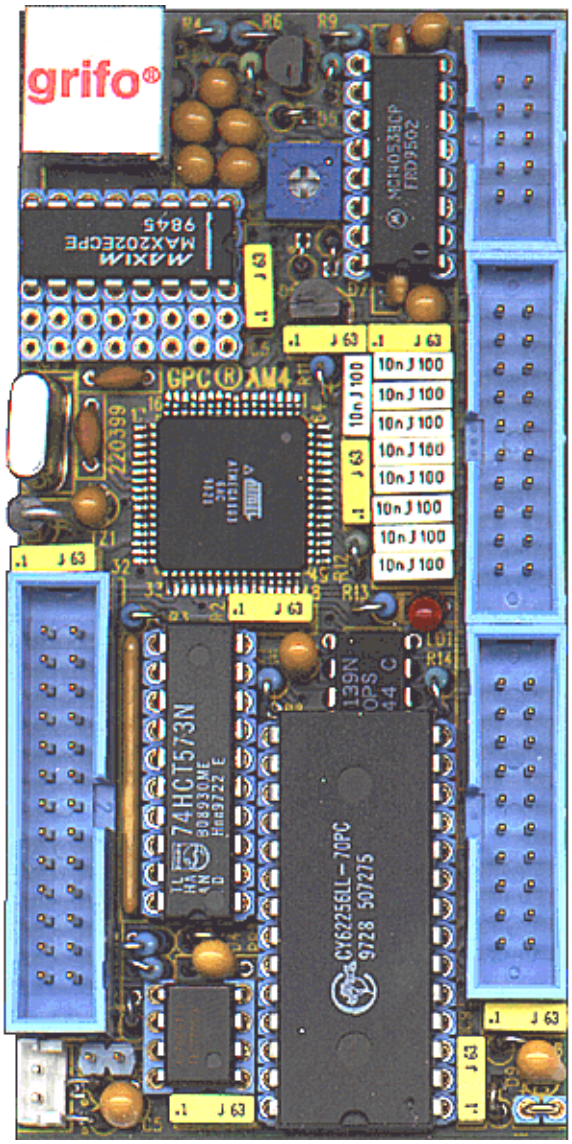
GPC® AM4

General Purpose Controller AVR Mega, 4 type

TECHNICAL MANUAL



grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
Email: grifo@grifo.it

http://www.grifo.it http://www.grifo.com
Tel. +39 051 892052 (a.r.) FAX +39 051 893661

GPC® AM4 Edition 5.00 *Rel. 24 November 1999*

abaco®, **GPC®**, **grifo®**, *are trade marks of* **grifo®**

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.

grifo®
ITALIAN TECHNOLOGY
Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
Email: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892052 (a.r.) FAX +39 051 893661

GPC® AM4 Edition 5.00 *Rel. 24 November 1999*

abaco®, **GPC®**, **grifo®**, *are trade marks of* **grifo®**

DOCUMENTATION COPYRIGHT BY grifo®, ALL RIGHTS RESERVED.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, either electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written consent of **grifo**®.

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® reservs 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

abaco®, **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

SOFTWARE 34

DEVICES MAP AND ADDRESSES 35
INTRODUCTION 35
ON BOARD DEVICES ADDRESSES 35
ABACO® I/O BUS ADDRESSES 35
I/O ADDRESSES 36
MEMORY ADDRESSES 37

PERIPHERAL DEVICES SOFTWARE DESCRIPTION 39
CPU INTERNAL PERIPHERALS 39
CONFIGURATION INPUT (J1) 39
REAL TIME CLOCK 39
DISABLE CONTROL SIGNALS 41

EXTERNAL CARDS 42

BIBLIOGRAPHY 46

APPENDIX A: CARD MECHANICAL MOUNTING A-1

APPENDIX B: ON BOARD DEVICES DESCRIPTION B-1

APPENDIX C: ELECTRIC DIAGRAMS C-1

APPENDIX D: ALPHABETICAL INDEX D-1

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 rispect 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 aquisition 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 Peipheral 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 developping software tools which allow the card to be used either in assembly or with evoluted languages. Noteworthy among these are the developement 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 developement 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 pleare 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 aganist 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 manifacturer 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** or **RS 485**, **RS 422**; in this last cases also directionality and line activation is programmable.

CLOCK

GPC® AM4 is provided with a circuitery that generates the CPU clock frequency; this frequence 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 manifactured 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 istrunctions (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 manifacturer 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.

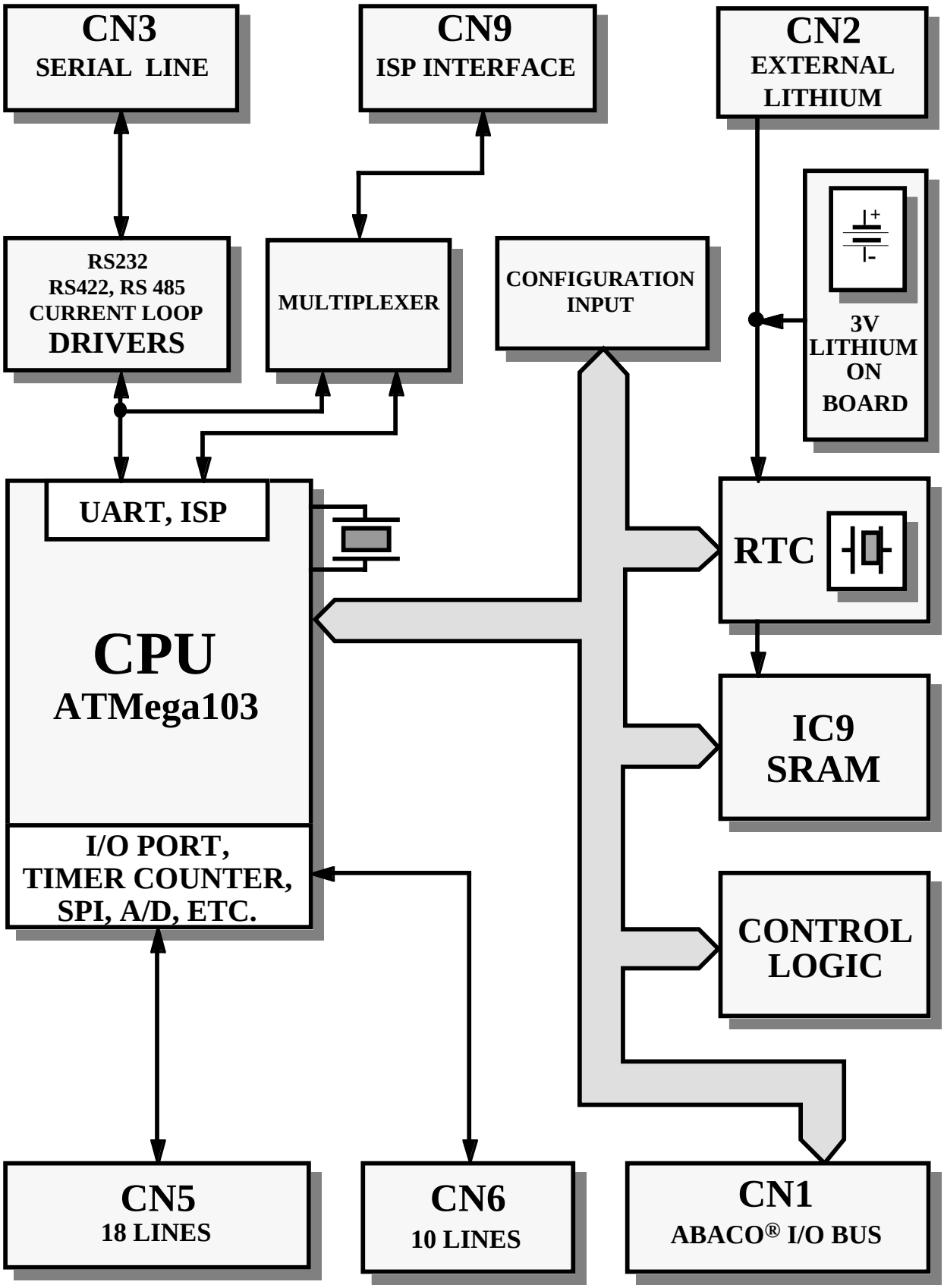


FIGURE 1: BLOCK DIAGRAM

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 circuitery that, if used, allows to exit from infinite loop or abnormal conditions not managed by the application program. This circuitery 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

Thecard 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 wich 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 an extremely fast programming speeds due to the use of dedicated hardware to generate the ISP waveforms.

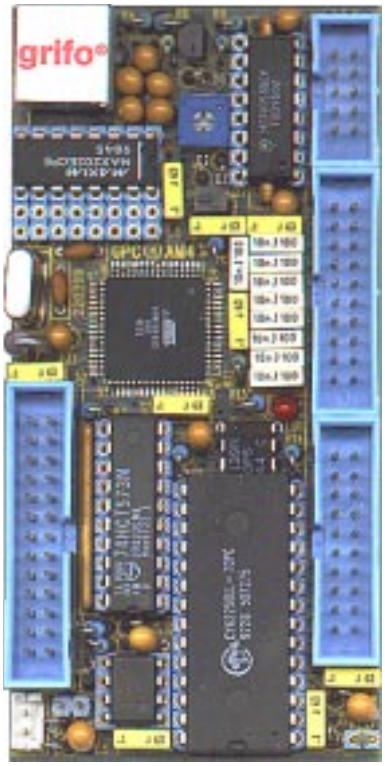


FIGURE 2: CARD PHOTO

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 degreeses
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).

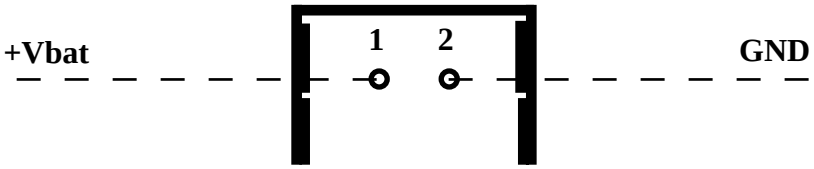


FIGURE 3: CN2 - EXTERNAL BACK UP BATTERY CONNECTOR

Signals description:

+Vbat	=	I	-	Positive pin of external back up battery
GND	=		-	Negative pin of external back up battery

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 exapnsion 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.

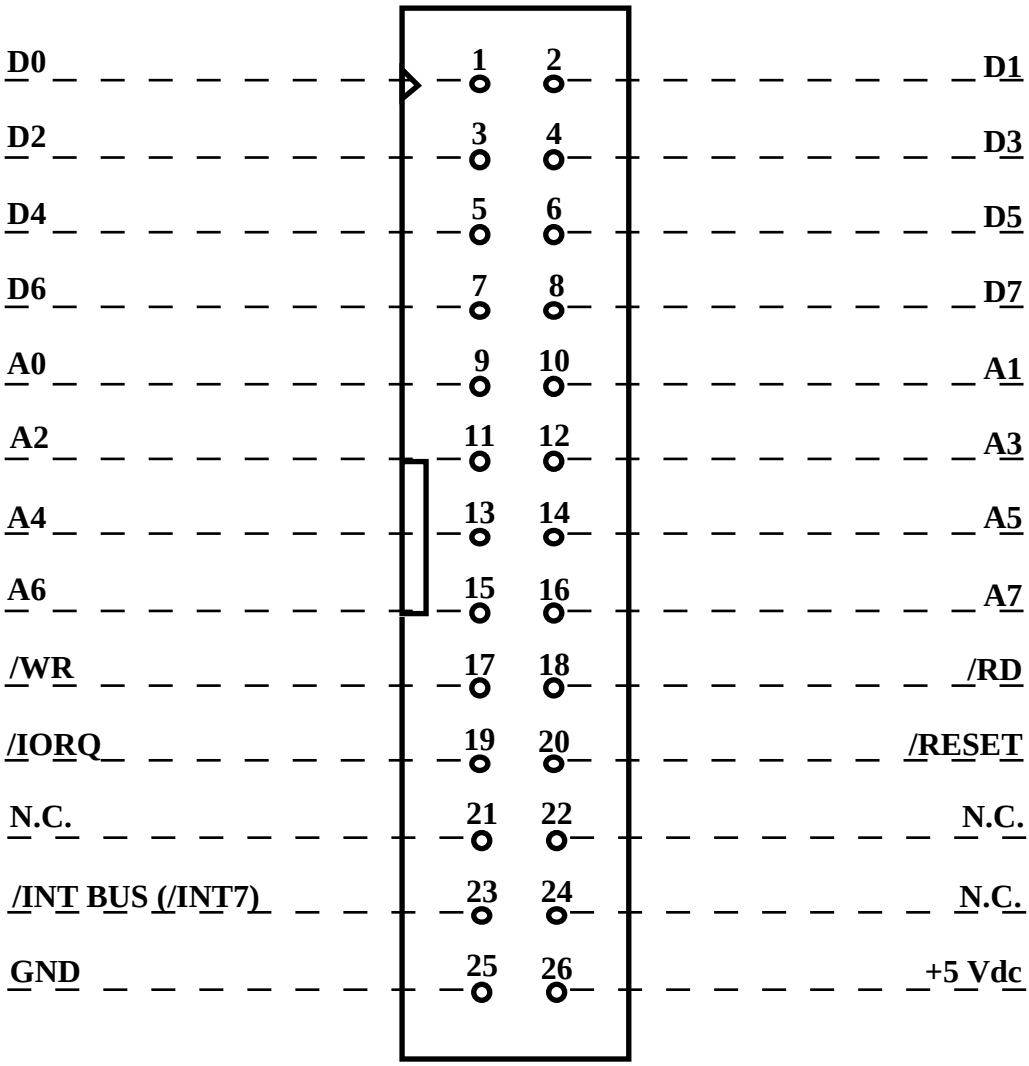


FIGURE 4: CN1 - ABACO® I/O BUS CONNECTOR

Signals description:

A0-A7	=	O	- Address BUS.
D0-D7	=	I/O	- Data BUS.
/INT BUS (/INT7)	=	I	- Interrupt requestconnected 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

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.

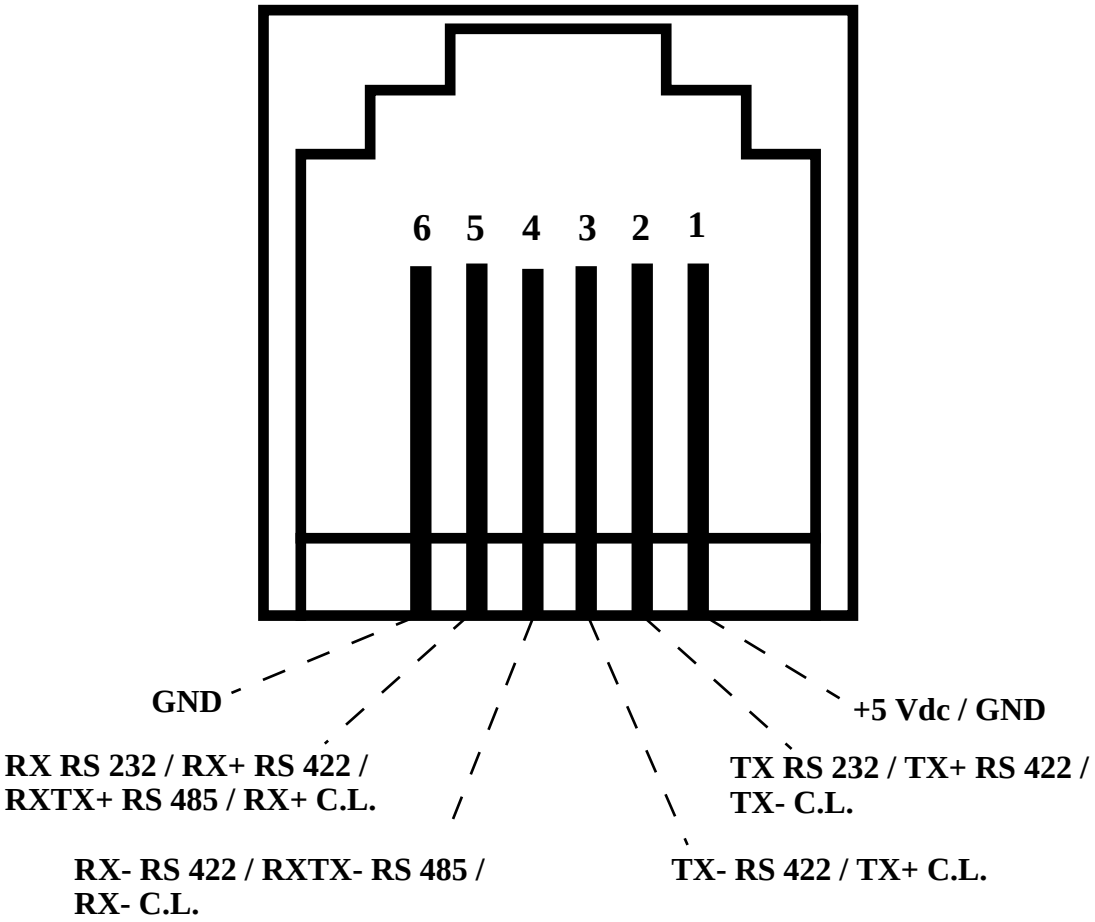


FIGURE 5: CN3 - SERIAL LINE CONNECTOR

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/0-	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.

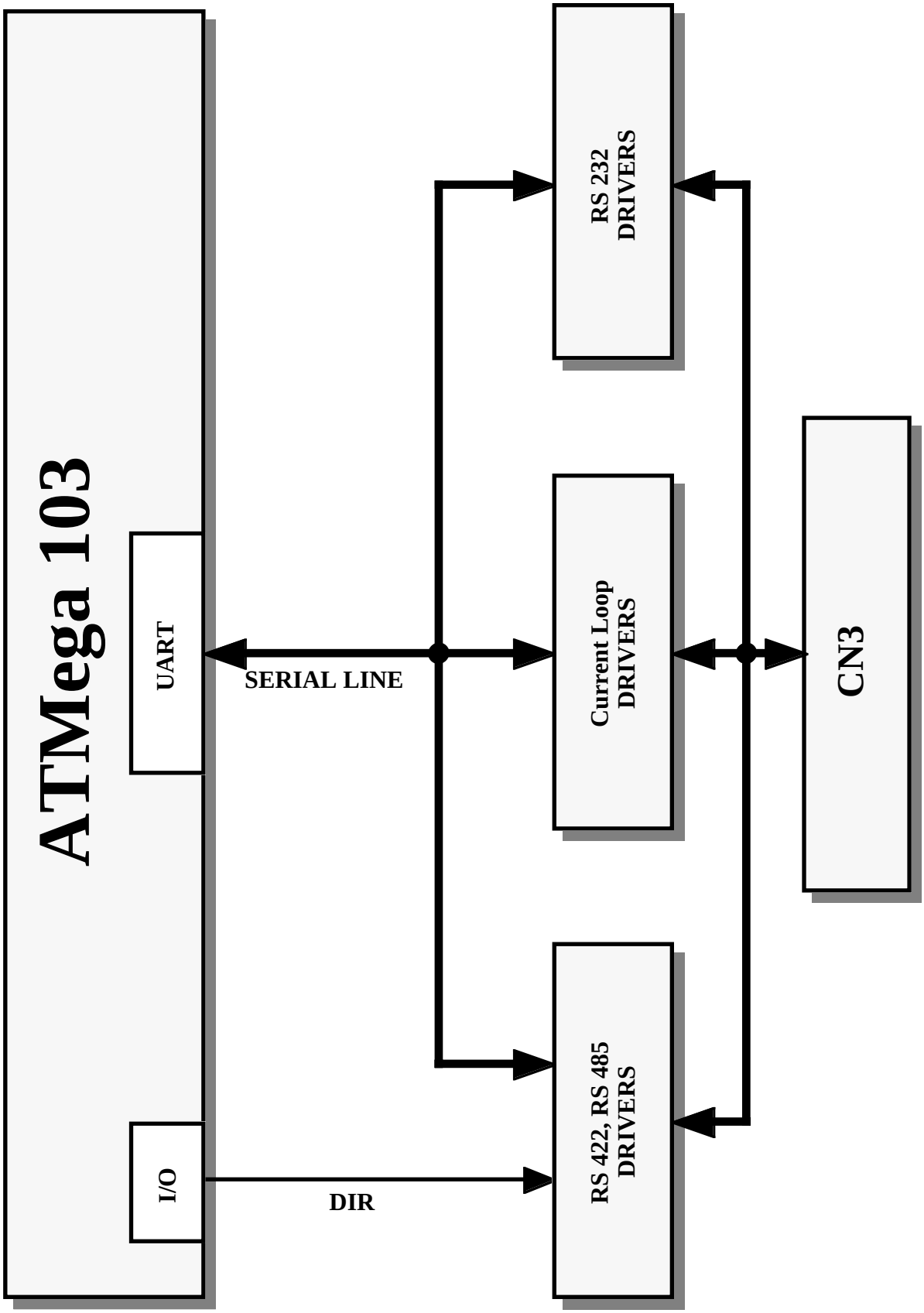


FIGURE 6: SERIAL COMMUNICATION DIAGRAM

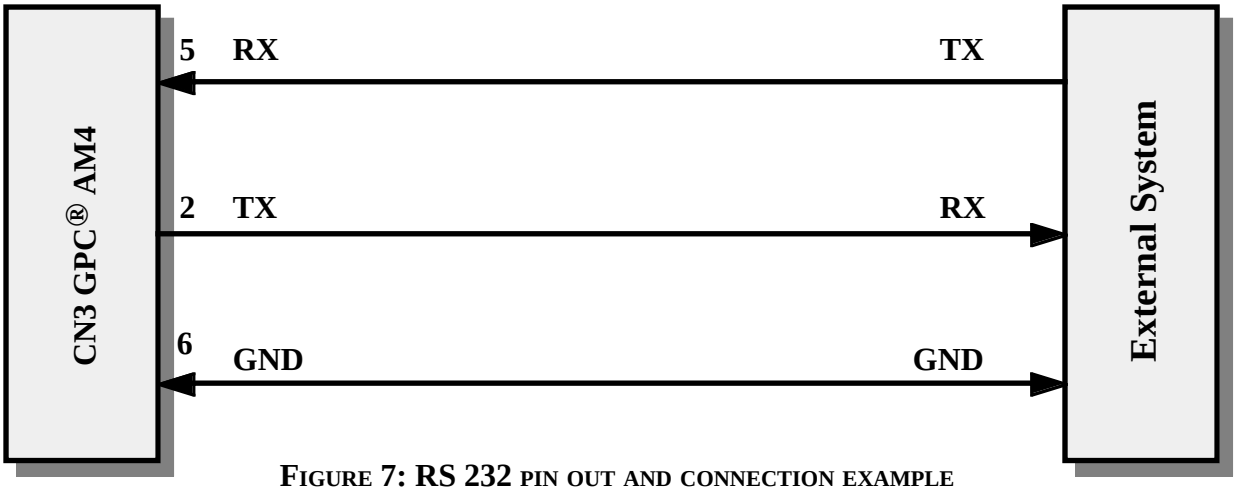


FIGURE 7: RS 232 PIN OUT AND CONNECTION EXAMPLE

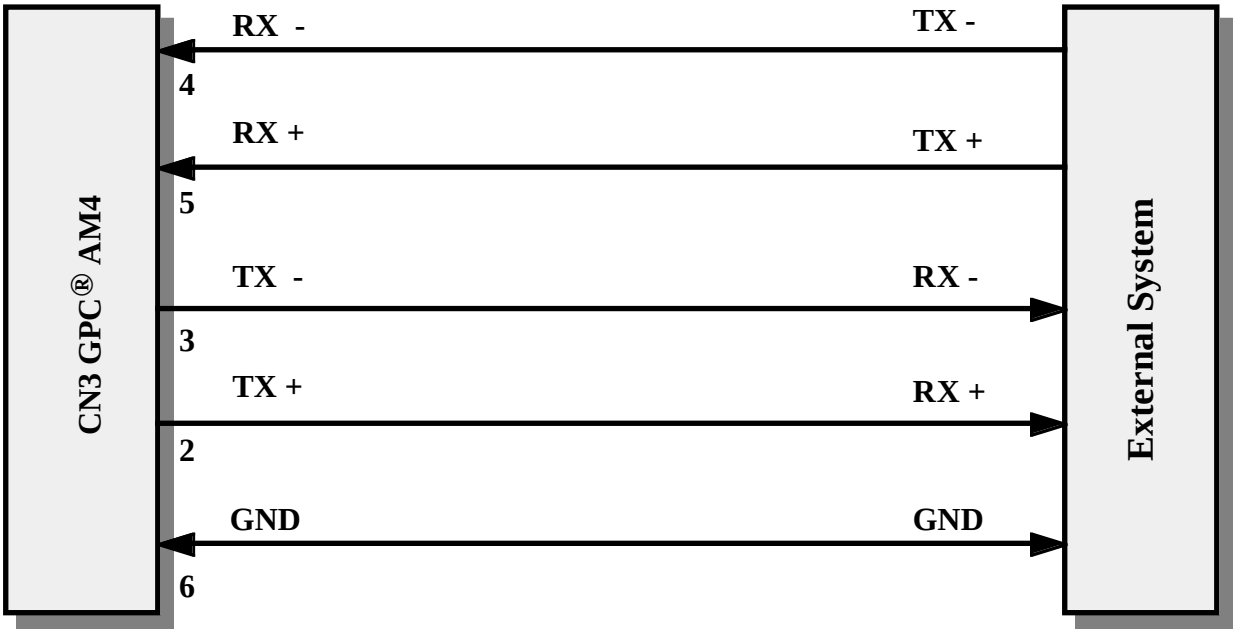


FIGURE 8: RS 422 PIN OUT AND POINT TO POINT CONNECTION EXAMPLE

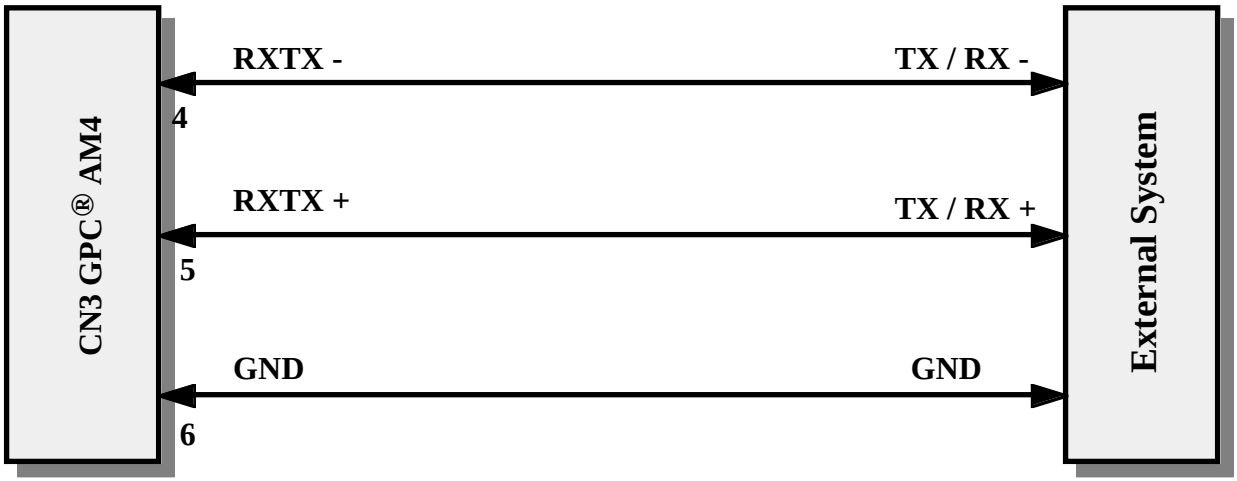


FIGURE 9: RS 485 PIN OUT AND POINT TO POINT CONNECTION EXAMPLE

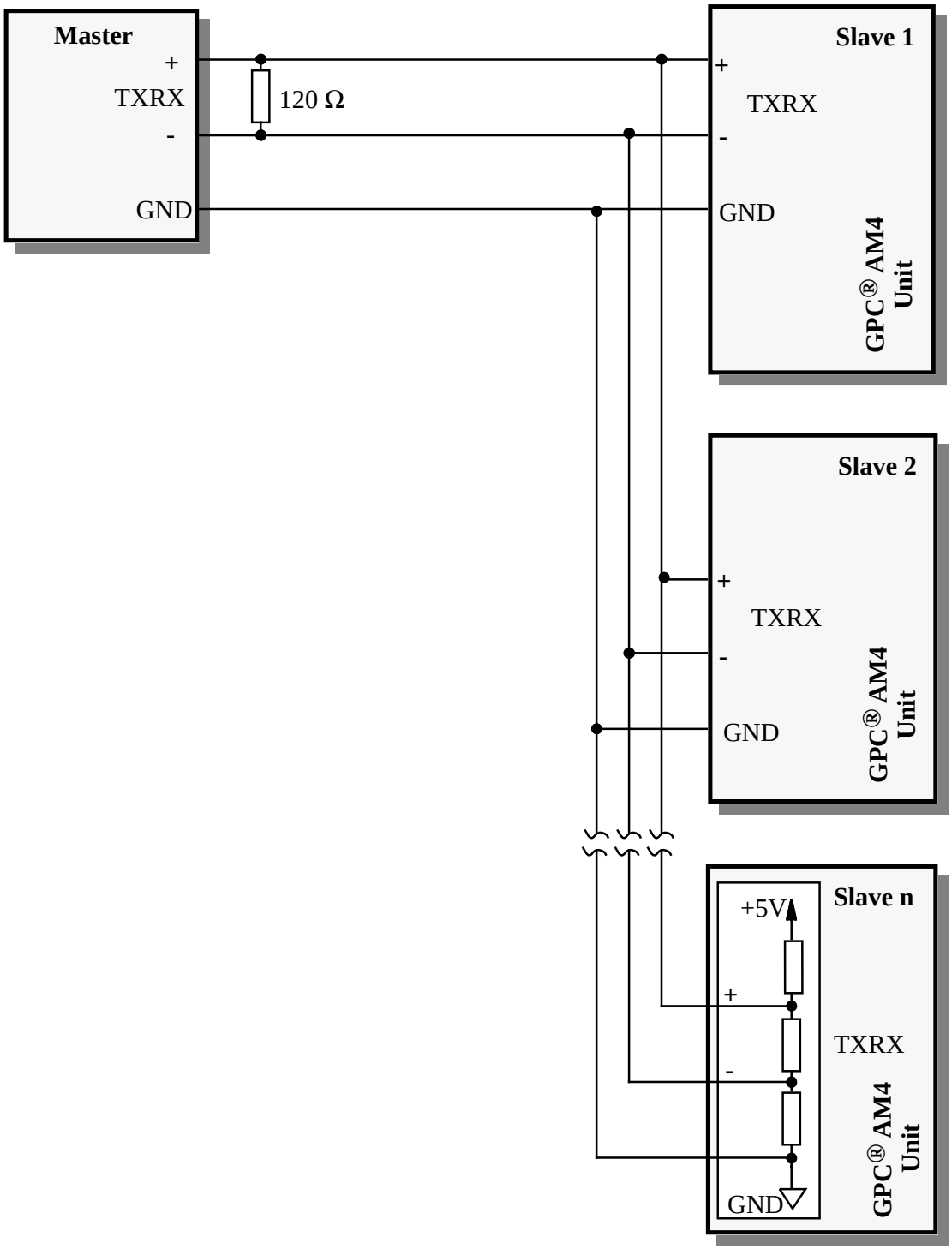


FIGURE 10: RS 485 PIN OUT AND NETWORK CONNECTION EXAMPLE

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.

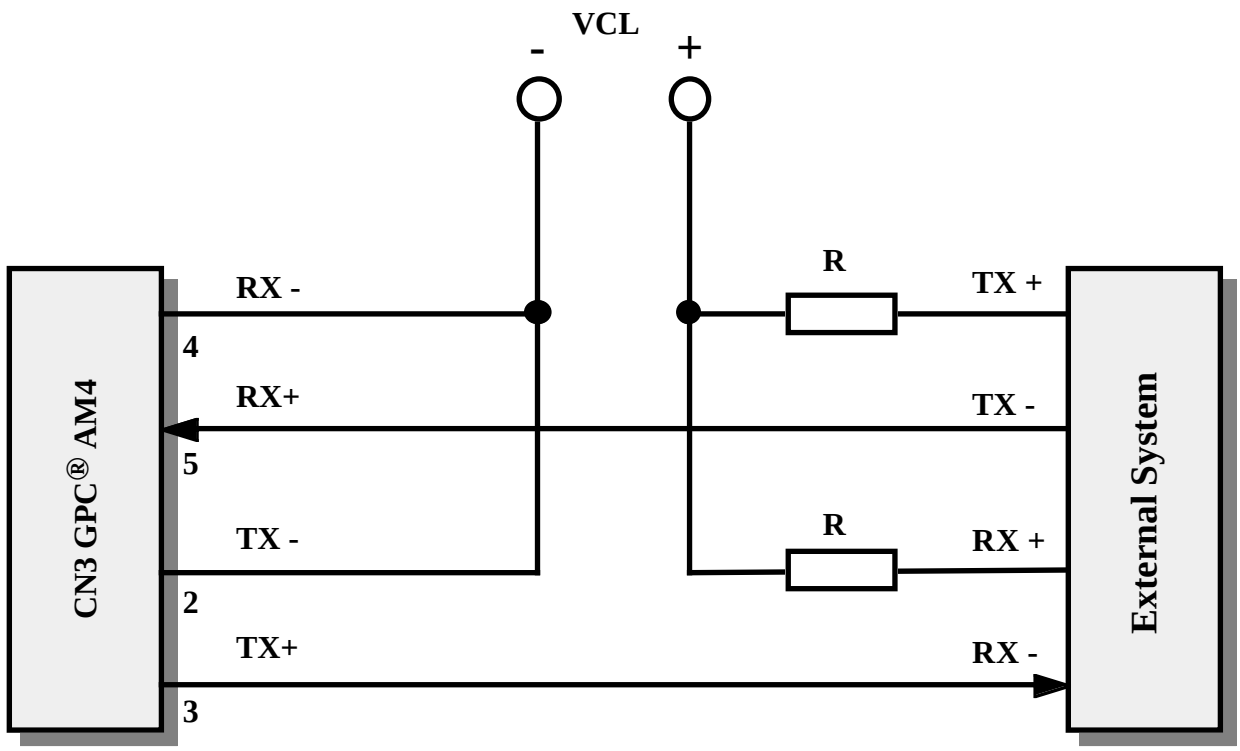


FIGURE 11: 4 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

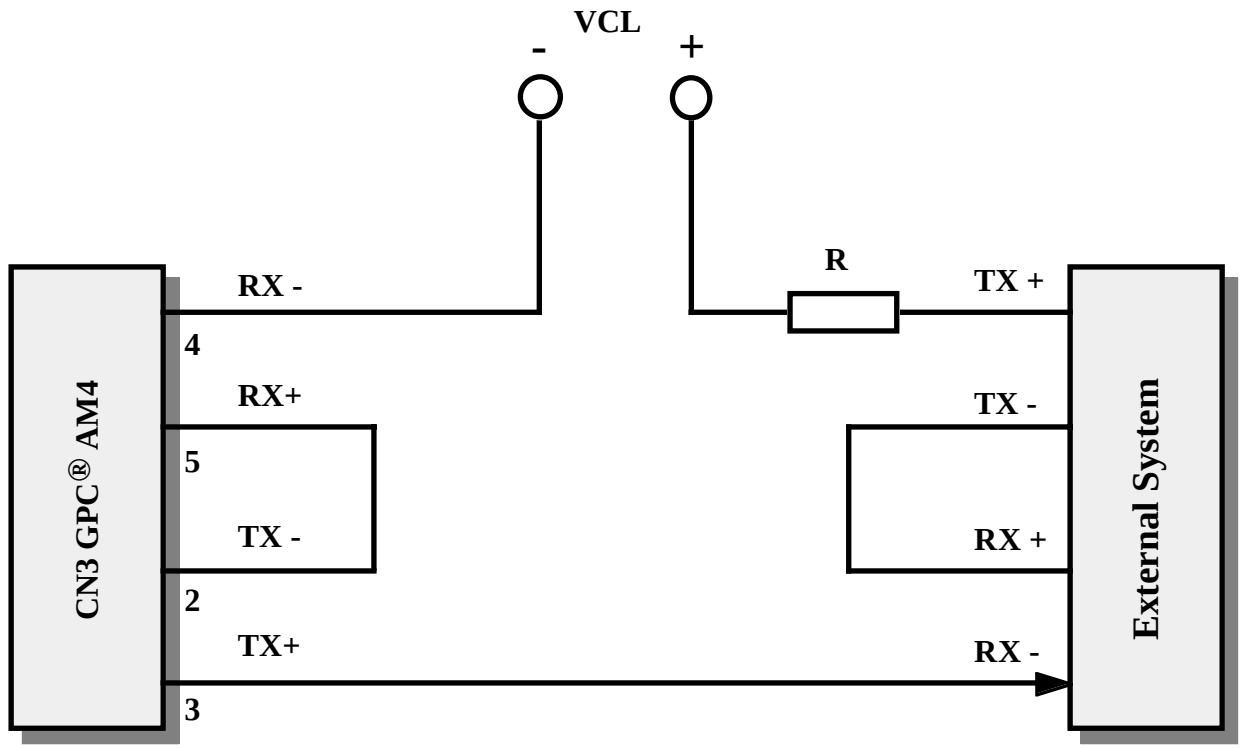
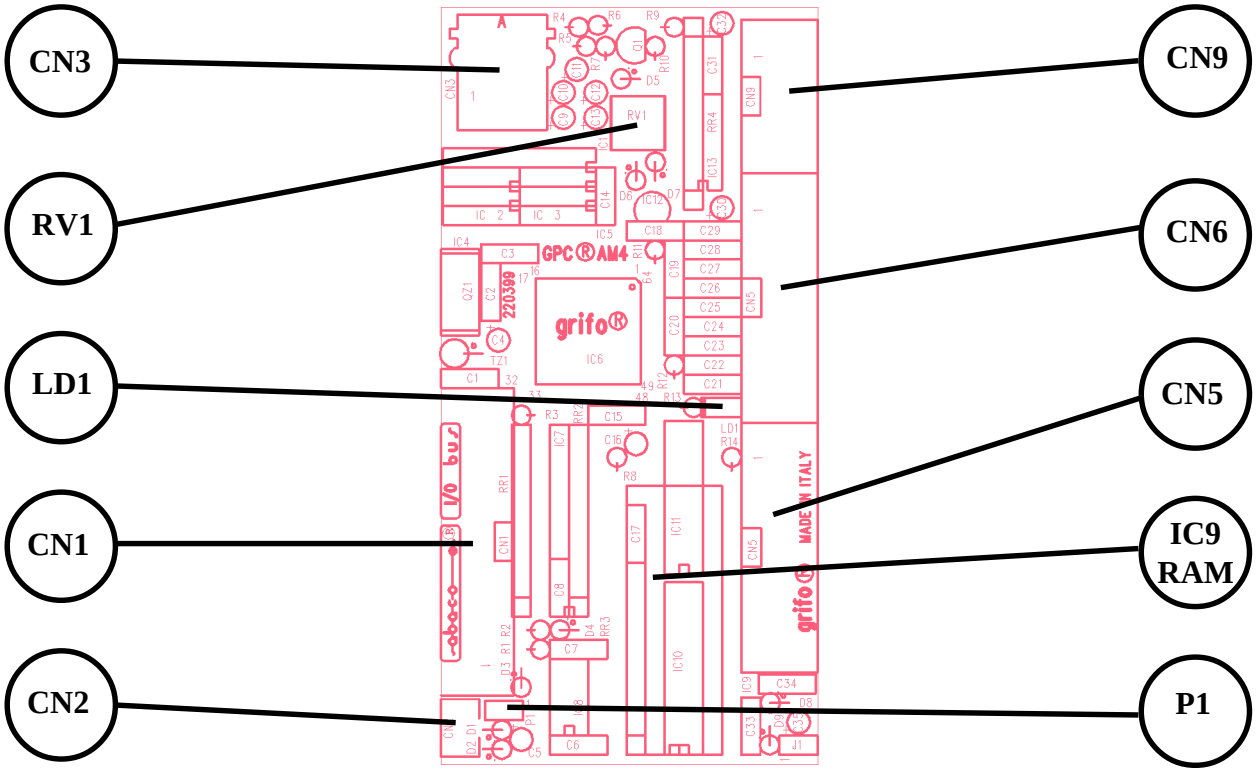


FIGURE 12: 2 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE



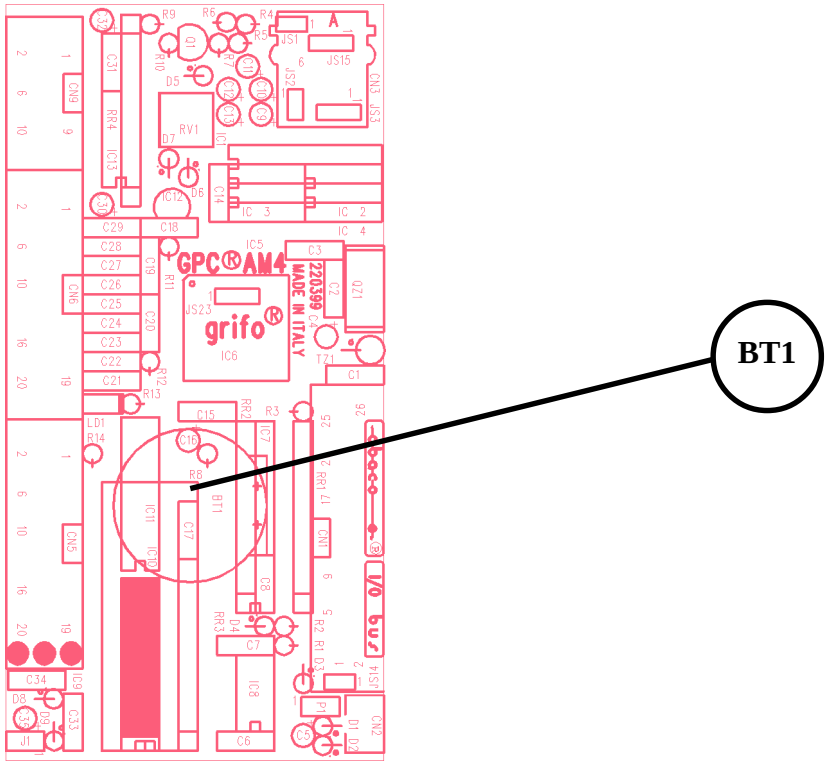


FIGURE 13: CONNECTORS, MEMORIES, TRIMMER, BATTERY, ETC. LOCATION

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.

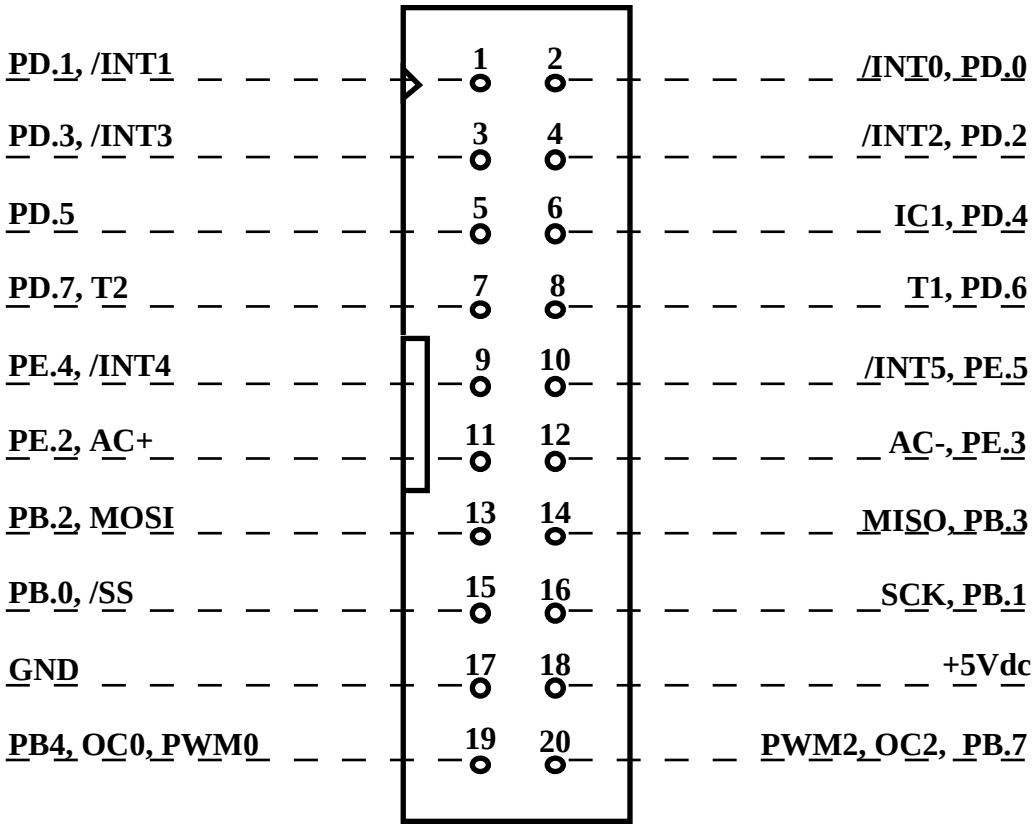


FIGURE 14: CN5 - DIGITAL I/O LINES CONNECTOR

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.

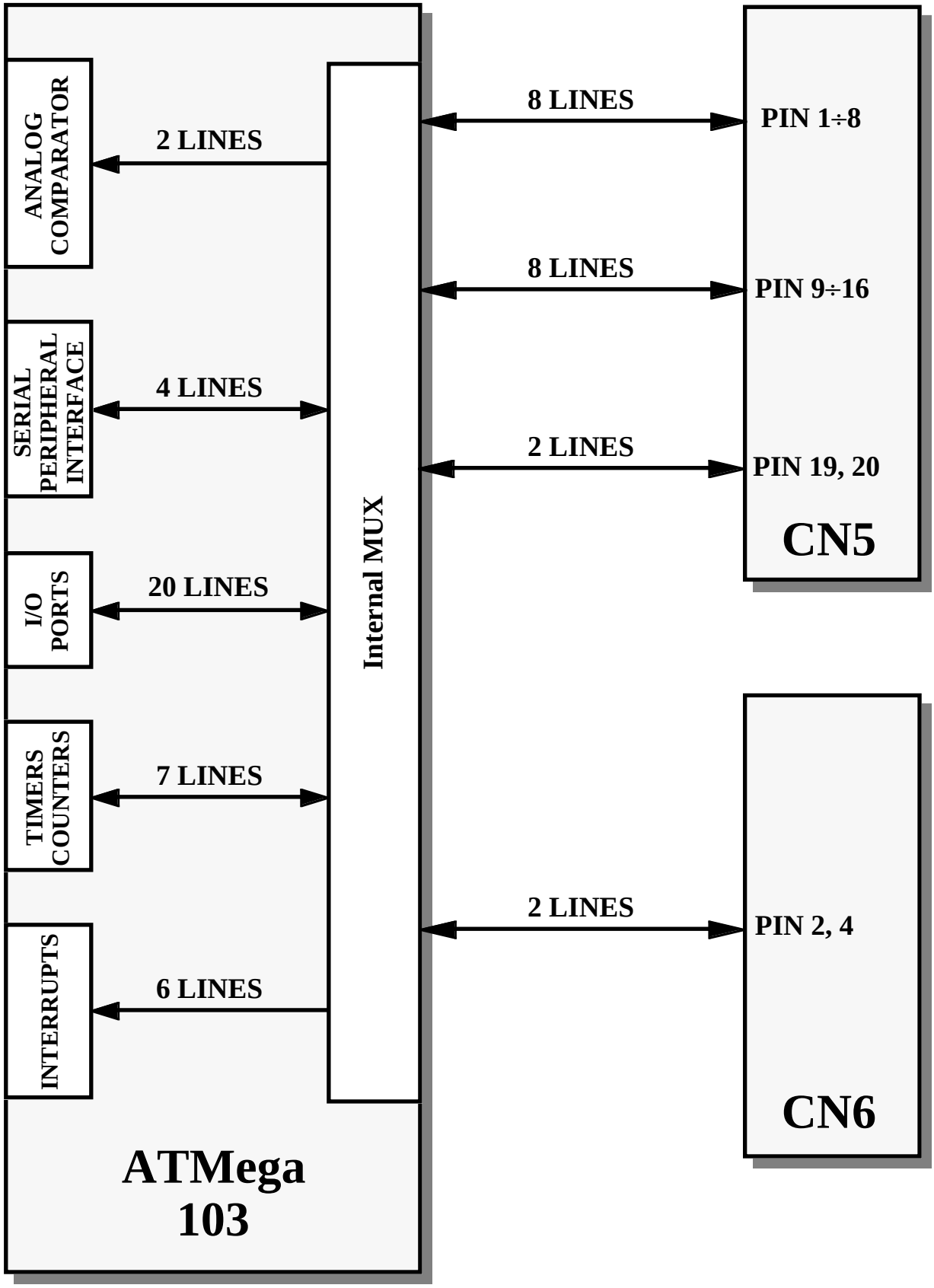


FIGURE 15: DIGITAL I/O SIGNALS CONNECTION DIAGRAM

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 outputlines, 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.

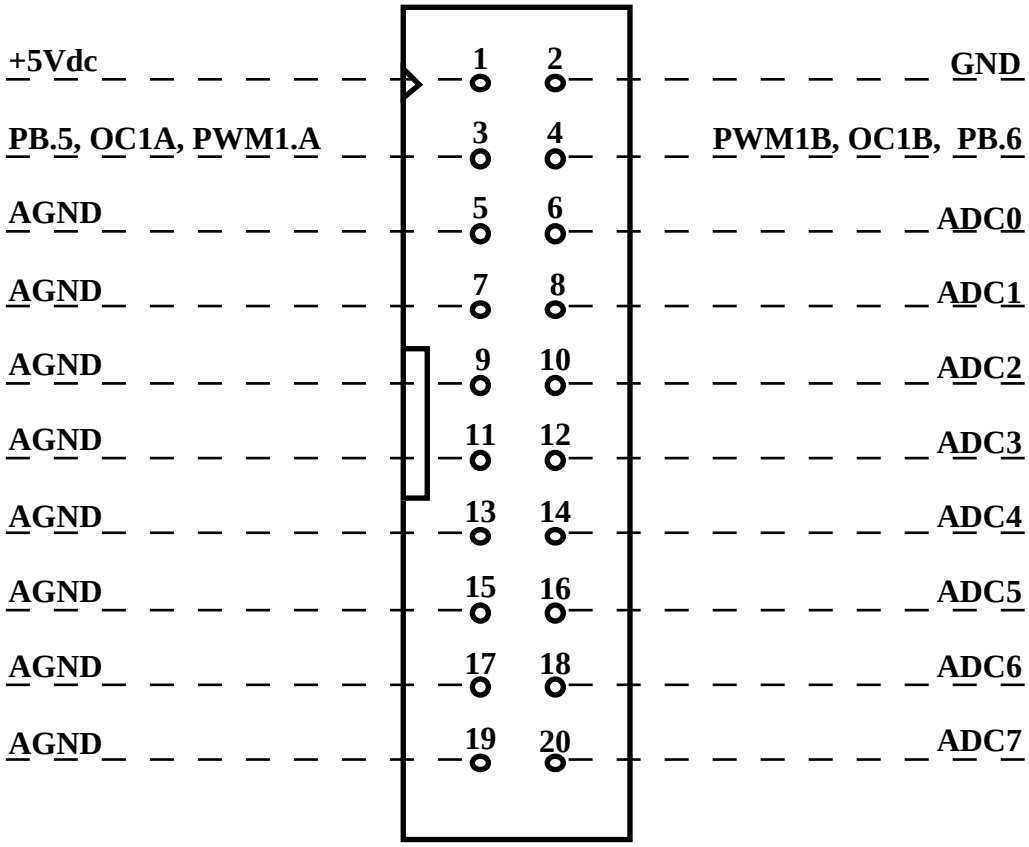


FIGURE 16: CN6 - A/D CONVERTER CONNECTOR

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.

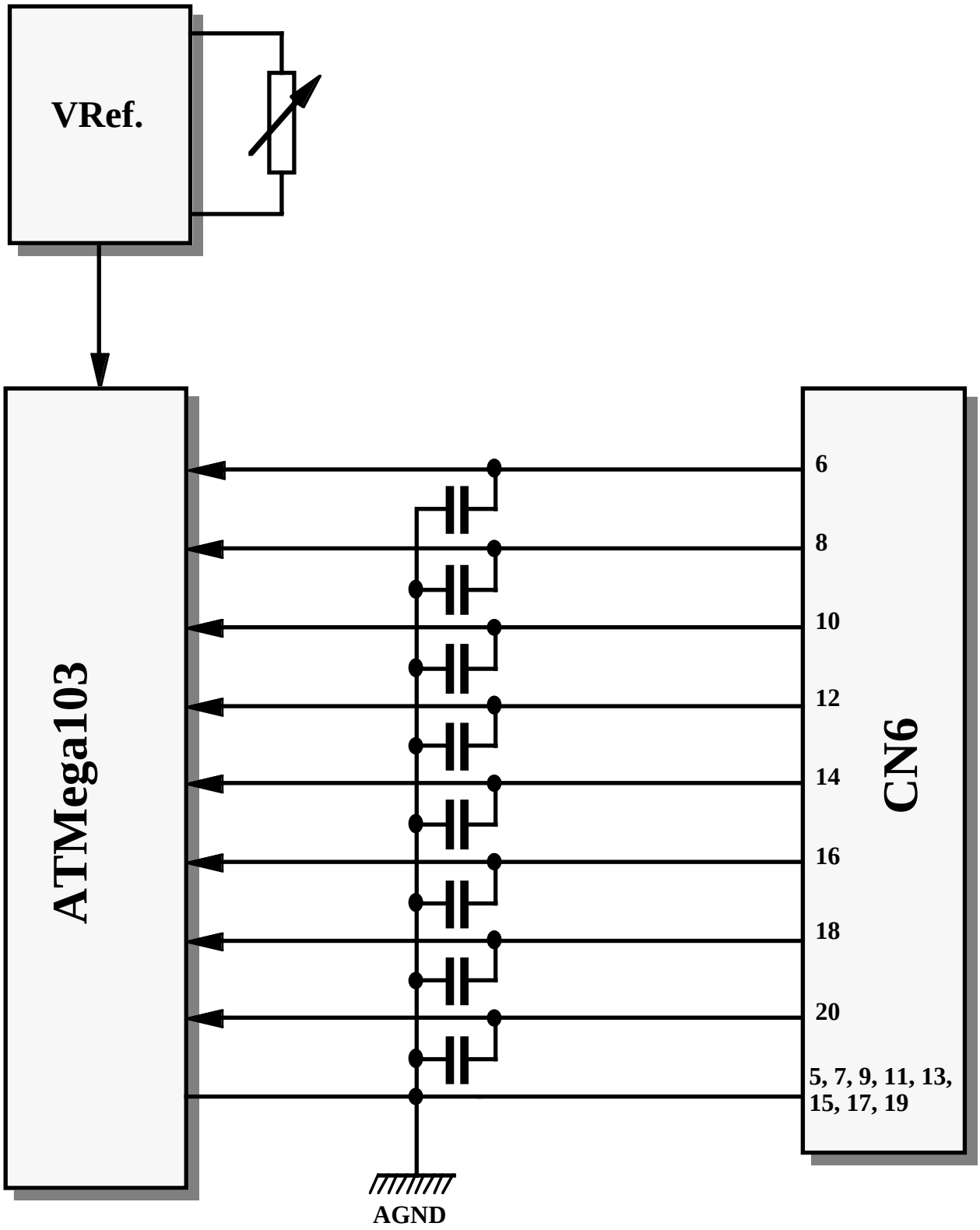


FIGURE 17: A/D CONVERTER INPUTS DIAGRAM

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.

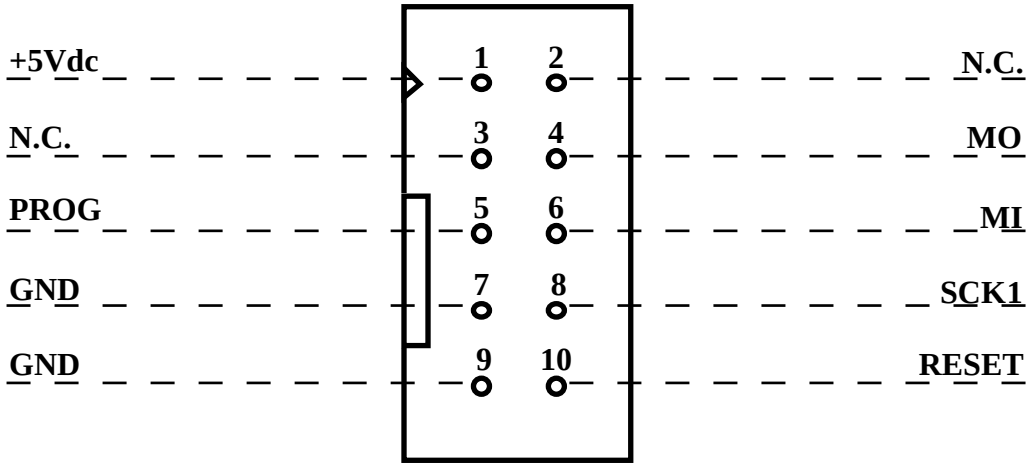


FIGURE 18: CN9 - ISP INTERFACE CONNECTOR

Signals description:

PROG	=	I	- ISP program selection signal.
MO	=	I	- ISP master output signal.
MI	=	O	- ISP master input signal.
SCK1	=	I	- ISP serial clock 1 signal.
RESET	=	I	- Target RESET control signal.
+5 Vdc	=	O	- +5 Vdc power supply.
GND	=		- Ground signal.
N.C.	=		- Not connected.

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 carefull test that verifies and calibrates all the card sections. The calibration is executed in laboratory, with a conntrolled +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 corrispondance 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 mantain calibration also in presence of mechanic stresses (vibrations, movings, 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.
The user must not modify the card calibration, but if thermic drifts,time drifts and so on, make necessary a new calibration, the user must strictly follow the previous described procedure.
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 instrunctions:

- 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 corrisponds 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 bandwith 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:

LED	COLOUR	DESCRIPTION
LD1	Red	Indicates the in system programmation status: active when ISP is enabled and viceversa (CN9 PROG signal).

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 workig 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.

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

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

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

FIGURE 21: 2 PINS JUMPERS TABLE

3 PINS JUMPERS

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

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 simultaneosly active.

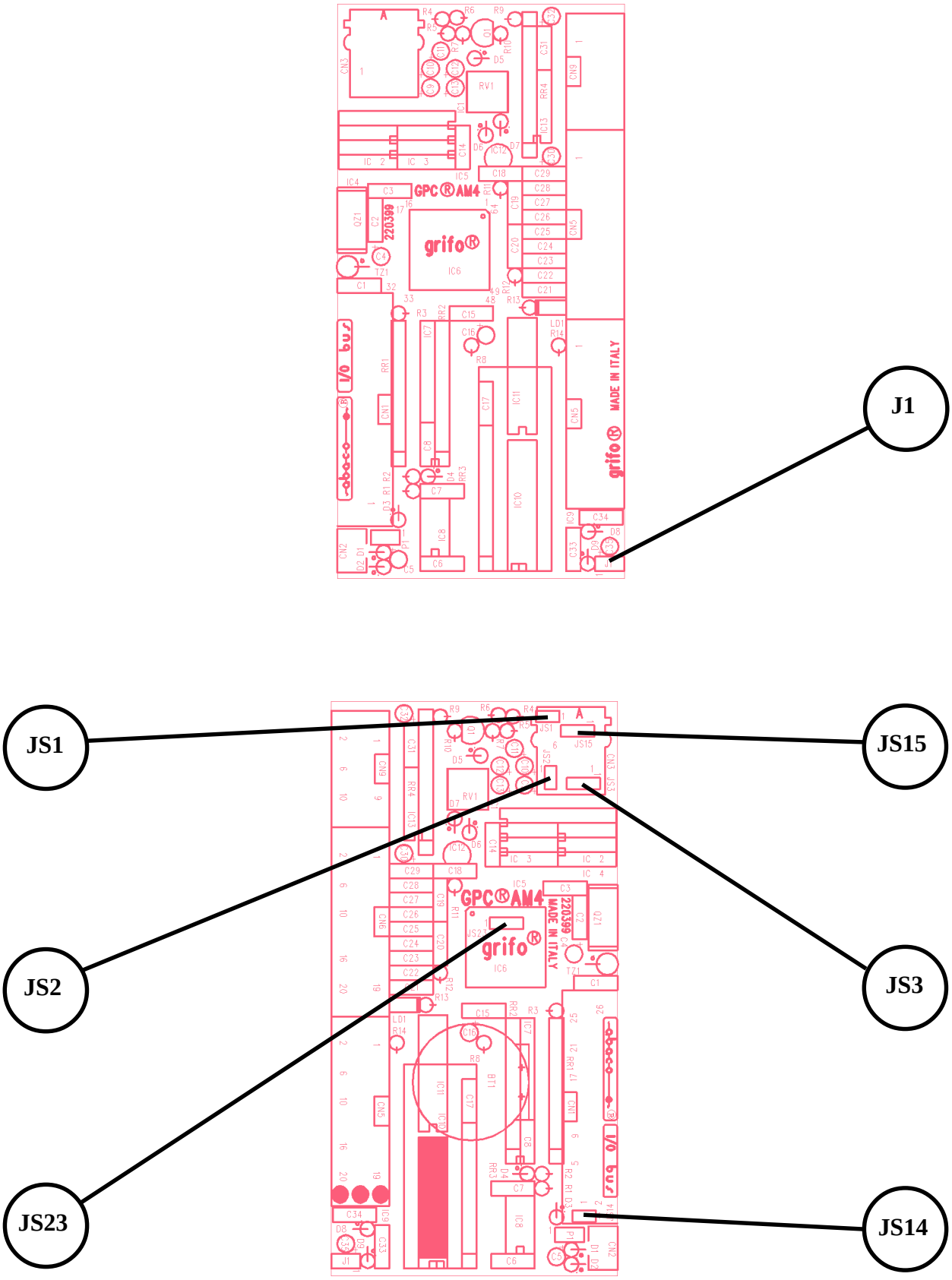


FIGURE 23: JUMPERS LOCATION

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)

			IC1	=	MAX 202 driver
JS1, JS2	=	not connected	IC2	=	no component
JS15	=	not connected	IC3	=	no component
JS23	=	position 1-2	IC4	=	no component
			IC5	=	no component

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

			IC1	=	no component
JS1, JS2	=	not connected	IC2	=	no component
JS15	=	not connected	IC3	=	no component
JS23	=	position 1-2	IC4	=	HP 4200 driver
			IC5	=	HP 4100 driver

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.

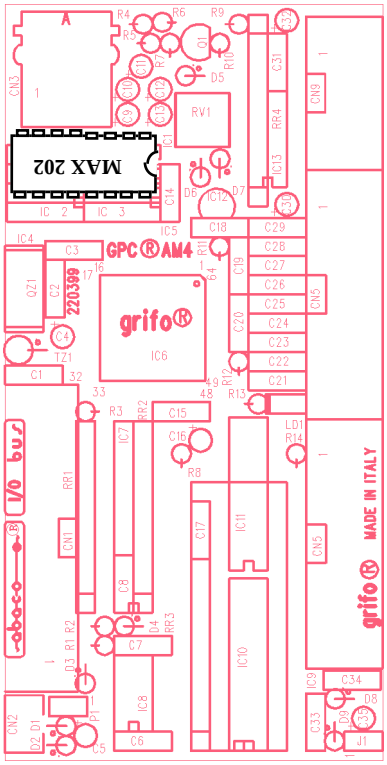
- SERIAL LINE CONFIGURED IN RS 422 (.RS422 option)

			IC1	=	no component
JS1, JS2	=	(*)	IC2	=	SN 75176 driver
JS15	=	position 2-3	IC3	=	SN 75176 driver
JS23	=	position 2-3	IC4	=	no component
			IC5	=	no component

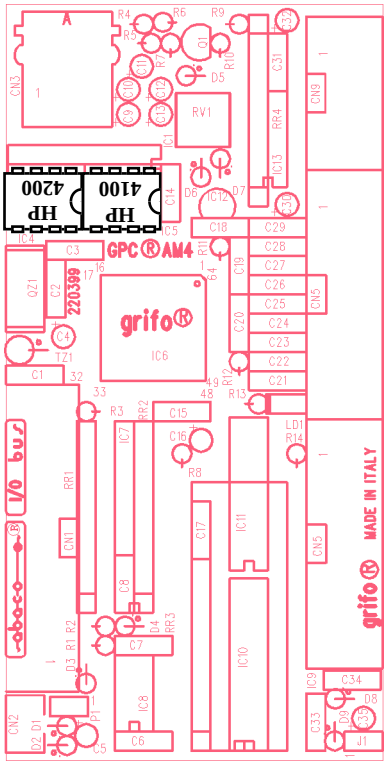
With PE.6=DIR signal (managed by software with PORT E registers) the user enables or disables the transmitter driver:

DIR	= low level	= 0 logic state	-> transmitter driver enabled
DIR	= high level	= 1 logic state	-> transmitter driver disabled

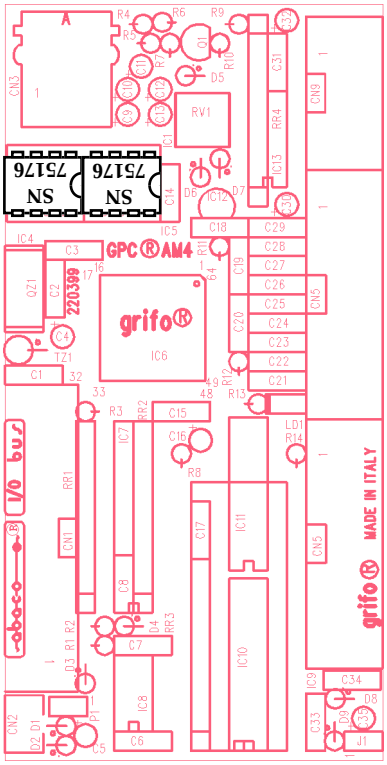
allowing either point to point (driver can be mantained always enabled) or network (driver is enabled only when the unit can hold the line) connection.



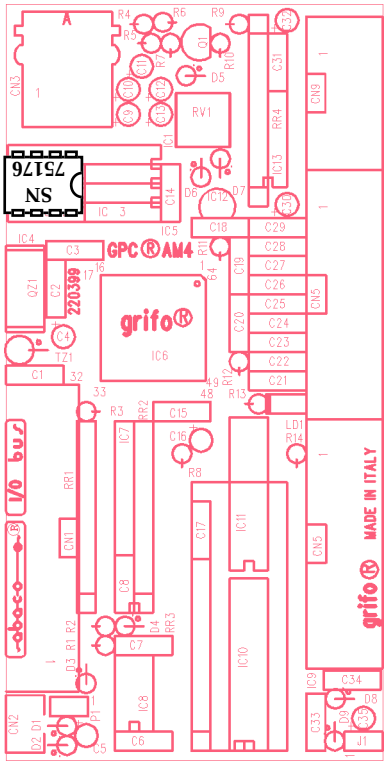
RS 232 serial line



Current loop serial line



RS 422 serial line



RS 485 serial line

FIGURE 24: SERIAL COMMUNICATION DRIVERS LOCATION

- SERIAL LINE CONFIGURED IN RS 485 (.RS485 option)

			IC1	=	no component
JS1, JS2	=	(*)	IC2	=	SN 75176 driver
JS15	=	position 1-2	IC3	=	no component
JS23	=	position 2-3	IC4	=	no component
			IC5	=	no component

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

DIR	= low level	= 0 logic state	->	RS 485 line transmitting
DIR	= high level	= 1 logic state	->	RS 485 line receiving

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 mantain 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 ha 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 phisically 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:

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

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

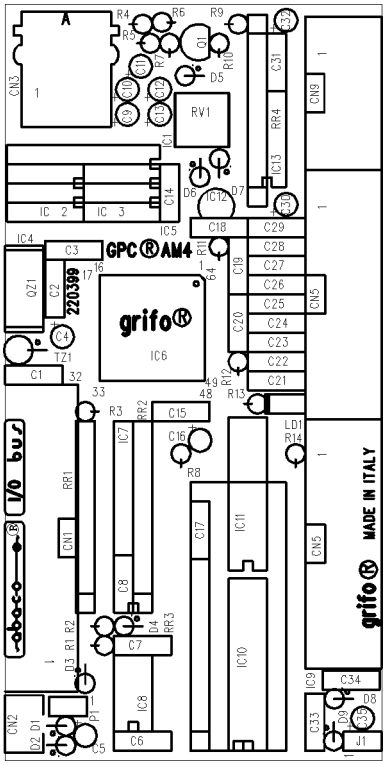
The watch dog circuit of **GPC® AM4** is really efficient and provided of easy software management. In details the most important features of this circuit are:

- 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 (about130 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.



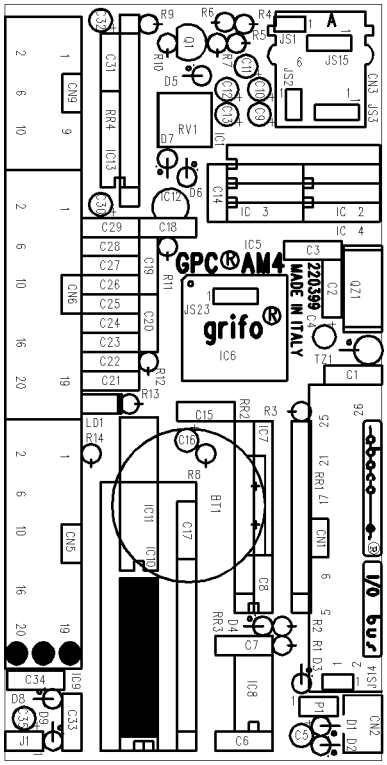


FIGURE 26: COMPONENTS MAPS (SOLDERING SIDE AND COMPONENTS SIDE)

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 powerfull 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 timeassembly code wich is optimised to run as fast as possible. The target AVR microcontroller therefore runs true assembly code rather than tokenised code wich is found in many other BASIC compilers. The upshot of this is that AVR BASIC code should be almost as fast assembler written code.

ICC AVR: cross compiler for C source program. It is a powerfull 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 ross 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 powerfull, 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 tolls 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.

DEVICES MAP AND ADDRESSES

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 adrresses 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 meaningfull.

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".

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

FIGURE 27: I/O ADDRESSES TABLE

The previous table shows only the registers of the peripheral devices xternal 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:

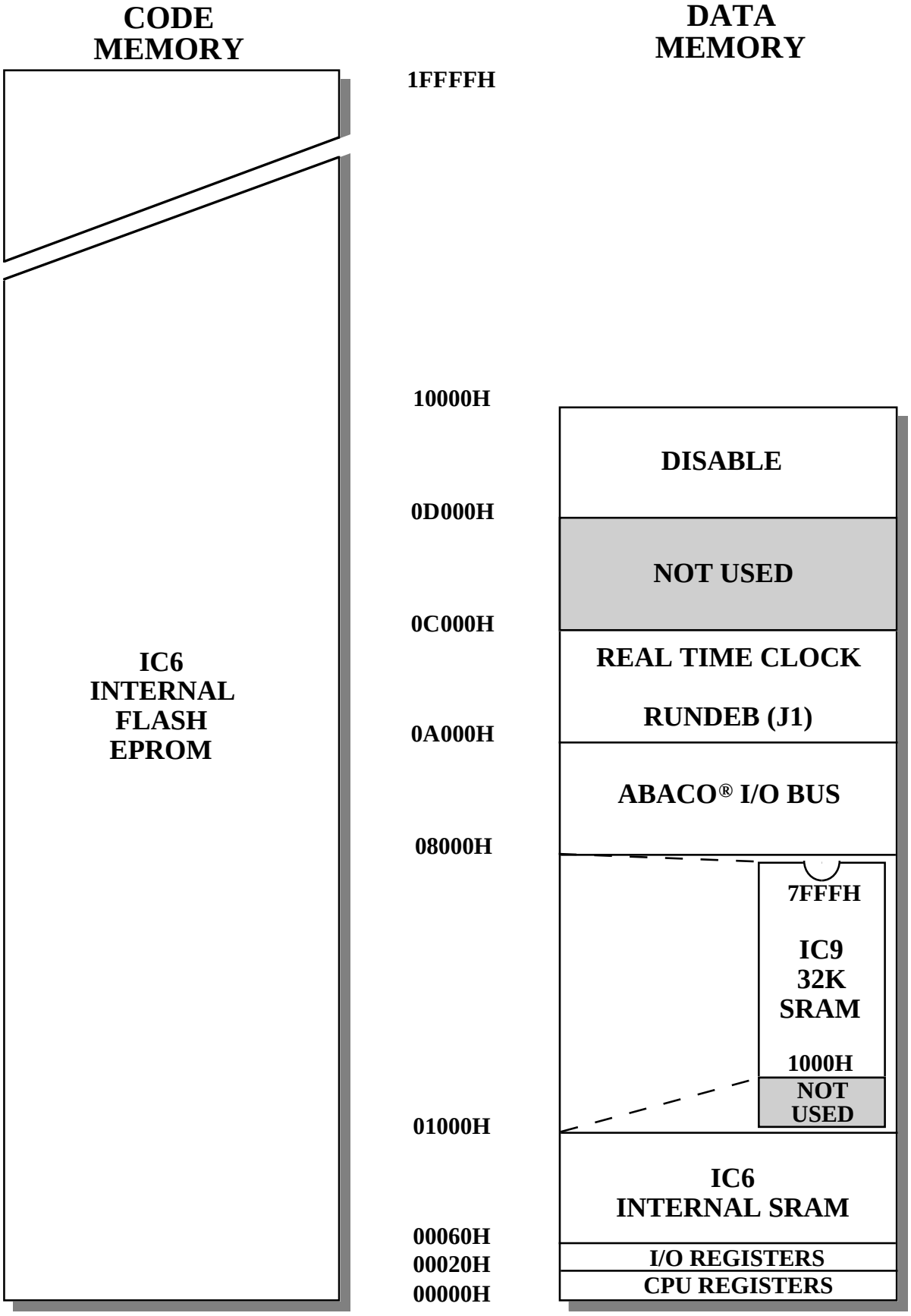


FIGURE 28: MEMORY ALLOCATION

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:

MEMORY	READ OPERATION	WRITE OPERATION
INTERNAL FLASH EPROM	ISP interface Direct CPU instructions	ISP interface
INTERNAL REGISTERS	Direct CPU instructions	Direct CPU instructions
INTERNAL SRAM	Direct CPU instructions	Direct CPU instructions
INTERNAL EEPROM	ISP interface Indirect CPU instructions	ISP interface Indirect CPU instructions
EXTERNAL SRAM	Direct CPU instructions	Direct CPU instructions
EXTERNAL I/O REGISTERS	Direct CPU instructions	Direct CPU instructions

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.

PERIPHERAL DEVICES SOFTWARE DESCRIPTION

In the previous paragraphes 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 decriptions 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 theRUN (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 peripherial 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 secondi	- 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
HOU10	- Decines of hours	- 2 least significant bits of HOU10.1÷HOU10.0 The third bit of HOU10.2 indicates AM/PM
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:

WEE.2	WEE.1	WEE.0	Day of the week
0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

While, the meaning of the three control registers is:

bit 7 6 5 4 3 2 1 0

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

Where:

NU = Not used.

30S = If high (1) itallowes 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 = Indictes 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.

bit 7 6 5 4 3 2 1 0

REG E = NU NU NU NU T1 T0 I M

Where:

NU = Not used.

T1 T0 = Determin the duration of the internal counters interrupt cycle.

T1	T0		
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 mask the interrupt status:

1 -> interrupt masked: the RTC interrupt signal is always disabled;

0 -> interrupt not masked: the RTC interrupt signal reflects interrupt status.

bit 7 6 5 4 3 2 1 0

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 mantains 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 mantain 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 on board 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 E^2 for set up and message. Pssibility 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. Indipendent 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 Comunication

Conversion card for serial communication, 2 RS 232 lines; 1 RS 422-485 line; 1 optical fibre line; selecatble 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.

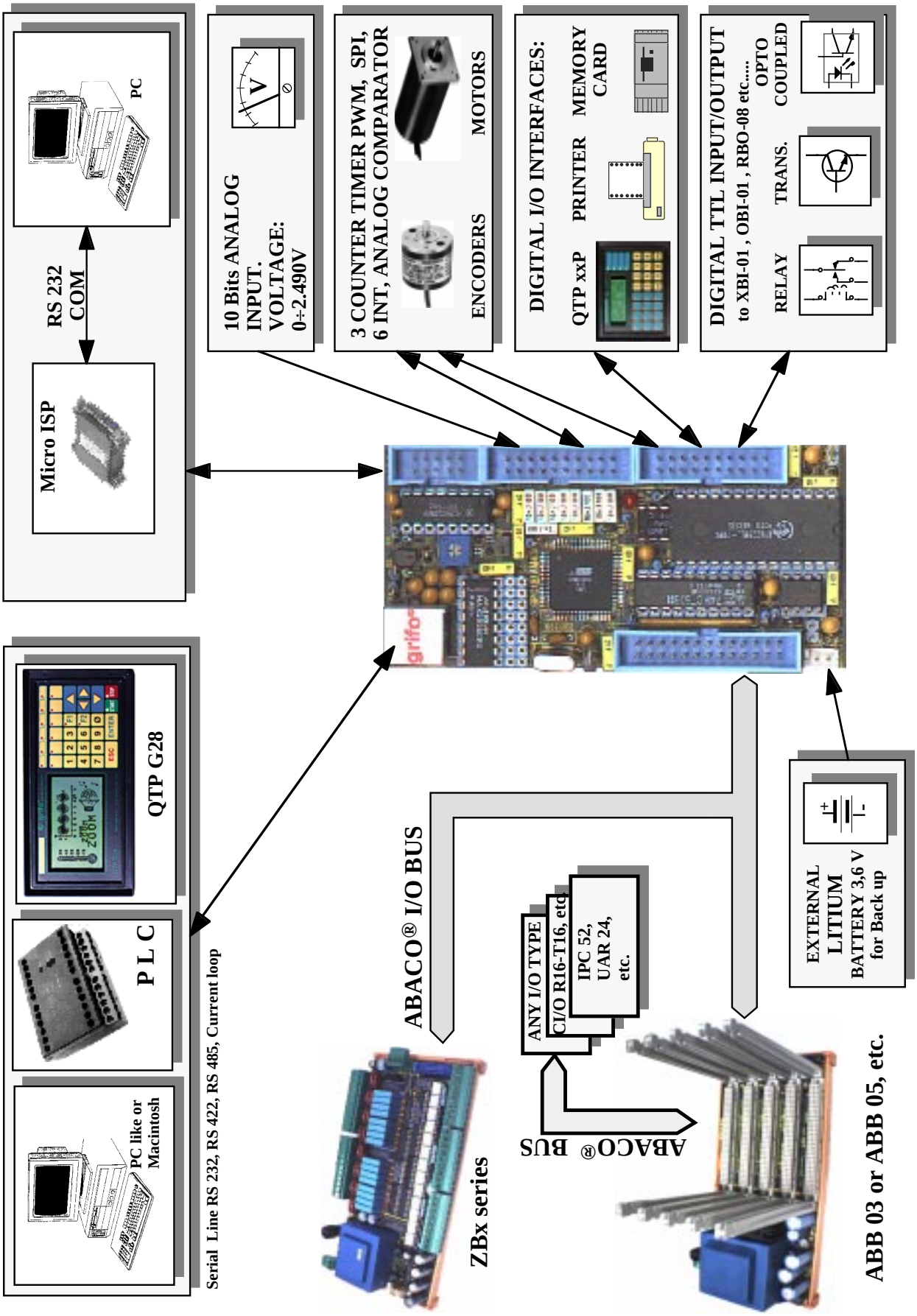


FIGURE 30: AVAILABLE CONNECTIONS DIAGRAM

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; 4 type dimension.

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 Ctype 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.

BIBLIOGRAPHY

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 247277-1 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.Elll** 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 immages that illustrate the connection modes.

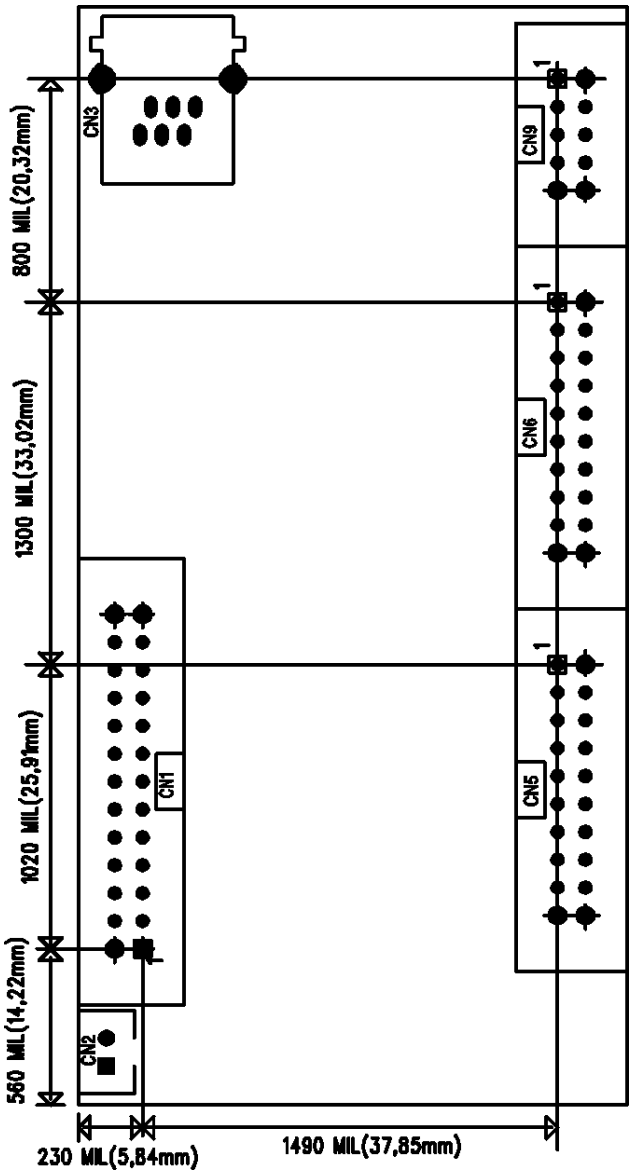
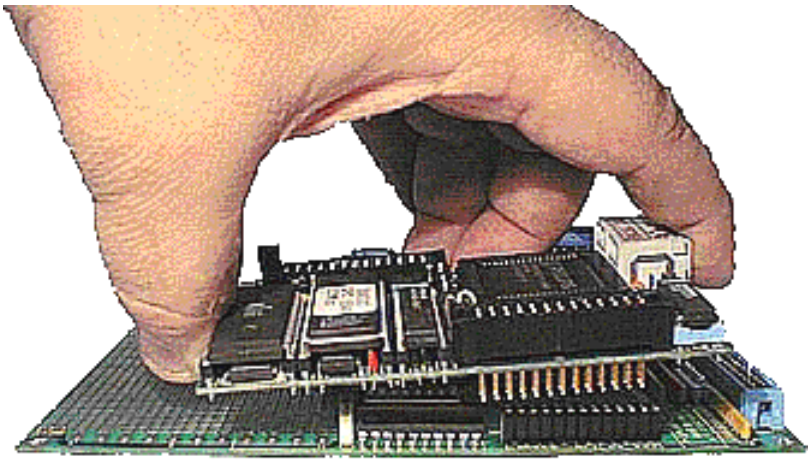


FIGURE A1: MODULE DIMENSION FOR PIGGY BACK MOUNTING



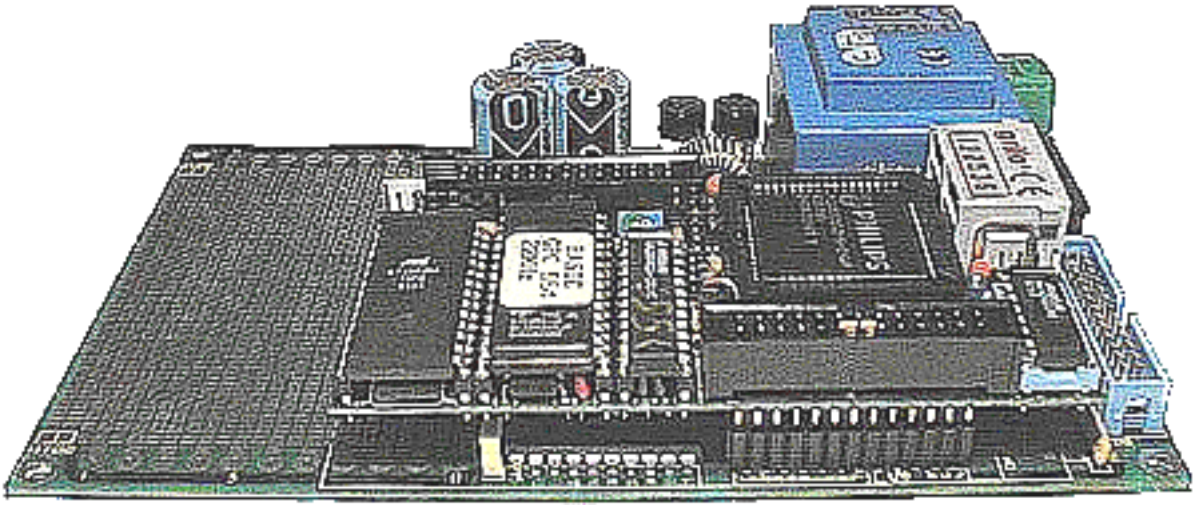


FIGURE A2: PIGGY BACK MOUNTING

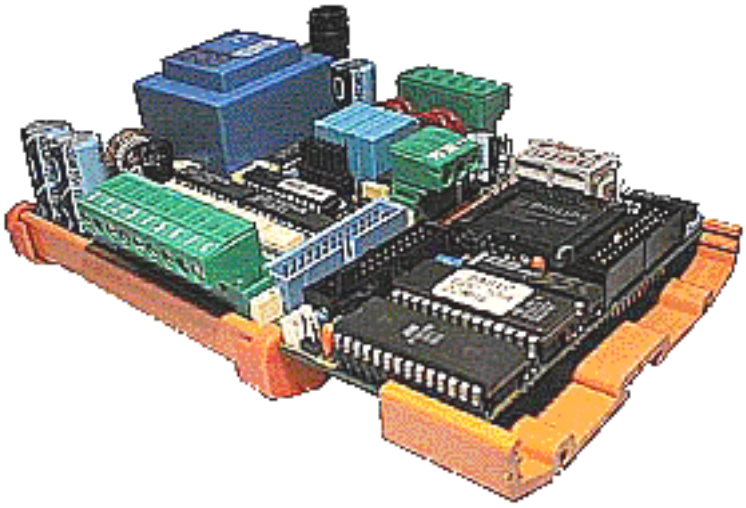


FIGURE A3: WEIDMULLER RAIL MOUNTING

Features

- Utilizes the AVR® RISC Architecture
- AVR - High-performance and Low-power RISC Architecture
  - 120/121 Powerful Instructions - Most Single Clock Cycle Execution
  - 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)

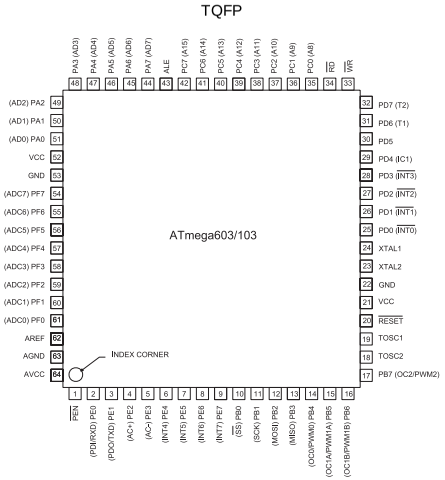
8-bit AVR® Microcontroller with 64K/128K Bytes In-Systen Programmable Flash

ATmega603
ATmega603L
ATmega103
ATmega103L

Preliminary

Rev. 0945C–04.

Pin Configuration



Description

The ATmega603/103 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executi powerful instructions in a single clock cycle, the ATmega603/103 achieves throughputs approaching 1 MIPS per MI allowing the system designer to optimize power consumption versus processing speed.

The AVR core is based on an enhanced RISC architecture that combines a rich instruction set with 32 general purpo working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independe registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more co efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

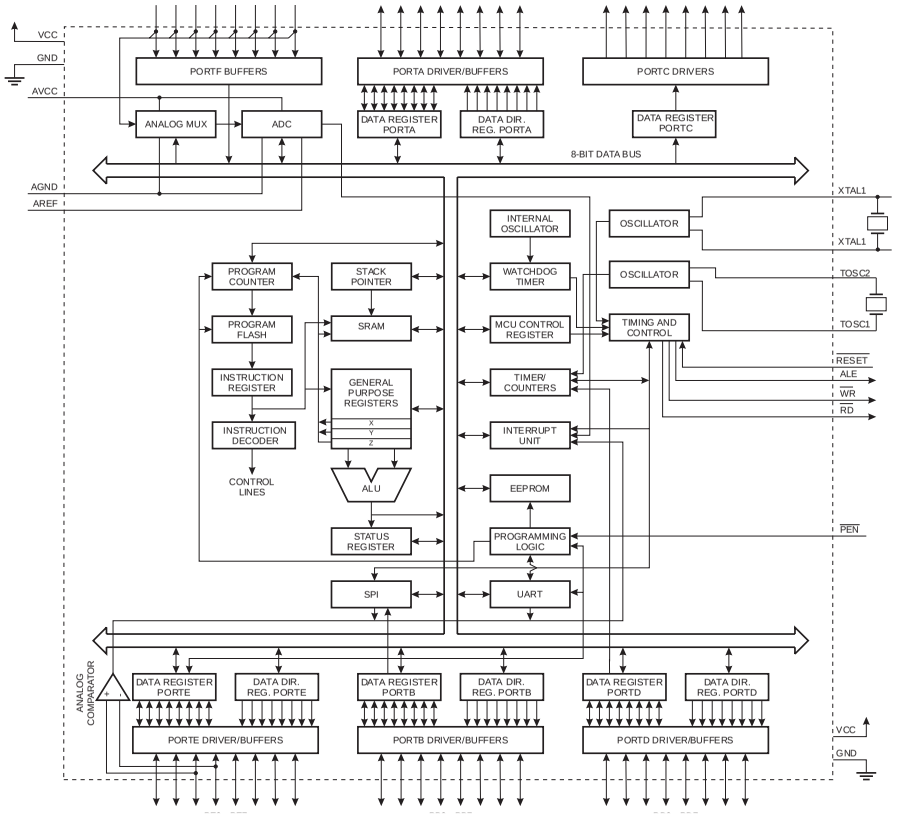
The ATmega603/103 provides the following features: 64K/128K bytes of In-system Programmable Flash, 2K/4K byt EEPROM, 4K bytes SRAM, 32 general purpose I/O lines, 8 Input lines, 8 Output lines, 32 general purpose working reg ters, Real Time Counter (RTC), 4 flexible timer/counters with compare modes and PWM, UART, programmable Watchd Timer with internal oscillator, an SPI serial port and three software selectable power saving modes. The Idle Mode sto the CPU while allowing the SRAM, timer/counters, SPI port and interrupt system to continue functioning. The Power Dov mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or har ware reset. In Power Save mode, the timer oscillator continues to run, allowing the user to maintain a timer base while t rest of the device is sleeping.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The on-chip ISP Flash allows t program memory to be reprogrammed in-system through a serial interface or by a conventional nonvolatile memory pr grammer. By combining an 8-bit RISC CPU with a large array of ISP Flash on a monolithic chip, the Atmel ATmega603/1 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded contr applications.

The ATmega603/103 AVR is supported with a full suite of program and system development tools including: C compile macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

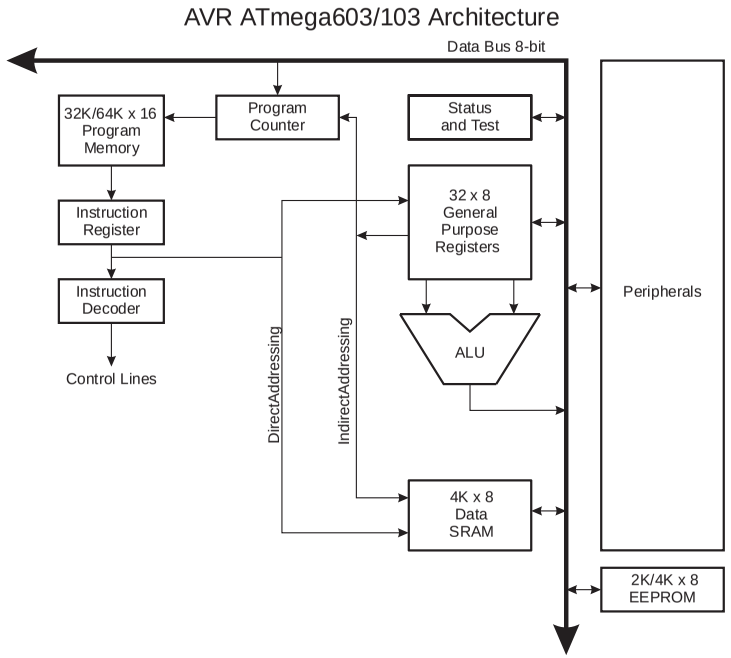
Block Diagram

Figure 1. The ATmega603/103 Block Diagram



Architectural Overview

Figure 4. The ATmega603/103 AVR RISC Architecture



The AVR uses a Harvard architecture concept - with separate memories and buses for program and data. The progra memory is accesses with a single level pipeline. While one instruction is being executed, the next instruction is pre-fetche 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 forma 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 allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size ar the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts a 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 AV architecture.

abacus bus grifo® ITALIAN TECHNOLOGY

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the stat register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of tl program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower tl 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

	7 … 0	Addr.	
	R0	$00	
	R1	$01	
	R2	$02	
	. . .		
	R13	$0D	
General	R14	$0E	
Purpose	R15	$0F	
Working	R16	$10	
Registers	R17	$11	
	. . .		
	R26	$1A	X-register low byte
	R27	$1B	X-register high byte
	R28	$1C	Y-register low byte
	R29	$1D	Y-register high byte
	R30	$1E	Z-register low byte
	R31	$1F	Z-register high byte

All the register operating instructions in the instruction set have direct and single cycle access to all registers. The or exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and register and the LDI instruction for load immediate constant data. These instructions apply to the second half of the regi ters in the register file - R16..R31. The general SBC, SUB, CP, AND and OR and all other operations between two registe or on a single register apply to the entire register file.

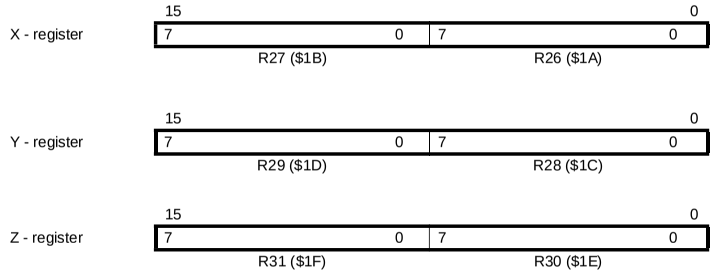
As shown in Figure 5, each register is also assigned a data memory address, mapping them directly into the first 3 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organizatic provides great flexibility in access of the registers, as the X,Y and Z registers can be set to index any register in the file.

The 4K bytes of SRAM available for general data are implemented as addresses $0060 to $0FFF.

X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are address pointe for indirect addressing of the SRAM. The three indirect address registers X, Y and Z are defined as:

Figure 6. X, Y and Z Registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment ar decrement (see the descriptions for the different instructions).

ALU - Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided in three main categories - arithmetic, logical and bit-functions.

ISP Flash Program Memory

The ATmega603/103 contains 64K/128K bytes on-chip In-system Programmable Flash memory for program storag Since all instructions are single or double 16-bit words, the Flash is organized as 64K x 16. The Flash memory has a endurance of at least 1000 write/erase cycles.

Constant tables can be allocated in the entire program memory space (see the LPM - Load Program Memory and ELP Extended Load Program Memory instruction descriptions).

SRAM Data Memory

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

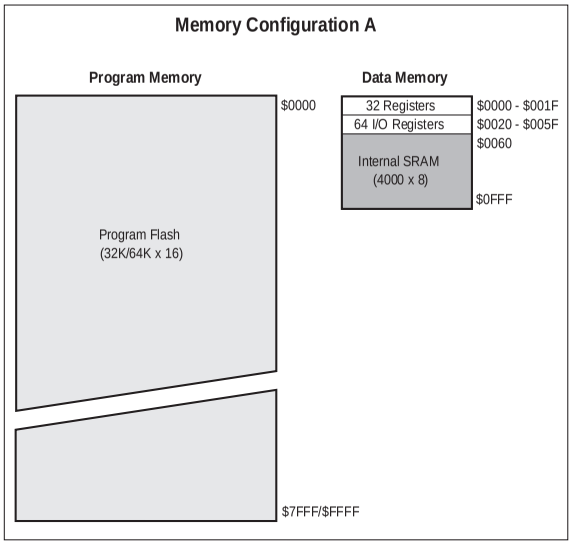
Table 2. Memory Configurations

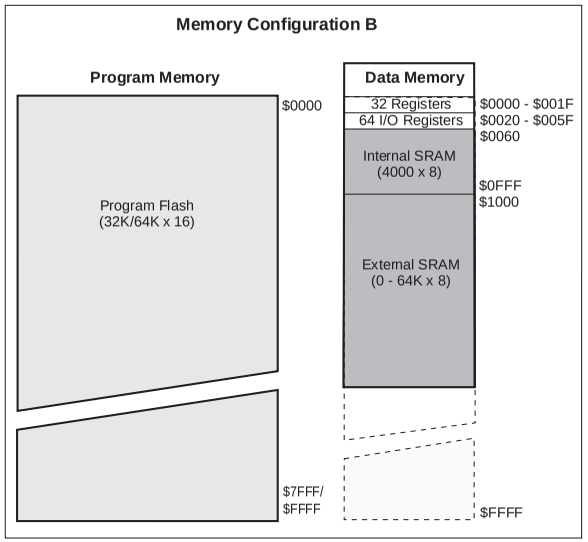
Configuration	Internal SRAM Data Memory	External SRAM Data Memory
A	4000	None
B	4000	up to 64K

Note: When using 64K of External SRAM, 60K will be available.

ITALIAN TECHNOLOGY grifo® abaco bus

Figure 7. Memory Configurations





The 4096 first Data Memory locations address both the Register file, the I/O Memory and the internal data SRAM. The fir 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 remainir address locations in the 64K address space. This area starts at the address following the internal SRAM. If a 64K extern 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 exceeds the internal data memory locations, the external da SRAM is accessed using the same instructions as for the internal data memory access. When the internal data memori are accessed, the read and write strobe pins ($\overline{RD}$ and $\overline{WR}$) are inactive during the whole access cycle. External SRA operation is enabled by setting the SRE bit in the MCUCR register.

Accessing external SRAM takes one additional clock cycle per byte compared to access of the internal SRAM. This mea that the commands LD, ST, LDS, STS, PUSH and POP take one additional clock cycle. If the stack is placed in extern SRAM, interrupts, subroutine calls and returns take two clock cycles extra because the two-byte program counter pushed and popped. When external SRAM interface is used with wait state, two additional clock cycles is used per byt This has the following effect: Data transfer instructions take two extra clock cycles, whereas interrupt, subroutine calls a 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 wi Pre-Decrement and Indirect with Post-Increment. In the register file, registers R26 to R31 feature the indirect addressi pointer registers.

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

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

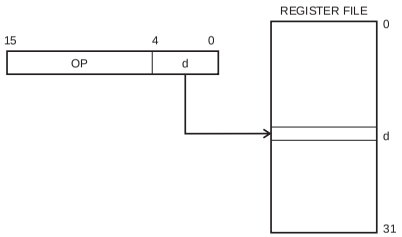
The entire data address space including the 32 general purpose working registers and the 64 I/O registers are all acces ble through all these addressing modes. See the next section for a detailed description of the different addressing modes

Program and Data Addressing Modes

The ATmega603/103 AVR RISC microcontroller supports powerful and efficient addressing modes for access to th program memory (Flash) and data memory (SRAM, Register File and I/O Memory). This section describes the differe addressing modes supported by the AVR architecture. In the figures, OP means the operation code part of the instructi 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

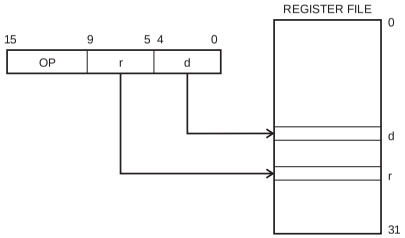


The operand is contained in register d (Rd).

ITALIAN TECHNOLOGY

Register Direct, Two Registers Rd and Rr

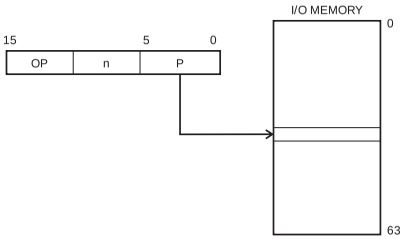
Figure 9. Direct Register Addressing, Two Registers



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

I/O Direct

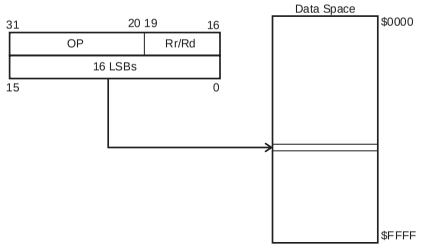
Figure 10. I/O Direct Addressing



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

Data Direct

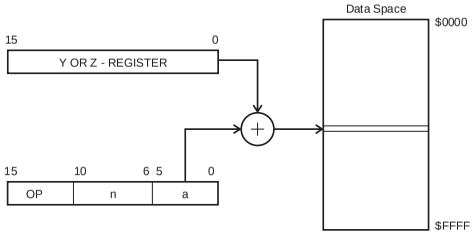
Figure 11. Direct Data Addressing



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

Data Indirect with Displacement

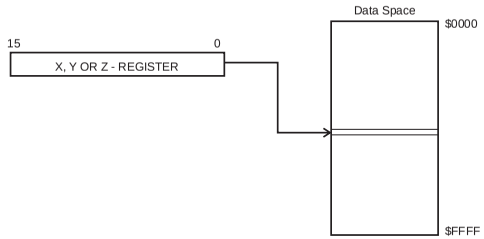
Figure 12. Data Indirect with Displacement



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

Data Indirect

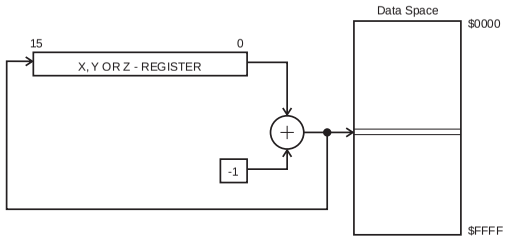
Figure 13. Data Indirect Addressing



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

Data Indirect With Pre-Decrement

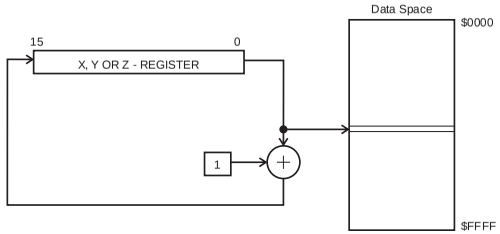
Figure 14. Data Indirect Addressing with Pre-Decrement



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

Data Indirect With Post-Increment

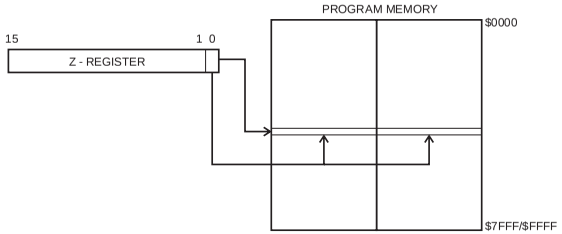
Figure 15. Data Indirect Addressing with Post-Increment



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

Constant Addressing Using the LPM and ELPM Instructions

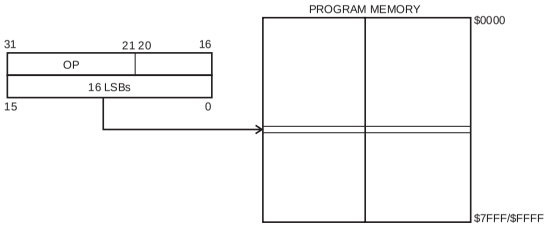
Figure 16. Code Memory Constant Addressing



Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 32K), LSB selects lo 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 select low or high memory page (RAMPZ0 = 0: Low Page, RAMPZ0 = 1: High Page). ELPM does not apply to th ATmega603.

Direct Program Address, JMP and CALL

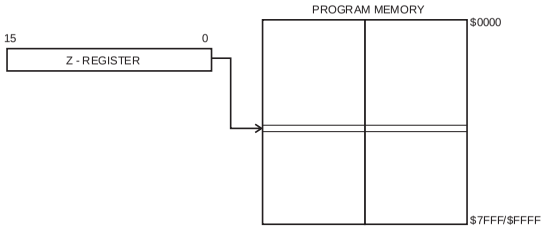
Figure 17. Direct Program Memory Addressing



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

Indirect Program Addressing, IJMP and ICALL

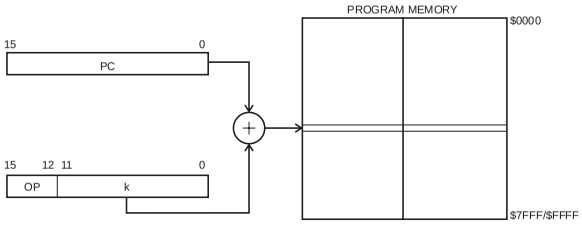
Figure 18. Indirect Program Memory Addressing



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

Relative Program Addressing, RJMP and RCALL

Figure 19. Relative Program Memory Addressing



Program execution continues at address PC + k + 1. The relative address k is -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. Tl EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU described on page 52 specifying the EEPROM address register, the EEPROM data register, and the EEPROM contr 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 clock crystal for the chip. No interr clock division is used.

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

Figure 20. The Parallel Instruction Fetches and Instruction Executions

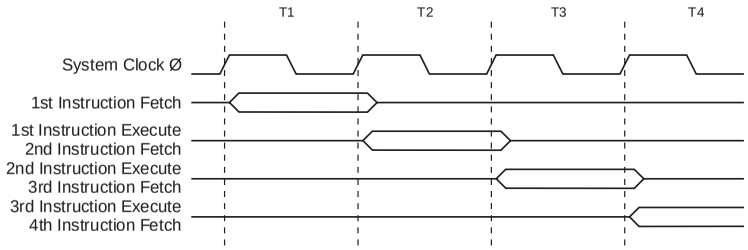


Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two regist operands is executed, and the result is stored back to the destination register.

I/O Memory

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

Table 3. ATmega603/103 I/O Space

I/O Address (SRAM Address)	Name	Function
$3F ($5F)	SREG	Status REGister
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3C ($5C)	XDIV	XTAL Divide Control Register
$3B ($5B)	RAMPZ	RAM Page Z Select Register
$3A ($5A)	EICR	External Interrupt Control Register
$39 ($59)	EIMSK	External Interrupt MaSK register
$38 ($58)	EIFR	External Interrupt Flag Register
$37 ($57)	TIMSK	Timer/Counter Interrupt MaSK register
$36 ($56)	TIFR	Timer/Counter Interrupt Flag register
$35 ($55)	MCUCR	MCU General Control Register
$34 ($54)	MCUSR	MCU Status Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register
$32 ($52)	TCNT0	Timer/Counter0 (8-bit)
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register
$30 ($50)	ASSR	Asynchronous Mode Status Register
$2F ($4F)	TCCR1A	Timer/Counter1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter1 Control Register B
$2D ($4D)	TCNT1H	Timer/Counter1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter1 Low Byte
$2B ($4B)	OCR1AH	Timer/Counter1 Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Timer/Counter1 Output Compare Register A Low Byte
$29 ($49)	OCR1BH	Timer/Counter1 Output Compare Register B High Byte
$28 ($48)	OCR1BL	Timer/Counter1 Output Compare Register B Low Byte
$27 ($47)	ICR1H	Timer/Counter1 Input Capture Register High Byte
$26 ($46)	ICR1L	Timer/Counter1 Input Capture Register Low Byte
$25 ($45)	TCCR2	Timer/Counter2 Control Register
$24 ($44)	TCNT2	Timer/Counter2 (8-bit)
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register
$21 ($41)	WDTCR	Watchdog Timer Control Register
$1F ($3F)	EEARH	EEPROM Address Register High
$1E ($3E)	EEARL	EERPOM Address Register Low
$1D ($3D)	EEDR	EEPROM Data Register
$1C ($3C)	EECR	EEPROM Control Register

Table 3. ATmega603/103 I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$1B ($3B)	PORTA	Data Register, Port A
$1A ($3A)	DDRA	Data Direction Register, Port A
$19 ($39)	PINA	Input Pins, Port A
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B
$15 ($35)	PORTC	Data Register, Port C
$12 ($32)	PORTD	Data Register, Port D
$11 ($31)	DDRD	Data Direction Register, Port D
$10 ($30)	PIND	Input Pins, Port D
$0F ($2F)	SPDR	SPI I/O Data Register
$0E ($2E)	SPSR	SPI Status Register
$0D ($2D)	SPCR	SPI Control Register
$0C ($2C)	UDR	UART I/O Data Register
$0B ($2B)	USR	UART Status Register
$0A ($2A)	UCR	UART Control Register
$09 ($29)	UBRR	UART Baud Rate Register
$08 ($28)	ACSR	Analog Comparator Control and Status Register
$07 ($27)	ADMUX	ADC Multiplexer Select Register
$06 ($26)	ADCSR	ADC Control and Status Register
$05 ($25)	ADCH	ADC Data Register High
$04 ($24)	ADCL	ADC Data Register Low
$03 ($23)	PORTE	Data Register, Port E
$02 ($22)	DDRE	Data Direction Register, Port E
$01 ($21)	PINE	Input Pins, Port E
$00 ($20)	PINF	Input Pins, Port F

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 different I/O locations are direc 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 instructi set chapter for more details. When using the I/O specific instructions IN, OUT, the I/O register address $00 - $3F are use When addressing I/O registers as SRAM, $20 must be added to this address. All I/O register addresses throughout th 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 address 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 instructio work with registers $00 to $1F only.

grifo® ITALIAN TECHNOLOGY

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:

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	**I**	**T**	**H**	**S**	**V**	**N**	**Z**	**C**	**SREG**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **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 then performed in separate control registers. If the global interrupt enable register is cleared (zero), none of the interrup are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt h occurred, 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 from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in register in the register file by the BLD instruction.

- **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 detail 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 Instru tion 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 f 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 Descripti 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 f 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 detail information.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning fro 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) a $3D ($5D). As the ATmega603/103 supports up to 64 kB memory, all 16-bits are used.

Bit	15	14	13	12	11	10	9	8	
$3E ($5E)	**SP15**	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
$3D ($5D)	**SP7**	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	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 Sta space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts a enabled. The stack pointer must be set to point above $60. The Stack Pointer is decremented by one when data is push onto the Stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack w subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the Stack with the PO 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

Bit	7	6	5	4	3	2	1	0	
$3B ($5B)	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**RAMPZ0**	**RAMPZ**
Read/Write	R	R	R	R	R	R	R	R/W	
Initial value	0	0	0	0	0	0	0	0	

The RAMPZ register is normally used to select which 64K RAM Page is accessed by the Z pointer. As the ATmega603/1 does not support more than 64K of SRAM memory, this register is used only to select which page in the program memo 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.

The ATmega603 does not contain the RAMPZ register, and it does not have the ELPM instruction. The ordinary LF instruction can reach the entire program memory in the ATmega603.

MCU Control Register - MCUCR

The MCU Control Register contains control bits for general MCU functions.

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	**SRE**	**SRW**	**SE**	**SM1**	**SM0**	**-**	**-**	**-**	**MCUCR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
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 AD0-7 (Port A), and A8-15 (P C) are activated as the alternate pin functions. Then the SRE bit overrides any pin direction settings in the respective da direction registers. When the SRE bit is cleared (zero), the external data SRAM is disabled, and the normal pin and da direction settings are used.

- **Bit 6 - SRW: External SRAM Wait State**

When the SRW bit is set (one), a one cycle wait state is inserted in the external data SRAM access cycle. When the SR 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 states73 and Figure 52.External SRAM Access Cycle with wait state74.

- **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 tl MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit ju before the execution of the SLEEP instruction.

- **Bits 4,3 - SM1/SM0: Sleep Mode Select bits 1 and 0**

This bit selects between the three available sleep modes as shown in the following table:

Table 4. Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle Mode
0	1	Reserved
1	0	Power Down
1	1	Power Save

- **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 fe ture can be used to decrease power consumption when the requirement for processing power is low.

Bit	7	6	5	4	3	2	1	0	
$3C ($5C)	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	XDIV
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - XDIVEN: XTAL Divide Enable**

When the XDIVEN bit is set (one), the clock frequency of the CPU and all peripherals is divided by the factor defined by tl setting of XDIV6 - XDIV0. This bit can be set and cleared run-time to vary the clock frequency as suitable to the applicatic

- **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). If the value of these bits is denoted the following formula defines the resulting CPU clock frequency f_{clk}:

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

The value of these bits can only be changed when XDIVEN is zero. When XDIVEN is set to one, the value written simult neously into XDIV6..XDIV0 is taken as the division factor. When XDIVEN is cleared to zero, the value writt 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.

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 s (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. Tl complete list of vectors is shown in Table 5. The list also determines the priority levels of the different interrupts. The low the address the higher is the priority level. RESET has the highest priority, and next is INT0 - the External Interrupt Reque 0 etc.

Table 5. Reset and Interrupt Vectors

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

The most typical program setup for the Reset and Interrupt Vector Addresses are:

```
Address  Labels   Code                      Comments
$0000             jmp    RESET            ; Reset Handler
$0002             jmp    EXT_INT0         ; IRQ0 Handler
$0004             jmp    EXT_INT1         ; IRQ1 Handler
$0006             jmp    EXT_INT2         ; IRQ2 Handler
$0008             jmp    EXT_INT3         ; IRQ3 Handler
$000A             jmp    EXT_INT4         ; IRQ4 Handler
$000C             jmp    EXT_INT5         ; IRQ5 Handler
$000E             jmp    EXT_INT6         ; IRQ6 Handler
$0010             jmp    EXT_INT7         ; IRQ7 Handler
$0012             jmp    TIM2_COMP        ; Timer2 Compare Handler
$0014             jmp    TIM2_OVF         ; Timer2 Overflow Handler
$0016             jmp    TIM1_CAPT        ; Timer1 Capture Handler
$0018             jmp    TIM1_COMPA       ; Timer1 CompareA Handler
$001A             jmp    TIM1_COMPB       ; Timer1 CompareB Handler
$001C             jmp    TIM1_OVF         ; Timer1 Overflow Handler
$001E             jmp    TIM0_COMP        ; Timer0 Compare Handler
$0020             jmp    TIM0_OVF         ; Timer0 Overflow Handler
$0022             jmp    SPI_STC          ; SPI Transfer Complete Handler
$0024             jmp    UART_RXC         ; UART RX Complete Handler
$0026             jmp    UART_DRE         ; UDR Empty Handler
$0028             jmp    UART_TXC         ; UART TX Complete Handler
$002A             jmp    ADC              ; ADC Conversion Complete Handler
$002C             jmp    EE_RDY           ; EEPROM Ready Handler
$002E             jmp    ANA_COMP         ; Analog Comparator Handler
;
$0030    MAIN:    ldi    r16, high(RAMEND); Main program start
$0031             out    SPH,r16
$0032             ldi    r16, low(RAMEND)
$0033             out    SPL,r16
$0034             <instr>  xxx
  ...     ...       ...    ...
```

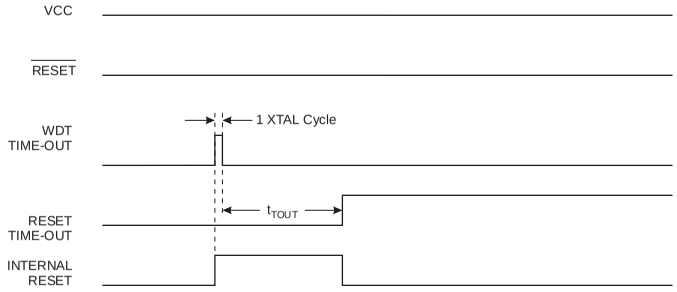
Reset Sources

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 $\overline{RESET}$ 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 ex cution from address $0000. The instruction placed in address $0000 must be a JMP - absolute jump instruction to the res handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular progra code can be placed at these locations. The circuit diagram in Figure 23 shows the reset logic. Table 6 defines the timi and electrical parameters of the reset circuitry.

Figure 27. Watchdog Reset During Operation



MCU Status Register - MCUSR

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

Bit	7	6	5	4	3	2	1	0	
$34 ($54)	-	-	-	-	-	-	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	See bit description		

- **Bits 7..2 - Res: Reserved Bits**

These bits are reserved bits in the ATmega603/103 and always read as zero.

- **Bit 1 - EXTRF: External 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 unchanged.

- **Bit 0 - 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

Reset Source	EXTRF	PORF
Power-on Reset	undefined	1
External Reset	1	unchanged
Watchdog Reset	unchanged	unchanged

To make use of these bits to identify a reset condition, the user software should clear both the PORF and EXTRF bits early as possible in the program. Checking the PORF and EXTRF values is done before the bits are 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

EXTRF	PORF	Reset Source
0	0	Watchdog Reset
0	1	Power-on Reset
1	0	External Reset
1	1	Power-on Reset

Interrupt Handling

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

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

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

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be s 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 interru flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priori

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition active.

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

External Interrupt Mask Register - EIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..4 - INT7 - INT4: External Interrupt Request 7-4 Enable**

When an INT7- INT4 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the corresponding external p interrupt is enabled. The Interrupt Sense Control bits in the External Interrupt Control Register - EICR defines whether t external interrupt is activated on rising or falling edge or level sensed. Activity on any of these pins will trigger an interru request even if the pin is enabled as an output. This provides a way of generating a software interrupt.

- **Bits 3..0 - INT3 - INT0: External Interrupt Request 3-0 Enable**

When an INT3 - INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the corresponding external p interrupt is enabled. The external interrupts are always low level triggered interrupts. Activity on any of these pins will tr ger an interrupt request even if the pin is enabled as an output. This provides a way of generating a software interru When enabled, a level triggered interrupt will generate an interrupt request as long as the pin is held low.

External Interrupt Flag Register - EIFR

Bit	7	6	5	4	3	2	1	0	
$38 ($58)	INTF7	INTF6	INTF5	INTF4	-	-	-	-	EIFR
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..4 - INTF7 - INTF4: External Interrupt 7-4 Flags**

When an event on the INT7 - INT4 pins triggers an interrupt request, the corresponding interrupt flag, INTF7 - INTI becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit, INT7 - INT4 in EIMSK, are set (one), t MCU will jump to the interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag cleared by writing a logical one to it.

- **Bits 3..0 - Res: Reserved Bits**

These bits are reserved bits in the ATmega603/103 and always read as zero.

External Interrupt Control Register - EICR

Bit	7	6	5	4	3	2	1	0	
$3A ($5A)	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	EICR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..0 - ISCX1, ISCX0: External Interrupt 7-4 Sense Control bits**

The External Interrupts 7 - 4 are activated by the external pins INT7 - INT4 if the SREG I-flag and the corresponding inte rupt mask in the EIMSK is set. The level and edges on the external pins that activate the interrupts are defined in t following table:

Table 10. Interrupt Sense Control

ISCX1	ISCX0	Description
0	0	The low level of INTX generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INTX generates an interrupt request.
1	1	The rising edge of INTX generates an interrupt request.

Note: X = 7, 6, 5 or 4.
When changing the ISCX1/ISCX0 bits, the interrupt must be disabled by clearing its Interrupt Enable bit in the GIMSK Registe Otherwise an interrupt can occur when the bits are changed.

The value on the INTX pin is sampled before detecting edges. If edge interrupt is selected, pulses that last longer than o CPU clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. Observe that CP clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. If low level interrupt is selected, t low level must be held until the completion of the currently executing instruction to generate an interrupt. If enabled, a lev triggered interrupt will generate an interrupt request as long as the pin is held low.

Timer/Counter Interrupt Mask Register - TIMSK

Bit	7	6	5	4	3	2	1	0	
$37 ($57)	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - OCIE2: Timer/Counter2 Output Compare Interrupt Enable**

When the OCIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match inte rupt is enabled. The corresponding interrupt (at vector $0012) is executed if a Compare match in Timer/Counter2 occu i.e., when the OCF2 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 6 - TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt enabled. The corresponding interrupt (at vector $0014) is executed if an overflow in Timer/Counter2 occurs, i.e., when t TOV2 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 5 - TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Eve Interrupt is enabled. The corresponding interrupt (at vector $0016) is executed if a capture-triggering event occurs on p 29, PD4(IC1), i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 4 - OCE1A: Timer/Counter1 Output CompareA Match Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareA Mat interrupt is enabled. The corresponding interrupt (at vector $0018) is executed if a CompareA match in Timer/Counte occurs, i.e., when the OCF1A bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

abaco bus grifo® ITALIAN TECHNOLOGY

- **Bit 3 - OCIE1B: Timer/Counter1 Output CompareB Match Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareB Mat interrupt is enabled. The corresponding interrupt (at vector $001A) is executed if a CompareB match in Timer/Counte 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 enabled. The corresponding interrupt (at vector $001C) is executed if an overflow in Timer/Counter1 occurs, i.e., when tl TOV1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 1 - OCIE0: Timer/Counter0 Output Compare Interrupt Enable**

When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match inte rupt is enabled. The corresponding interrupt (at vector $001E) is executed if a Compare match in Timer/Counter0 occu i.e., when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 0 - TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt enabled. The corresponding interrupt (at vector $0020) is executed if an overflow in Timer/Counter0 occurs, i.e., when tl TOV0 bit is set in the Timer/Counter Interrupt Flag Register - TIFR

Timer/Counter Interrupt Flag Register - TIFR

Bit	7	6	5	4	3	2	1	0	
$36 ($56)	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - OCF2: Output Compare Flag 2:**

The OCF2 bit is set (one) when compare match occurs between Timer/Counter2 and the data in OCR2 - Output Compa Register 2. OCF2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCI is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE2 (Timer/Counter2 Compare Interru Enable), and the OCF2 are set (one), the Timer/Counter2 Output Compare Interrupt is executed.

- **Bit 6 - TOV2: Timer/Counter2 Overflow Flag**

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by hardware when executing tl corresponding interrupt handling vector. Alternatively, TOV2 is cleared by writing a logic one to the flag. When the I-bit SREG, and TOIE2 (Timer/Counter1 Overflow Interrupt Enable), and TOV2 are set (one), the Timer/Counter2 Overflo Interrupt is executed. In PWM mode, this bit is set when Timer/Counter2 advances from $00.

- **Bit 5 - ICF1: Input Capture Flag 1**

The ICF1 bit is set (one) to flag an input capture event, indicating that the Timer/Counter1 value has been transferred to tl input capture register - ICR1. ICF1 is cleared by hardware when executing the corresponding interrupt handling vect Alternatively, ICF1 is cleared by writing a logic one to the flag. When the SREG I-bit, and TICIE1 (Timer/Counter1 Inp Capture Interrupt Enable), and ICF1 are set (one), the Timer/Counter1 Capture Interrupt is executed.

- **Bit 4 - OCF1A: Output Compare Flag 1A**

The OCF1A bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1A - Outp Compare Register 1A. OCF1A is cleared by hardware when executing the corresponding interrupt handling vector. Altern tively, OCF1A is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE1A (Timer/Counter1 Compa Interrupt Enable), and the OCF1A are set (one), the Timer/Counter1 Compare A match Interrupt is executed.

- **Bit 3 - OCF1B: Output Compare Flag 1B**

The OCF1B bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1B - Outp Compare Register 1B. OCF1B is cleared by hardware when executing the corresponding interrupt handling vector. Altern tively, OCF1B is cleared by writing a logic one to the flag.. When the I-bit in SREG, and OCIE1B (Timer/Counter1 Compa match Interrupt Enable), and the OCF1B are set (one), the Timer/Counter1 Compare B match Interrupt is executed.

- **Bit 2 - TOV1: Timer/Counter1 Overflow Flag**

The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the co responding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logic one to the flag. When the I-bit SREG, and TOIE1 (Timer/Counter1 Overflow Interrupt Enable), and TOV1 are set (one), the Timer/Counter1 Overflo Interrupt is executed. In PWM mode, this bit is set when Timer/Counter1 advances from $0000.

- **Bit 1 - OCF0: Output Compare Flag 0**

The OCF0 bit is set (one) when compare match occurs between Timer/Counter0 and the data in OCR0 - Output Compa Register 0. OCF0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCI is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE0 (Timer/Counter2 Compare Interru Enable), and the OCF0 are set (one), the Timer/Counter0 Output Compare Interrupt is executed.

- **Bit 0 - TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing tl corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interru is executed. In PWM mode, this bit is set when Timer/Counter0 advances from $00.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. 4 clock cycles after tl interrupt flag 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 by The vector is normally a jump to the interrupt routine, and this jump takes 3 clock cycles. If an interrupt occurs during ex cution 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 clo cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2. When tl AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pendi 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 execute The SM1 and SM0 bits in the MCUCR register select which sleep mode (Idle, Power Down, or Power Save) will be ac vated by the SLEEP instruction, see Table 4.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine, a resumes execution from the instruction following SLEEP. The contents of the register file, SRAM, and I/O memory are una tered. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset vector

Idle Mode

When the SM1/SM0 bits are set to 00, the SLEEP instruction makes the MCU enter the Idle Mode, stopping the CPU b allowing SPI, UART, Analog Comparator, ADC, Timer/Counters, Watchdog and the interrupt system to continue operatin This enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow a UART Receive Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the analog compar tor can be powered down by setting the ACD-bit in the Analog Comparator Control and Status register - ACSR. This w reduce power consumption in Idle Mode. When the MCU wakes up from Idle mode, the CPU starts program executi 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 moc 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 f some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by tl watchdog oscillator clock, and if the input has the required level during this time, the MCU will wake up. The period of tl watchdog oscillator is 1 us (nominal) at 5.0V and 25C. The frequency of the watchdog oscillator is voltage dependent 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 becom effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined the same SUT fuses that define the reset time-out period. The wake-up period is equal to the clock reset period, as shov in Table 6.

If the wake-up condition disappears before the MCU wakes up and starts to execute, e.g. a low level on is not held lo 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 identic 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 ad tion to the Power Down wake-up sources, the device can also wake up from either Timer Overflow or Output Compa event from Timer/Counter0 if the corresponding Timer/Counter0 interrupt enable bits are set in TIMSK. To ensure that tl part executes the Interrupt routine when waking up, also set the global interrupt enable bit in SREG.

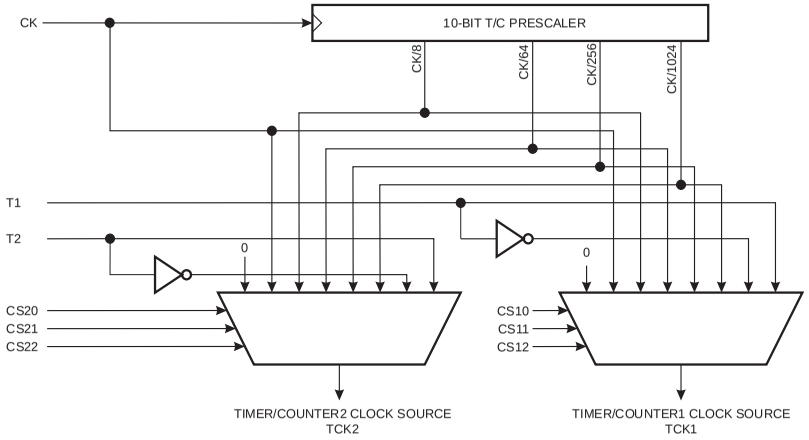
When waking up from Power Save Mode by an external interrupt, 2 instruction cycles are executed before the interru flags are updated. When waking up by the asynchronous timer, 3 instruction cycles are executed before the flags a updated. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and tl 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/Counte can optionally be asynchronously clocked from an external oscillator. This oscillator is optimized for use with a 32.768 k crystal, enabling use of Timer/Counter0 as a Real Time Clock (RTC). Timer/Counter0 has its own prescale Timer/Counters 1 and 2 have individual prescaling selection from the same 10-bit prescaling timer. These Timer/Counte can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which trigge the counting.

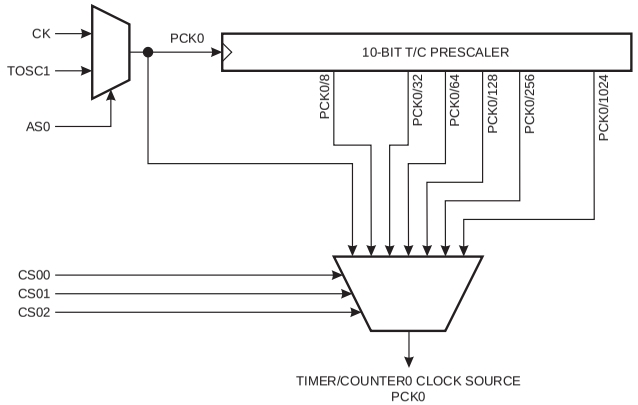
Timer/Counter Prescalers

Figure 28. Prescaler for Timer/Counter 1 and Timer/Counter2



For Timer/Counters 1 and 2, the four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024 where CK the CPU clock. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. F Timer/Counters 1 and 2, added selections as CK, external source and stop, can be selected as clock sources.

Figure 29. The Timer/Counter0 Prescaler



abaco bus grifo® ITALIAN TECHNOLOGY

The clock source for Timer/Counter0 prescaler is named PCK0. PCK0 is by default connected to the main system clo
CK. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. By setting t
AS0 bit in ASSR, Timer/Counter 0 prescaler is asynchronously clocked from the TOSC1 pin. This enables use
Timer/Counter0 as a Real Time Clock (RTC). A crystal can be connected between the TOSC1 and TOSC2 pins to serve
an independent clock source for Timer/Counter0. This oscillator is optimized for use with a 32.768 kHz crystal.

8-bit Timer/Counters T/C0 and T/C2

Figure 30 shows the block diagram for Timer/Counter0.

Figure 30. Timer/Counter0 Block Diagram

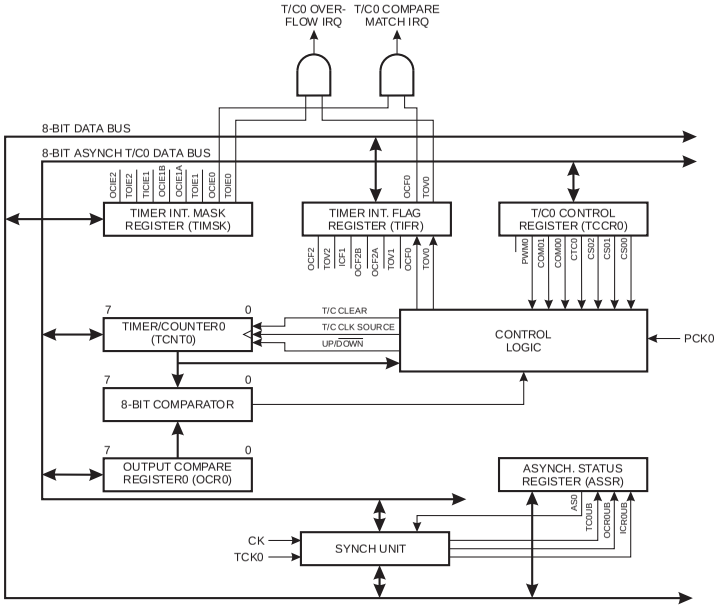
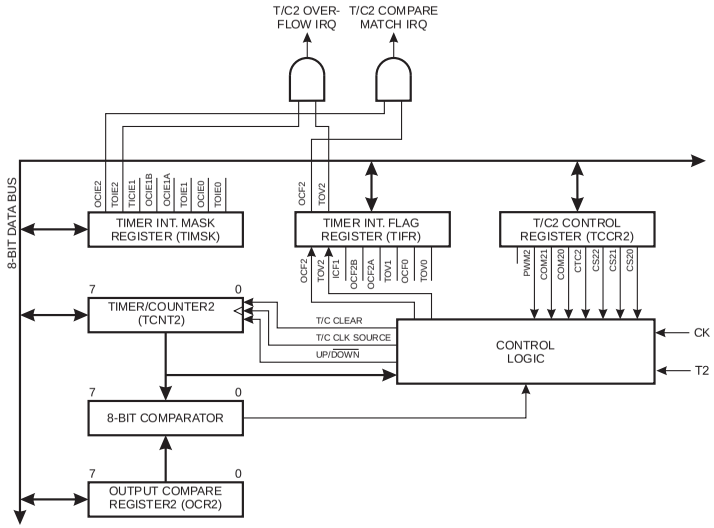


Figure 31. Timer/Counter2 Block Diagram



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 sele
clock source from CK, prescaled CK, or an external pin. Both Timer/Counters can be stopped as described in the specific
tion 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 Re
ister - TIFR. Control signals are found in the Timer/Counter Control Registers - TCCR0 and TCCR2. The interru
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.
assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least o
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. Si
ilarly, the high prescaling opportunities make these units useful for lower speed functions or exact timing functions w
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 functions include optional clearing
the counter on compare match, and action on the Output Compare Pins - PB4(OC0/PWM0) and PB7(OC2/PWM2) -
compare match.

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

Timer/Counter0 Control Register - TCCR0

Bit	7	6	5	4	3	2	1	0	
33 ($53)	-	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	TCCR0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Timer/Counter2 Control Register - TCCR2

Bit	7	6	5	4	3	2	1	0	
$25 ($45)	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	TCCR2
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **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 port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown Table 11.

Table 11. Compare Mode Select

COMn1	COMn0	Description
0	0	Timer/Counter disconnected from output pin OCn/PWMn
0	1	Toggle the OCn/PWMn output line.
1	0	Clear the OCn/PWMn output line (to zero).
1	1	Set the OCn/PWMn output line (to one).

Note: n = 0 or 2
In PWM mode, these bits have a different function. Refer to Table 14 for a detailed description.

- **Bit 3 - CTC0 / CTC2: Clear Timer/Counter on Compare match**

When the CTC0 or CTC2 control bit is set (one), the Timer/Counter is reset to $00 in the CPU clock cycle after a compa match. If the control bit is cleared, the Timer continues counting and is unaffected by a compare match. Since the compa match is detected in the CPU clock cycle following the match, this function will behave differently when a prescaling higl than 1 is used for the timer. When a prescaling of 1 is used, and the compare register is set to C, the timer will count as fi lows if CTC0/2 is set:

... | C-2 | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0, 0 | 1, 1, 1, ...

In PWM mode, this bit has no effect.

- **Bits 2,1,0 - CS02, CS01, CS00 / CS22, CS21, CS20: Clock Select bits 2,1 and 0**

The Clock Select2 bits 2,1 and 0 define the prescaling source of the Timer/Counter.

Table 12. Timer/Counter0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Timer/Counter0 is stopped.
0	0	1	PCK0
0	1	0	PCK0/8
0	1	1	PCK0/32
1	0	0	PCK0/64
1	0	1	PCK0/128
1	1	0	PCK0/256
1	1	1	PCK0/1024

Table 13. Timer/Counter2 Prescale Select

CS22	CS21	CS20	Description
0	0	0	Timer/Counter2 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin PD7(T2), falling edge
1	1	1	External Pin PD7(T2), rising edge

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

Timer/Counter0 - TCNT0

Bit	7	6	5	4	3	2	1	0	
$32 ($42)	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Timer/Counter2 - TCNT2

Bit	7	6	5	4	3	2	1	0	
$24 ($44)	MSB							LSB	TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

These 8-bit registers contains 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 tl 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.

abaco bus grifo® ITALIAN TECHNOLOGY

Timer/Counter0 Output Compare Register - OCR0

Bit	7	6	5	4	3	2	1	0	
$31 ($51)	MSB							LSB	OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Timer/Counter2 Output Compare Register - OCR2

Bit	7	6	5	4	3	2	1	0	
$23 ($43)	MSB							LSB	OCR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

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/Count Actions on compare matches are specified in TCCR0 and TCCR2. A compare match does only occur if the Timer/Count counts to the OCR value. A software write that sets the Timer/Counter and Output Compare Register to the same val 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-b free-running, glitch-free and phase correct PWM with outputs on the PB4(OC0/PWM0) or PB7(OC2/PWM2) pin. Tl Timer/Counter acts as an up/down counter, counting up from $00 to $FF, where it turns and counts down again to ze before the cycle is repeated. When the counter value matches the contents of the Output Compare register, tl PB4(OC0/PWM0) or PB7(OC2/PWM2) pin is set or cleared according to the settings of the COM01/COM00 COM21/COM20 bits in the Timer/Counter Control Registers TCCR0 and TCCR2. Refer to Table 14 for details.

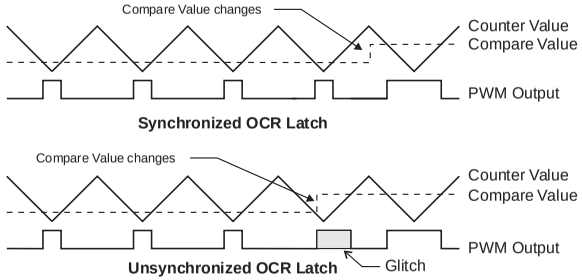
Table 14. Compare Mode Select in PWM Mode

COMn1	COMn0	Effect on Compare/PWM Pin
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).
1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).

Note: n = 0 or 2

Note that 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 t event of an unsynchronized OCR0 or OCR2 write. See Figure 32 for an example.

Figure 32. Effects on Unsynchronized OCR Latching



During the time between the write and the latch operation, a read from OCR0 or OCR2 will read the contents of the temp rary 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 hi 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

COMn1	COMn0	OCRn	Output PWMn
1	0	$00	L
1	0	$FF	H
1	1	$00	H
1	1	$FF	L

Note: n = 0 or 2

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

The frequency of the PWM will be Timer Clock Frequency divided by 510.

Asynchronous Status Register - ASSR

Bit	7	6	5	4	3	2	1	0	
$30 ($50)	-	-	-	-	AS0	TCN0UB	OCR0UB	TCR0UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7..4 - Res: Reserved Bits**

These bits are reserved bits in the ATmega603/103 and always reads as zero.

- **Bit 3 - AS0: Asynchronous Timer/Counter0**

When set(one) Timer/Counter0 is clocked from the TOSC1 pin. When cleared (zero) Timer/Counter0 is clocked from tl internal system clock, CK. When the value of this bit is changed the contents of TCNT0 might get corrupted.

- **Bit 2 - TCN0UB: Timer/Counter0 Update Busy**

When Timer/Counter0 operates asynchronously and TCNT0 is written, this bit becomes set (one). When TCNT0 has be updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates th TCNT0 is ready to be updated with a new value.

- **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 be updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates th OCR0 is ready to be updated with a new value.

- **Bit 0 - TCR0UB: Timer/Counter Control Register0 Update Busy**

When Timer/Counter0 operates asynchronously and TCCR0 is written, this bit becomes set (one). When TCCR0 has be updated from the temporary storage register, this bit is cleared (zero) by hardware. A logical zero in this bit indicates th 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 val might get corrupted and cause an unintentional interrupt to occur.

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 asynchr nous operation, however, some considerations must be taken.

- WARNING: When switching between asynchronous and synchronous clocking of Timer/Counter0, the timer registers, TCNT0, OCR0 and TCCR0 might get corrupted. Safe procedure for switching clock source:
  1. Disable the timer 0 interrupts OCIE0 and TOIE0.
  2. Select clock source by setting ASO as appropriate.
  3. Write new values to TCNT0, OCR0 and TCCR0.
  4. If switching to asynchronous operation: Wait for TCNT0UB, OCR0UB and TCR0UB to be cleared.
  5. Enable interrupts if needed.
- The oscillator is optimized for use with a 32,768Hz watch crystal. An external clock signal applied to this pin goes through the same amplifier having a bandwidth of 256kHz. The external clock signal should therefore be in the interva 0Hz - 256kHz. The frequency of the clock signal applied to the TOSC1 pin must be lower than one fourth of the CPU main clock frequency. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled.
- When writing to one 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 temporar 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 t the destination register has taken place, a Asynchronous Status Register - ASSR has been implemented.
- When entering Power Save mode after having written to TCNT0, OCR0 or TCCR0, the user must wait until the written register has been updated if Timer/Counter0 is used to wake up the device. Otherwise, the MCU will go to sleep befor the changes have had any effect. This is extremely important if the output compare0 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 u
- If Timer/Counter0 is used to wake up the device from Power Save mode, precautions must be taken if the user wants t re-enter Power Save mode; The interrupt logic needs one TOSC1 cycle to get reset. If the time between wake up and r 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 is in doubt whether the time before re-entering Power Save is sufficient, the following algorithm can be used t ensure that one TOSC1 cycle has elapsed:
  1. Write a value to TCCR0, TCNT0 or OCR0
  2. Wait until the corresponding Update Busy flag in ASSR returns to zero.
  3. Enter Power Save mode
- When asynchronous operation is selected, the 32 kHz oscillator for Timer/Counter0 is always running, except in powe down mode. After a power up reset or wake-up from power down, the user should be aware of the fact that this oscillat might take as long as one second to stabilize. Therefore, the content of all Timer/Counter0 registers must be considere

lost after a wake-up from power down, due to the unstable clock signal. The user is advised to wait for at least one second before using Timer/Counter0 after power-up or wake-up from power down.

- Description of wake up from power save mode when the timer is clocked asynchronously: When the interrupt condition met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by a 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 part will still wake up from power down, bu 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 th interrupt routine has not started yet.
- During asynchronous operation, the synchronization of the interrupt flags for the asynchronous timer takes 3 processc 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.

16-bit Timer/Counter1

Figure 33 shows the block diagram for Timer/Counter1.

The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopp as described in the specification for the Timer/Counter1 Control Register - TCCR1B. The different status flags (overflo compare match and capture event) are found in the Timer/Counter Interrupt Flag Register - TIFR. Control signals are fou in the Timer/Counter1 Control Registers - TCCR1A and TCCR1B. The interrupt enable/disable settings for Timer/Counte are found in the Timer/Counter Interrupt Mask Register - TIMSK.

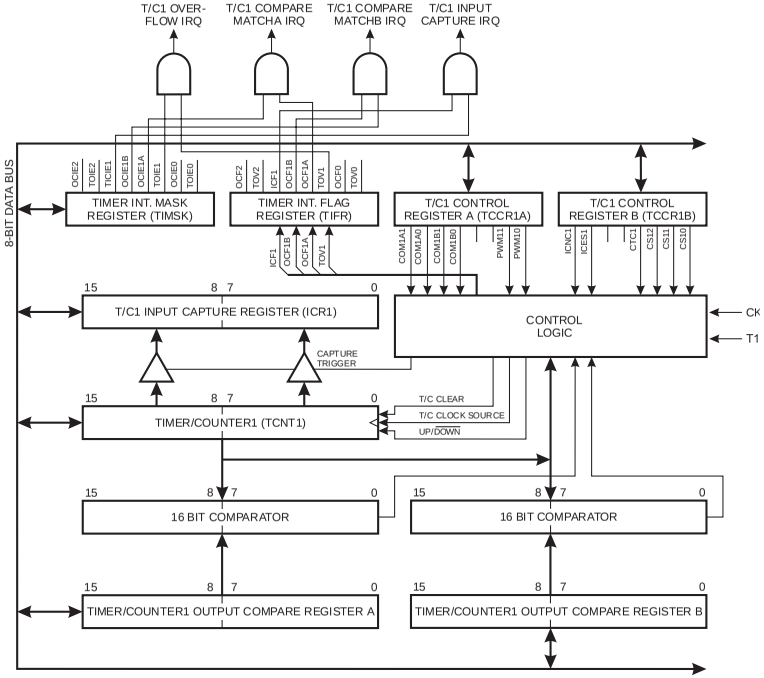
When Timer/Counter1 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least o internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 16-bit Timer/Counter1 features both a high resolution and a high accuracy usage with the lower prescaling opportu ties. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timi functions with infrequent actions.

The Timer/Counter1 supports two Output Compare functions using the Output Compare Register 1 A and B - OCR1A a OCR1B as the data sources to be compared to the Timer/Counter1 contents. The Output Compare functions inclu optional clearing of the counter on compareA match, and actions on the Output Compare pins on both compare matches

abacus bus grifo® ITALIAN TECHNOLOGY

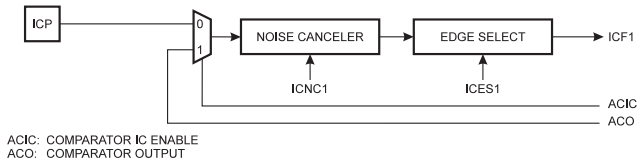
Figure 33. Timer/Counter1 Block Diagram



Timer/Counter1 can also be used as a 8, 9 or 10-bit Pulse With Modulator. In this mode the counter and tl 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 Captu Register - ICR1, triggered by an external event on the Input Capture Pin - PD4/(IC1). The actual capture event settin are defined by the Timer/Counter1 Control Register - TCCR1B. In addition, the Analog Comparator can be set to trigg the Input Capture. Refer to the paragraph, "The Analog Comparator", for details on this. The ICP pin logic is shov in Figure 34.

Figure 34. ICP Pin Schematic Diagram



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

Timer/Counter1 Control Register A - TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F ($4F)	**COM1A1**	**COM1A0**	**COM1B1**	**COM1B0**	**-**	**-**	**PWM11**	**PWM10**	**TCCR1A**
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..6 - COM1A1, COM1A0: Compare Output Mode1A, bits 1 and 0**

The COM1A1 and COM1A0 control bits determine any output pin action following a compare match in Timer/Counter Any output pin actions affect pin OC1A - Output CompareA pin 1. This is an alternative function to an I/O port, and the co responding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 16

- **Bits 5..4 - COM1B1, COM1B0: Compare Output Mode1B, bits 1 and 0**

The COM1B1 and COM1B0 control bits determine any output pin action following a compare match in Timer/Counter Any output pin actions affect pin OC1B - Output CompareB. Since this is an alternative function to an I/O port, the corr sponding direction control bit must be set (one) to control an output pin. The following control configuration is given:

Table 16. Compare 1 Mode Select

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

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

- **Bits 3..2 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and always read zero.

- **Bits 1..0 - PWM11, PWM10: Pulse Width Modulator Select Bits**

These bits select PWM operation of Timer/Counter1 as specified in Table 19. This mode is described on page 49.

Table 17. PWM Mode Select

PWM11	PWM10	Description
0	0	PWM operation of Timer/Counter1 is disabled
0	1	Timer/Counter1 is an 8-bit PWM
1	0	Timer/Counter1 is a 9-bit PWM
1	1	Timer/Counter1 is a 10-bit PWM

ITALIAN TECHNOLOGY grifo® abaco bus

Timer/Counter1 Control Register B - TCCR1B

Bit	7	6	5	4	3	2	1	0	
$2E ($4E)	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **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 tri gered at the first rising/falling edge sampled on the input capture pin PD4(IC1) as specified. When the ICNC1 bit is s (one), four successive samples are measures on PD4(IC1), and all samples must be high/low according to the input ca ture 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 - the falling edge of the input capture pin - PD4(IC1). While the ICES1 bit is set (one), the Timer/Counter1 contents are trar ferred 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 compareA match. the CTC1 control bit is cleared, Timer/Counter1 continues counting and is unaffected by a compare match. Since the co pare match is detected in the CPU clock cycle following the match, this function will behave differently when a prescali 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 v count as follows i CTC1 is set:

... | C-2 | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0 | ...

In PWM mode, this bit has no effect.

- **Bits 2,1,0 - CS12, CS11, CS10: Clock Select1, bit 2,1 and 0**

The lock Select1 bits 2,1 and 0 define the prescaling source of Timer/Counter1.

Table 18. Clock 1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Stop, the Timer/Counter1 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T1, falling edge
1	1	1	External Pin T1, rising edge

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

Timer/Counter1 - TCNT1H and TCNT1L

Bit	15	14	13	12	11	10	9	8	
$2D ($4D)	MSB								TCNT1H
$2C ($4C)								LSB	TCNT1L
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low byt are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit tei porary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B and ICR1. If the ma program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during acce 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 CF writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP register, and all 16 bits a written to the TCNT1 Timer/Counter1 register simultaneously. Consequently, the high byte TCNT1H must be access first for a full 16-bit register write operation. When using Timer/Counter1 as an 8-bit timer, it is sufficient to write the lo byte 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 by TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives tl data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register rea 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 up/down (in PWM mode) counter with read and write access. If Timer/Counte is written to and a clock source is selected, the Timer/Counter1 continues counting in the clock cycle after it is preset w the written value.

grifo® ITALIAN TECHNOLOGY

Since the Input Capture Register - ICR1 - is a 16-bit register, a temporary register TEMP is used when ICR1 is read ensure that both bytes are read simultaneously. When the CPU reads the low byte ICR1L, the data is sent to the CPU a the data of the high byte ICR1H is placed in the TEMP register. When the CPU reads the data in the high byte ICR1H, t CPU receives the data in the TEMP register. Consequently, the low byte ICR1L must be accessed first for a full 16-bit re ister read operation.

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

Timer/Counter1 in PWM mode

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1A - OCR1A and the Output Compa Register1B - OCR1B, form a dual 8, 9 or 10-bit, free-running, glitch-free and phase correct PWM with outputs on t PB5(OC1A) and PB6(OC1B) pins. Timer/Counter1 acts as an up/down counter, counting up from $0000 to TOP (see Tab 17), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the co tents of the 10 least significant bits of OCR1A or OCR1B, the PB5(OC1A)/PB6(OC1B) pins are set or cleared according the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register TCCR1A. Refer Table 20 for details.

Table 19. Timer TOP Values and PWM Frequency

PWM Resolution	Timer TOP value	Frequency
8-bit	$00FF (255)	$f_{TCK1}/510$
9-bit	$01FF (511)	$f_{TCK1}/1022$
10-bit	$03FF(1023)	$f_{TCK1}/2046$

Table 20. Compare1 Mode Select in PWM Mode

COM1X1	COM1X0	Effect on OCX1
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).
1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).

Note: X = A or B

Note that in the PWM mode, the 10 least significant OCR1A/OCR1B bits, when written, are transferred to a temporary loc tion. They are latched when Timer/Counter1 reaches the value TOP. This prevents the occurrence of odd-length PW pulses (glitches) in the event of an unsynchronized OCR1A/OCR1B write. See Figure 35 for an example.

Timer/Counter1 Output Compare Register - OCR1AH and OCR1AL

Bit	15	14	13	12	11	10	9	8	
$2B	MSB								OCR1AH
$2A								LSB	OCR1AL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Timer/Counter1 Output Compare Register - OCR1BH and OCR1BL

Bit	15	14	13	12	11	10	9	8	
$29	MSB								OCR1BH
$28								LSB	OCR1BL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

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/Counte Actions on compare matches are specified in the Timer/Counter1 Control and Status register. A compare match does o occur if Timer/Counter1 counts to the OCR value. A software write that sets TCNT1 and OCR1A or OCR1B to the sa 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 wh OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1 or OCR1BH, the data is temporarily stored in the TEMP register. When the CPU writes the low byte, OCR1AL or OCR1B the TEMP register is simultaneously written to OCR1AH or OCR1BH. Consequently, the high byte OCR1AH or OCR1B must be written first for a full 16-bit register write operation.

The TEMP register is also used when accessing TCNT1, and ICR1. If the main program and also interrupt routines perfo access to registers using TEMP, interrupts must be disabled during access from the main program.

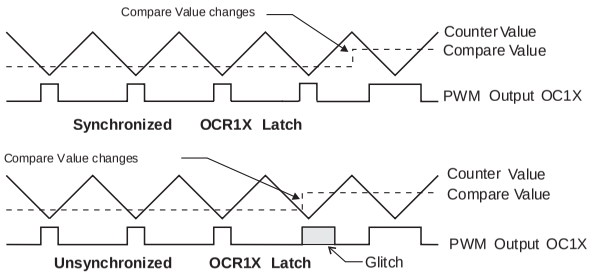
Timer/Counter1 Input Capture Register - ICR1H and ICR1L

Bit	15	14	13	12	11	10	9	8	
$27 ($37)	MSB								ICR1H
$26 ($36)								LSB	ICR1L
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

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 pi PD4(IC1) - is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register - ICR1. At t same time, the input capture flag - ICF1 - is set (one).

Figure 35. Effects on Unsynchronized OCR1 Latching



Note: X = A or B

During the time between the write and the latch operation, a read from OCR1A or OCR1B will read the contents of the ter porary 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 compa 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

COM1X1	COM1X0	OCR1X	Output OC1X
1	0	$0000	L
1	0	TOP	H
1	1	$0000	H
1	1	TOP	L

Note: X = A or B

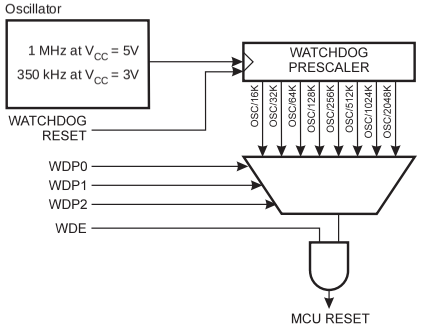
In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter advances from $0000. Timer Overflow Interrup operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV1 is set provided that Timer Overflo Interrupt1 and global interrupts are enabled. This does also apply to the Timer Output Compare1 flags and interrupts.

Watchdog Timer

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

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

Figure 36. Watchdog Timer



Watchdog Timer Control Register - WDTCR

Bit	7	6	5	4	3	2	1	0	
$21 ($41)	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..5 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and will always read as zero.

- **Bit 4 - WDTOE: Watch Dog Turn Off Enable**

This bit must be set (one) when the WDE bit is cleared, Otherwise, the watchdog will not be disabled. Once set, hardwa will clear this bit to zero after four clock cycles. Refer to the description of the WDE bit for a watchdog disable procedure.

- **Bit 3 - WDE: Watch Dog Enable**

When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer functi is disabled. WDE can only be cleared if the WDTOE bit is set (one). To disable an enabled watchdog timer, the followi procedure must be followed:

1. In the same operation, write a logical one to WDTOE and WDE. A logical one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog.

- **Bits 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. T different prescaling values and their corresponding Time-out Periods are shown in Table 22.

grifo® ITALIAN TECHNOLOGY

Table 22. Watch Dog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator cycles	Typical time-out at Vcc = 3.0V	Typical time-out at Vcc = 5.0V
0	0	0	16K cycles	47 ms	15 ms
0	0	1	32K cycles	94 ms	30 ms
0	1	0	64K cycles	0.19 s	60 ms
0	1	1	128K cycles	0.38 s	0.12 s
1	0	0	256K cycles	0.75 s	0,24 s
1	0	1	512K cycles	1.5 s	0.49 s
1	1	0	1,024K cycles	3.0 s	0.97 s
1	1	1	2,048K cycles	6.0 s	1.9 s

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 th
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 write access time is in the range of 2.5 - 4ms, depending on the V_{CC} voltages. A self-timing function lets the user sc
ware detect when the next byte can be written. A special EEPROM Ready interrupt can be set to trigger when tl
EEPROM is ready to accept new data.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description
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 it
read, the CPU is halted for 4 clock cycles.

EEPROM Address Register - EEARH, EEARL

Bit	15	14	13	12	11	10	9	8	
$1F ($3F)	-	-	-	-	EEAR11	EEAR10	EEAR9	EEAR8	EEARH
$1E ($3E)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The EEPROM Address Registers - EEARH and EEARL specify the EEPROM address in the 2K/4K-byte EEPROM spac
The EEPROM data bytes are addressed linearly between 0 and 2047/4095. The ATmega603 has an EEPROM addre
space of 2K, and the EEAR11 I/O bit is read-only with initial value of 0.

EEPROM Data Register - EEDR

Bit	7	6	5	4	3	2	1	0	
$1D ($3D)	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..0 - EEDR7..0: EEPROM Data:**

For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address giv
by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at t
address given by EEAR.

EEPROM Control Register - EECR

Bit	7	6	5	4	3	2	1	0	
$1C ($3C)	-	-	-	-	EERIE	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **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 I bit in SREG and EERIE are set (one), the EEPROM Ready Interrupt is enabled. When cleared (zero), the inte
rupt is disabled. The EEPROM Ready interrupt constantly generates an interrupt request when EEWE is cleared (zero).

- **Bit 2 - EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set(on
setting EEWE will write data to the EEPROM at the selected address If EEMWE is zero, setting EEWE will have no effe
When EEMWE has been set (one) by software, hardware clears the bit to zero after four clock cycles. See the descripti
of the EEWE 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 u
the EEWE bit must be set to write the value into the EEPROM. The EEMWE bit must be set when the logical one is writt
to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing tl
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 w
time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR and EEDR re
ister will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt fl
cleared during the 4 last steps to avoid these problems.

When the write access time (typically 2.5 ms at V_{CC} = 5V or 4 ms at V_{CC} = 2.7V) has elapsed, the EEWE bit is clear
(zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE h
been 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 tl
EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in tl
EEDR register. The EEPROM read access takes one instruction there is no need to poll the EERE bit. When EERE h
been set, the CPU is halted for four cycles before the next instruction is executed.

ITALIAN TECHNOLOGY grifo® abaco bus

The user should poll the EEWE bit before starting the read operation. If a write operation is in progress when new data address 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 tl EEPROM to operate properly. These issues are the same as for board level systems using the EEPROM, and the sar 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 sequen to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions inc rectly, 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):

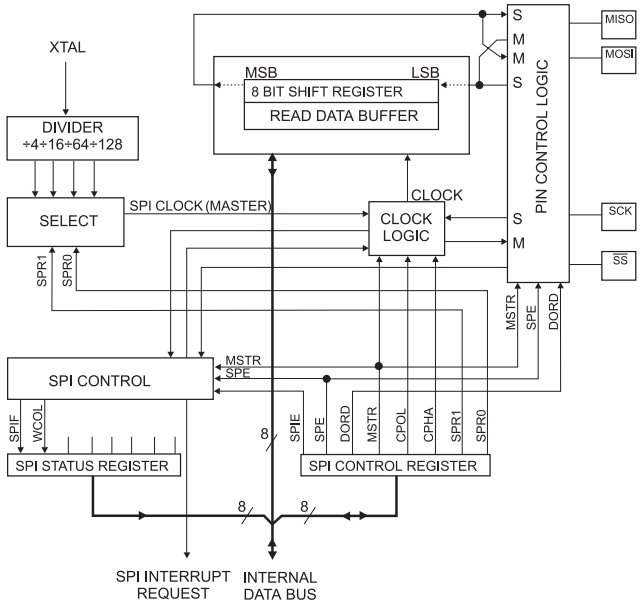
1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This is best done by an external low V_{CC} Reset Protection circuit, often referred to as a Brown-Out Detector (BOD). Please refer to application note AVR 180 for design considerations regarding power-on reset and 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 a 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)

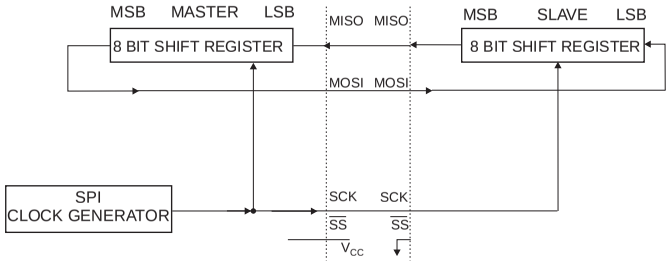
Figure 37. SPI Block Diagram



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

grifo® ITALIAN TECHNOLOGY

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 chara ters to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receivi data, however, a received byte must be read from the SPI Data Register before the next byte has been completely shifte in. Otherwise, the first byte is lost.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and $\overline{SS}$ pins is overridden according to the followi table:

Table 23. SPI Pin Overrides

PIN	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

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.

$\overline{SS}$ Pin Functionality

When the SPI is configured as a master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin. If $\overline{SS}$ configured as an output, the pin is a general output pin which does not affect the SPI system. If $\overline{SS}$ is configured as input, it must be hold high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI configured as master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another master selecting t 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 transmittal is used in master mode, and there exists a possibility that $\overline{SS}$ is driven low, t interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it mu be set by the user to re-enable SPI master mode.

When the SPI is configured as a slave, the $\overline{SS}$ pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MIS becomes an output if configured so by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs, a the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ p is brought high. If the $\overline{SS}$ pin is brought high during a transmission, the SPI will stop sending and receiving immediately a 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 b 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

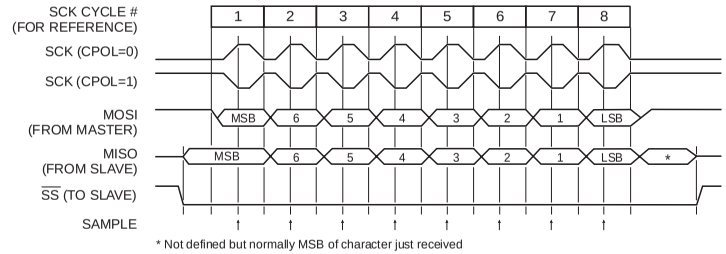
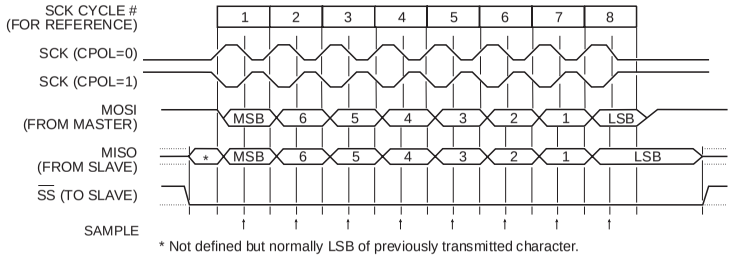


Figure 40. SPI Transfer Format with CPHA = 1and DORD = 0



SPI Control Register - SPCR

Bit	7	6	5	4	3	2	1	0	
$0D ($2D)	**SPIE**	**SPE**	**DORD**	**MSTR**	**CPOL**	**CPHA**	**SPR1**	**SPR0**	**SPCR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - SPIE: SPI Interrupt Enable**

This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR register is set and the global interrupts are enable

- **Bit 6 - SPE: SPI Enable**

When the SPE bit is set (one), the SPI is enabled and $\overline{SS}$, MOSI, MISO and SCK are connected to pins PB0, PB1, PB2 a PB3.

- **Bit 5 - DORD: Data Order**

When the DORD bit is set (one), the LSB of the data word is transmitted first.

When the DORD bit is cleared (zero), the MSB of the data word is transmitted first.

- **Bit 4 - MSTR: Master/Slave Select**

This bit selects Master SPI mode when set (one), and Slave SPI mode when cleared (zero). If $\overline{SS}$ is configured as an inp and is driven low while 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 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. T relationship between SCK and the CPU Clock frequency f_{cl} is shown in the following table:

Table 24. Relationship Between SCK and the Oscillator Frequency

SPR1	SPR0	SCK Frequency
0	0	f_{cl} / 4
0	1	f_{cl} / 16
1	0	f_{cl} / 64
1	1	f_{cl} / 128

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

SPI Status Register - SPSR

Bit	7	6	5	4	3	2	1	0	
$0E	SPIF	WCOL	-	-	-	-	-	-	SPSR
Read/Write	R	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) a global interrupts are enabled. SPIF is cleared by hardware when executing the corresponding interrupt handling vect Alternatively, the SPIF bit is cleared by first reading the SPI status register with SPIF set (one), then accessing the S Data Register (SPDR).

- **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 (and the SPIF bit) a cleared (zero) by first reading the SPI Status Register with WCOL set (one), and 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

Bit	7	6	5	4	3	2	1	0	
$0F ($2F)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	x	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 regist Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be rea

UART

The ATmega603/103 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver a Transmitter (UART). The main features are:

- Baud rate generator that can generate a large number of baud rates (bps)
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection
- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete

Data Transmission

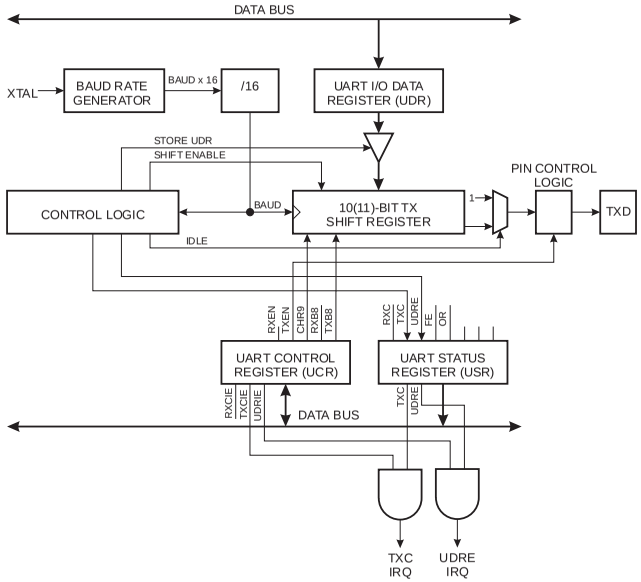
A block schematic of the UART transmitter is shown in Figure 41.

Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data is transferr from UDR to the Transmit shift register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The shift register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The sh register is loaded when the stop bit of the character currently being transmitted has been shifted out.

If the 10(11)-bit Transmitter shift register is empty, data is transferred from UDR to the shift register. At this time the UDF (UART Data Register Empty) bit in the UART Status Register, USR, is set. When this bit is set (one), the UART is ready receive the next character. Writing to UDR clears UDRE. At the same time as the data is transferred from UDR to t 10(11)-bit shift register, bit 0 of the shift register is cleared (start bit) and bit 9 or 10 is set (stop bit). If 9 bit data word selected (the CHR9 bit in the UART Control Register, UCR is set), the TXB8 bit in UCR is transferred to bit 9 in the Tran mit shift register.

Figure 41. UART Transmitter

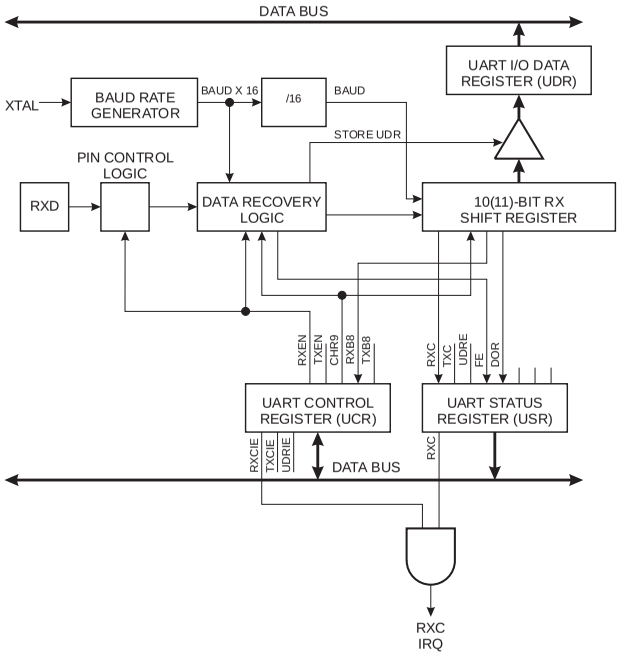


On the Baud Rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin, f lowed by the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has be written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to se 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 f 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 l used for general I/O. When TXEN is set, the UART Transmitter will be connected to PE1, which is forced to be an outp pin regardless of the setting of the DDE1 bit in DDRE.

Data Reception

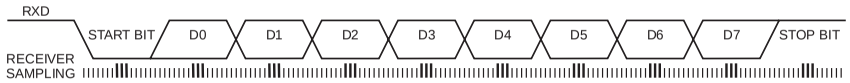
Figure 42. UART Receiver



The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is id 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 sample 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 noi 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 al sampled at samples 8, 9, and 10. The logical value found in at least two of the 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 Figu 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 mo samples are logical zeros, the Framing Error (FE) flag in the UART Status Register (USR) is set when the received byte transferred to UDR. Before reading the UDR register, the user should always check the FE bit to detect Framing Errors. F 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 tl RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received dat When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register accessed. If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR loaded with bit 9 in the Transmit shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been accessed since the last receive, the OverRun (OR) fl in USR is set. This means that the new data transferred to the shift register could not be transferred to UDR and is lost. Tl OR bit is buffered, and is available when the valid data byte in UDR has been read. The user should always check the C 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 us 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 p regardless of the setting of the DDE0 bit in DDRE. When PE0 is forced to input by the UART, the PORTE0 bit can still l used to control the pull-up resistor on the pin.

When the CHR9 bit in the UCR register is set, transmitted and received characters are 9-bit long plus start and stop bi 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 tran mission is initated by writing to the UDR register. The 9th

UART Control

UART I/O Data Register - UDR

Bit	7	6	5	4	3	2	1	0	
$0C ($2C)	MSB							LSB	UDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UDR register is actually two physically separate registers sharing the same I/O address. When writing to the registe the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

UART Status Register - USR

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	RXC	TXC	UDRE	FE	OR	-	-	-	USR
Read/Write	R	R/W	R	R	R	R	R	R	
Initial value	0	0	1	0	0	0	0	0	

The USR register is a read-only register providing information on the UART Status.

- **Bit 7 - RXC: UART Receive Complete**

This bit is set (one) when a received character is transferred from the Receiver Shift register to UDR. The bit is set regar less of any detected framing errors. When the RXCIE bit in UCR is set, the UART Receive Complete interrupt will l executed when RXC is set(one). RXC is cleared by reading UDR. When interrupt-driven data reception is used, the UAF Receive Complete Interrupt routine must read UDR in order to clear RXC, otherwise a new interrupt will occur once tl interrupt routine terminates.

- **Bit 6 - TXC: UART Transmit Complete**

This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift register has been shifted out a no new data has been written to the UDR. This flag is especially useful in half-duplex communications interfaces, where transmitting application must enter receive mode and free the communications bus immediately after completing tl transmission.

When the TXCIE 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 clear (zero) by writing a logical one to the bit.

- **Bit 5 - UDRE: UART Data Register Empty**

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

When the UDRIE bit in UCR is set, the UART Transmit Complete interrupt to be executed as long as UDRE is set. UDRE cleared by writing UDR. When interrupt-driven data transmittal is used, the UART Data Register Empty Interrupt routi must write UDR in order to clear UDRE, otherwise a new interrupt will occur once the interrupt routine terminates.

UDRE is set (one) during reset to indicate that the transmitter is ready.

- **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 one.

- **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 re before the next character is transferred from the Receiver Shift register. The OR bit is buffered, which means that it is w be set once the valid data still in UDRE is read.

The OR bit is cleared (zero) when data is received and transferred to UDR.

- **Bits 2..0 - Res: Reserved bits**

These bits are reserved bits in the ATmega603/103 and will always read as zero.

UART Control Register - UCR

Bit	7	6	5	4	3	2	1	0	
$0A ($2A)	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	UCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W	
Initial value	0	0	0	0	0	0	1	0	

- **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 execut 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 execute 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 l executed provided that global interrupts are enabled.

- **Bit 4 - RXEN: Receiver Enable**

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

- **Bit 3 - TXEN: Transmitter Enable**

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

- **Bit 2 - 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 is read a written by using the RXB8 and TXB8 bits in UCR, respectively. The 9th data bit can be used as an extra stop bit or a par bit.

- **Bit 1 - RXB8: Receive Data Bit 8**

When CHR9 is set (one), RXB8 is the 9th data bit of the received character.

grifo® ITALIAN TECHNOLOGY

- **Bit 0 - TXB8: Transmit Data Bit 8**

When CHR9 is set (one), TXB8 is the 9th data bit in the character to be transmitted.

Baud Rate Generator

The baud rate generator is a frequency divider which generates baud rates according to the following equation:

$$BAUD = \frac{f_{CK}}{16(UBRR + 1)}$$

- BAUD = Baud Rate
- f_{CK} = CPU Clock frequency
- UBRR = Contents of the UART Baud Rate register, UBRR (0 - 255)

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBRR settings Table 25. Observe that CPU clock frequency can be lower than the XTAL frequency if the XTAL divider is enabled. UBF values which yield an actual baud rate differing less than 2% from the target baud rate, are bolded in the table. Howev using baud rates that have more than 1% error is not recommended. High error ratings give less noise resistance.

Table 25. UBRR Settings at Various CPU Frequencies

Baud Rate	1 MHz		%Error	1.8432 MHz		%Error	2 MHz		%Error	2.4576 MHz		%Error
2400	UBRR=	**25**	**0.2**	UBRR=	**47**	**0.0**	UBRR=	**51**	**0.2**	UBRR=	**63**	**0.0**
4800	UBRR=	**12**	**0.2**	UBRR=	**23**	**0.0**	UBRR=	**25**	**0.2**	UBRR=	**31**	**0.0**
9600	UBRR=	6	7.5	UBRR=	**11**	**0.0**	UBRR=	**12**	**0.2**	UBRR=	**15**	**0.0**
14400	UBRR=	3	7.8	UBRR=	**7**	**0.0**	UBRR=	8	3.7	UBRR=	10	3.1
19200	UBRR=	2	7.8	UBRR=	**5**	**0.0**	UBRR=	6	7.5	UBRR=	**7**	**0.0**
28800	UBRR=	1	7.8	UBRR=	**3**	**0.0**	UBRR=	3	7.8	UBRR=	4	6.3
38400	UBRR=	1	22.9	UBRR=	**2**	**0.0**	UBRR=	2	7.8	UBRR=	**3**	**0.0**
57600	UBRR=	0	7.8	UBRR=	**1**	**0.0**	UBRR=	1	7.8	UBRR=	2	12.5
76800	UBRR=	0	22.9	UBRR=	1	33.3	UBRR=	1	22.9	UBRR=	**1**	**0.0**
115200	UBRR=	0	84.3	UBRR=	**0**	**0.0**	UBRR=	0	7.8	UBRR=	0	25.0

Baud Rate	3.2768 MHz		%Error	3.6864 MHz		%Error	4 MHz		%Error	4.608 MHz		%Error
2400	UBRR=	**84**	**0.4**	UBRR=	**95**	**0.0**	UBRR=	**103**	**0.2**	UBRR=	**119**	**0.0**
4800	UBRR=	**42**	**0.8**	UBRR=	**47**	**0.0**	UBRR=	**51**	**0.2**	UBRR=	**59**	**0.0**
9600	UBRR=	**20**	**1.6**	UBRR=	**23**	**0.0**	UBRR=	**25**	**0.2**	UBRR=	**29**	**0.0**
14400	UBRR=	**13**	**1.6**	UBRR=	**15**	**0.0**	UBRR=	16	2.1	UBRR=	**19**	**0.0**
19200	UBRR=	10	3.1	UBRR=	**11**	**0.0**	UBRR=	**12**	**0.2**	UBRR=	**14**	**0.0**
28800	UBRR=	**6**	**1.6**	UBRR=	**7**	**0.0**	UBRR=	8	3.7	UBRR=	**9**	**0.0**
38400	UBRR=	4	6.3	UBRR=	**5**	**0.0**	UBRR=	6	7.5	UBRR=	7	6.7
57600	UBRR=	3	12.5	UBRR=	**3**	**0.0**	UBRR=	3	7.8	UBRR=	**4**	**0.0**
76800	UBRR=	2	12.5	UBRR=	**2**	**0.0**	UBRR=	2	7.8	UBRR=	3	6.7
115200	UBRR=	1	12.5	UBRR=	**1**	**0.0**	UBRR=	1	7.8	UBRR=	2	20.0

Baud Rate	7.3728 MHz		%Error	8 MHz		%Error	9.216 MHz		%Error	11.059 MHz		%Error
2400	UBRR=	**191**	**0.0**	UBRR=	**207**	**0.2**	UBRR=	**239**	**0.0**	UBRR=	287	-
4800	UBRR=	**95**	**0.0**	UBRR=	**103**	**0.2**	UBRR=	**119**	**0.0**	UBRR=	**143**	**0.0**
9600	UBRR=	**47**	**0.0**	UBRR=	**51**	**0.2**	UBRR=	**59**	**0.0**	UBRR=	**71**	**0.0**
14400	UBRR=	**31**	**0.0**	UBRR=	**34**	**0.8**	UBRR=	**39**	**0.0**	UBRR=	**47**	**0.0**
19200	UBRR=	**23**	**0.0**	UBRR=	**25**	**0.2**	UBRR=	**29**	**0.0**	UBRR=	**35**	**0.0**
28800	UBRR=	**15**	**0.0**	UBRR=	16	2.1	UBRR=	**19**	**0.0**	UBRR=	**23**	**0.0**
38400	UBRR=	**11**	**0.0**	UBRR=	**12**	**0.2**	UBRR=	**14**	**0.0**	UBRR=	**17**	**0.0**
57600	UBRR=	**7**	**0.0**	UBRR=	8	3.7	UBRR=	**9**	**0.0**	UBRR=	**11**	**0.0**
76800	UBRR=	**5**	**0.0**	UBRR=	6	7.5	UBRR=	7	6.7	UBRR=	**8**	**0.0**
115200	UBRR=	**3**	**0.0**	UBRR=	3	7.8	UBRR=	**4**	**0.0**	UBRR=	**5**	**0.0**

UART Baud Rate Register - UBRR

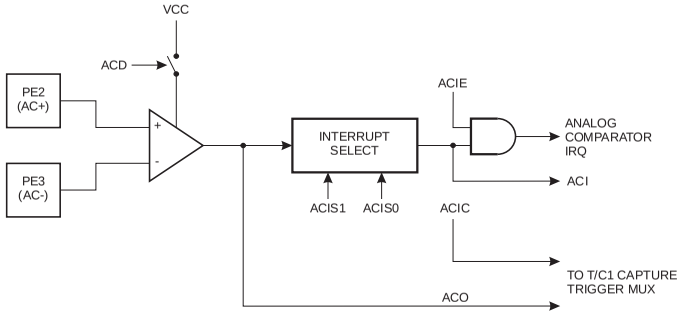
Bit	7	6	5	4	3	2	1	0	
$09 ($29)	**MSB**							**LSB**	**UBRR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UBRR is an 8-bit read/write register which specifies the UART Baud Rate according to the description on the previo page.

Analog Comparator

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

Figure 44. Analog Comparator Block Diagram



Analog Comparator Control and Status Register - ACSR

Bit	7	6	5	4	3	2	1	0	
$08 ($28)	**ACD**	**-**	**ACO**	**ACI**	**ACIE**	**ACIC**	**ACIS1**	**ACIS0**	**ACSR**
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **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 t analog comparator. This will reduce power consumption in active and idle mode. When changing the ACD bit, the Anal Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit changed.

- **Bit 6 - Res: Reserved bit**

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

- **Bit 5 - ACO: Analog Comparator Output**

ACO is directly connected to the comparator output.

- **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 Anal Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one the flag. Observe however, that if another bit in this register is modified using the SBI or CBI instruction, ACI will be clear if it has become set before the operation.

- **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 activate 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 comparat The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator utili the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no co nection between the analog comparator and the Input Capture function is given. To make the comparator trigger t Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

- **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 a shown in Table 26.

Table 26. ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge
1	1	Comparator Interrupt on Rising Output Edge

When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enat 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:

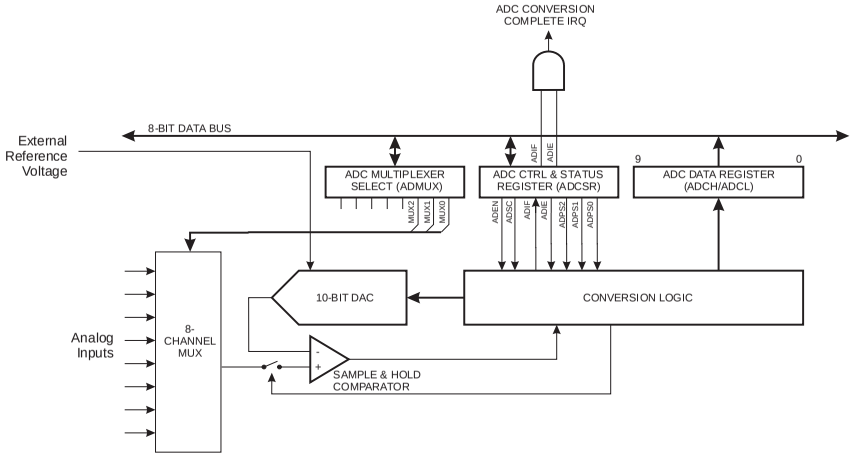
- 10-bit Resolution
- ±2 LSB absolute accuracy
- 0.5 LSB Integral Non-Linearity
- 70 - 280 µs conversion time
- Up to 14 kSPS
- 8 Multiplexed Input Channels
- Interrupt on ADC conversion complete.
- Sleep Mode Noise Canceler

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

The ADC has two separate analog supply voltage pins, AV_{CC} and AGND. AGND must be connected to GND, and the vo age on AV_{CC} must not differ more than ± 0.3 V from V_{CC}. See the section “ADC Noise Canceling Techniques” on page 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 - AV_{CC}.

Figure 45. Analog to Digital Converter Block Schematic



Operation

The ADC operates in Single Conversion mode, and each conversion will have to be initiated by the user.

The ADC is enabled by writing a logical one to the ADC Enable bit, ADEN in ADCSR. The first conversion that is start after enabling the ADC, will be preceded by a dummy conversion to initialize the ADC. To the user, the only difference v be that this conversion takes 13 more ADC clock pulses than a normal conversion. (See Figure 48.)

A conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit will stay high as long as t conversion is in progress and be set to zero by hardware when the conversion is completed. If a different data channel selected while a conversion is in progress, the ADC will finish the current conversion before performing the chann change.

As the ADC generates a 10-bit result, two data registers, ADCH and ADCL, must be read to get the result when the conve sion is complete. Special data protection logic is used to ensure that the contents of the data registers belong to the sar result when they are read. This mechanism works as follows:

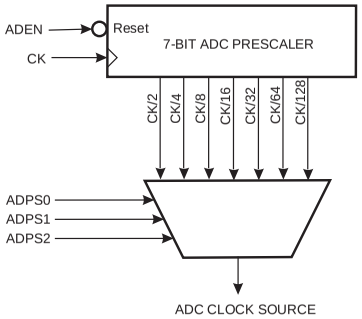
When reading data, ADCL must be read first. Once ADCL is read, ADC access to data registers is blocked. This mea that if ADCL has been read, and a conversion completes before ADCH is read, none of the registers are updated and t result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL registers is re-enabled.

The ADC has its own interrupt, ADIF, which can be triggered when a conversion completes. When ADC access to the da registers is prohibited between reading of ADCL and ADCH, the interrupt will trigger even if the result is lost.

abaco bus grifo® ITALIAN TECHNOLOGY

Prescaling

Figure 46. ADC Prescaler



The ADC contains a prescaler, which divides the system clock to an acceptable ADC clock frequency. The ADC accep input clock frequencies in the range 50 - 200 kHz. Applying a higher input frequency will result in a poorer accuracy, s "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 frequen above 100 kHz. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCS 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 t ADC clock cycle. The actual sample-and-hold takes place one ADC clock cycle after the start of the conversion. The res is ready and written to the ADC Result Register after 13 cycles. The ADC needs 2 more clock cycles before a new conv sion can be started. If ADSC is set high in this period, the ADC will start the new conversion immediately. For a summary conversion times, see Table 27.

Figure 47. ADC timing diagram, first conversion

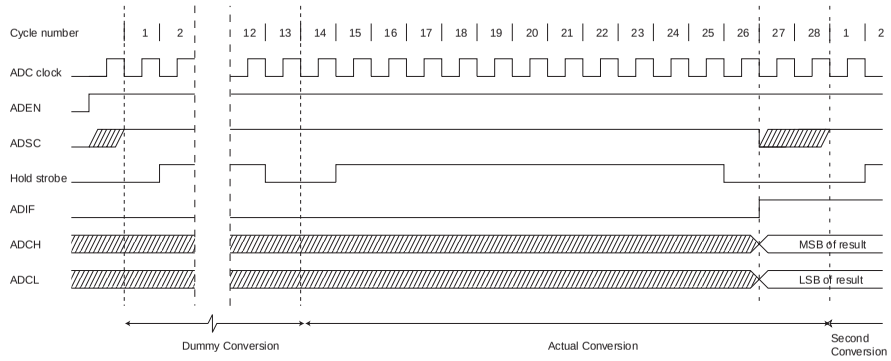
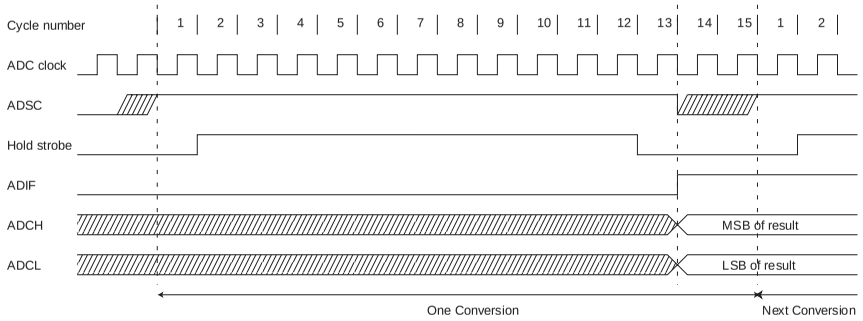


Table 27. ADC Conversion Time

Condition	Sample Cycle Number	Result Ready (cycle number)	Total Conversion Time (cycles)	Total Conversion Time (µs)
1st Conversion	14	26	28	140 - 560
Single Conversion	1	13	15	75 - 300

Figure 48. ADC Timing Diagram



ADC Noise Canceler Function

The ADC features a noise canceler that enables conversion during idle mode to reduce noise induced from the CPU co 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 and execute the ADC conversion complete interrupt routine.

ADC Multiplexer Select Register - ADMUX

Bit	7	6	5	4	3	2	1	0	
$07 ($27)	-	-	-	-	-	MUX2	MUX1	MUX0	ADMUX
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **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.

ADC Control and Status Register - ADCSR

Bit	7	6	5	4	3	2	1	0	
$06 ($26)	ADEN	ADSC	-	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **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 wh 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 h been enabled, or if ADSC is written at the same time as the ADC is enabled, a dummy conversion will precede the initiat conversion. This dummy conversion performs initialization of the ADC.

ADSC remains high during the conversion. ADSC goes low after the conversion is complete, but before the result is writt to the ADC Data Registers. This allows a new conversion to be initiated before the current conversion is complete. The ne conversion will then start immediately after the current conversion completes. When a dummy conversion precedes a re conversion, ADSC 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 the result is written to the ADC Data Registers a updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set (one). ADIF cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing logical one to the flag. Beware that if doing a read-modify-write on ADCSR, a pending interrupt can be disabled. This al applies if the SBI and CBI instructions are used.

- **Bit 3 - ADIE: ADC Interrupt Enable**

When this bit is set (one) and the I-bit in SREG is set (one), the ADC Conversion Complete Interrupt is activated.

- **Bits 2..0 - ADPS2..ADPS0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

Table 28. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	Invalid
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

ADC Data Register - ADCL and ADCH

Bit	15	14	13	12	11	10	9	8	
$05 ($25)	-	-	-	-	-	-	ADC9	ADC8	ADCH
$04 ($24)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

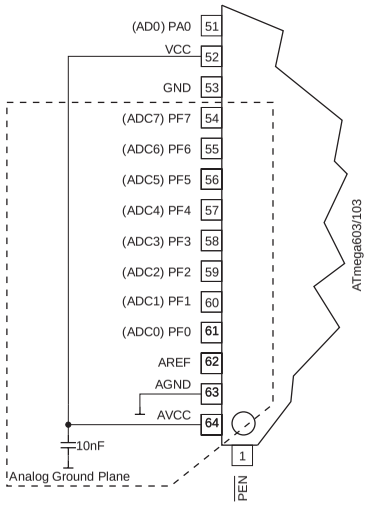
When an ADC conversion is complete, the result is found in these two registers. It is essential that both registers are rea and that ADCL is read before ADCH.

ADC Noise Canceling Techniques

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

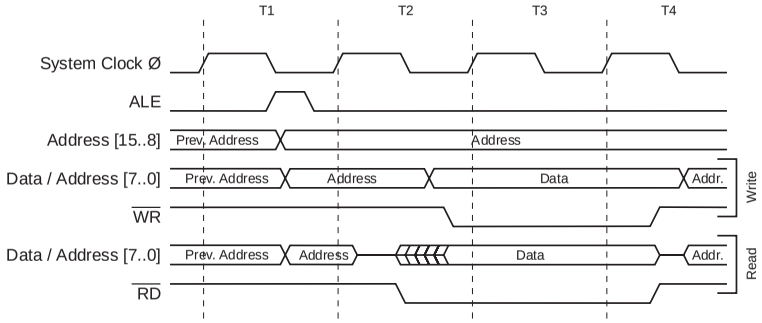
1. The analog part of the ATmega603/103 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.
2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
3. The AV_{CC} 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.

Figure 49. ADC Power Connections



abaco bus grifo® ITALIAN TECHNOLOGY

Figure 52. External SRAM Access Cycle with wait state



I/O-Ports

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the dire tion 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), Da Direction Register - DDRA, $1A($3A) and the Port A Input Pins - PINA, $19($39). The Port A Input Pins address is rea 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 LE displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if t 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 t 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 Reg ter, the alternate settings override the data direction register.

Port A Data Register - PORTA

Bit	7	6	5	4	3	2	1	0	
$1B ($3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port A Data Direction Register - DDRA

Bit	7	6	5	4	3	2	1	0	
$1A ($3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port A Input Pins Address - PINA

Bit	7	6	5	4	3	2	1	0	
$19 ($39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port A Input Pins address - PINA - is not a register, and this address enables access to the physical value on each P A pin. When reading PORTA the Port A Data Latch is read, and when reading PINA, the logical values present on the pi are read.

Port A as General Digital I/O

All 8 pins in Port A have equal functionality when used as digital I/O pins.

PAn, General I/O pin: The DDAn bit in the DDRA register selects the direction of this pin, if DDAn is set (one), PAn is cc figured as an output pin. If DDAn is cleared (zero), PAn is configured as an input pin. If PORTAn is set (one) when the p configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, PORTAn has to cleared (zero) or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becom active, even if the clock is not running.

Table 29. DDAn Effects on Port A Pins

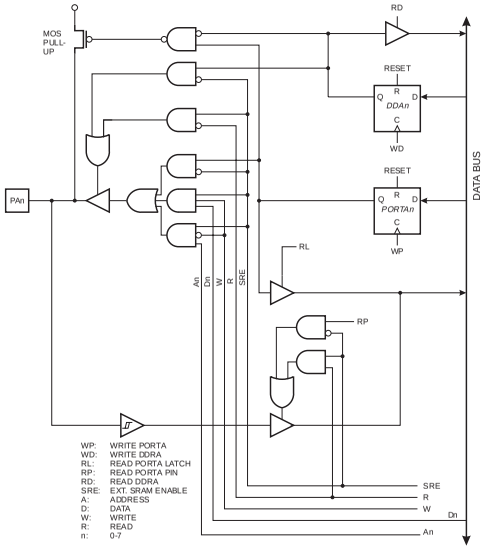
DDAn	PORTAn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PAn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

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

Port A Schematics

Note that all port pins are synchronized. The synchronization latch is however, not shown in the figure.

Figure 53. Port A Schematic Diagrams (Pins PA0 - PA7)



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), Da Direction Register - DDRB, $17($37) and the Port B Input Pins - PINB, $16($36). The Port B Input Pins address is rea only, 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 d plays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if tl 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

Port Pin	Alternate Functions
PB0	$\overline{SS}$ (SPI Slave Select input)
PB1	SCK (SPI Bus Serial Clock)
PB2	MOSI (SPI Bus Master Output/Slave Input)
PB3	MISO (SPI Bus Master Input/Slave Output)
PB4	OC0/PWM0 (Output Compare and PWM Output for Timer/Counter0)
PB5	OC1A/PWM1A (Output Compare and PWM Output A for Timer/Counter1)
PB6	OC1B/PWM1B (Output Compare and PWM Output B for Timer/Counter1)
PB7	OC2/PWM2 (Output Compare and PWM Output for Timer/Counter2

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

Port B Data Register - PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port B Data Direction Register - DDRB

Bit	7	6	5	4	3	2	1	0	
$17 ($37)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port B Input Pins Address - PINB

Bit	7	6	5	4	3	2	1	0	
$16 ($36)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port B Input Pins address - PINB - is not a register, and this address enables access to the physical value on each P B pin. When reading PORTB, the Port B Data Latch is read, and when reading PINB, the logical values present on the pi 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 cc figured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PORTBn is set (one) when the p configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PORTBn has to l cleared (zero) or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becom active, even if the clock is not running.

abaco bus grifo® ITALIAN TECHNOLOGY

Table 31. DDBn Effects on Port B Pins

DDBn	PORTBn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PBn will source current if ext. pulled low
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

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

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 p 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 Mod 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 Mod 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 p 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 configur as an input regardless of the setting of DDB3. When the SPI is enabled as a slave, the data direction of this pin is controll 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 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 configur as an input regardless of the setting of DDB2. When the SPI is enabled as a master, the data direction of this pin is co trolled by DDB2. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB2 bit. See tl 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 configur as an input regardless of the setting of DDB1. When the SPI is enabled as a master, the data direction of this pin is co trolled by DDB1. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB1 bit. See tl description of the SPI port for further details.

- **$\overline{SS}$ - Port B, Bit 0**

$\overline{SS}$: Slave port select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the settir 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 dire tion of this pin is controlled by DDB0. When the pin is forced to be an input, the pull-up can still be controlled by tl PORTB0 bit. See the description of the SPI port for further details.

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)

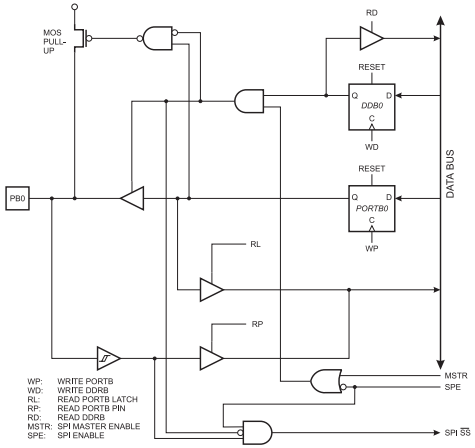


Figure 55. Port B Schematic Diagram (Pin PB1)

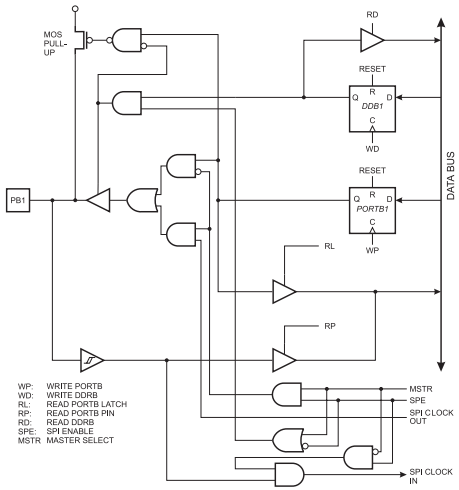


Figure 56. Port B Schematic Diagram (Pin PB2)

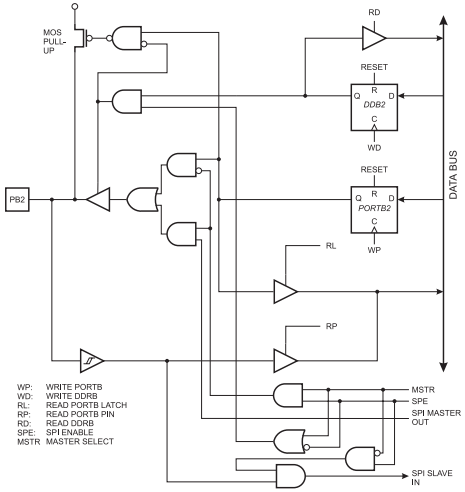


Figure 57. Port B Schematic Diagram (Pin PB3)

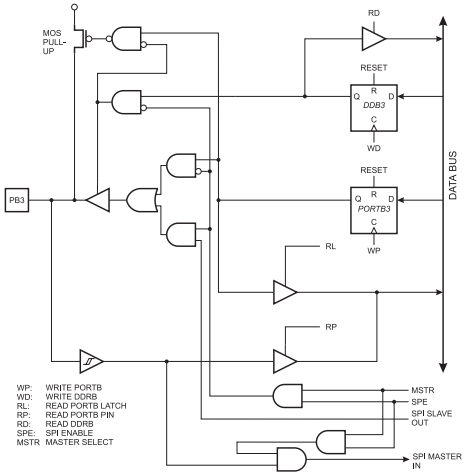


Figure 58. Port B Schematic Diagram (Pin PB4)

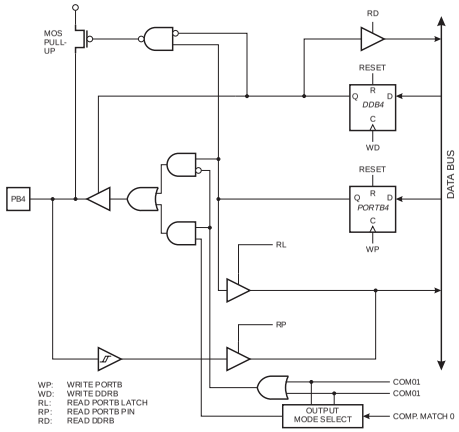


Figure 59. Port B Schematic Diagram (Pins PB5 and PB6)

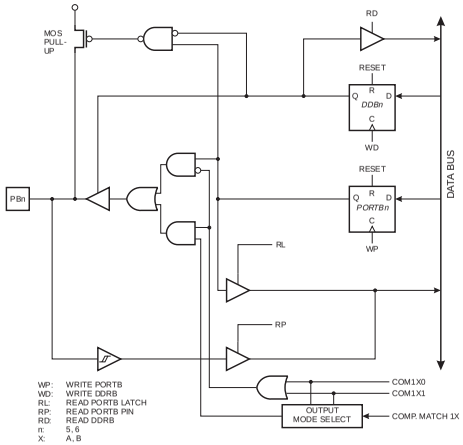
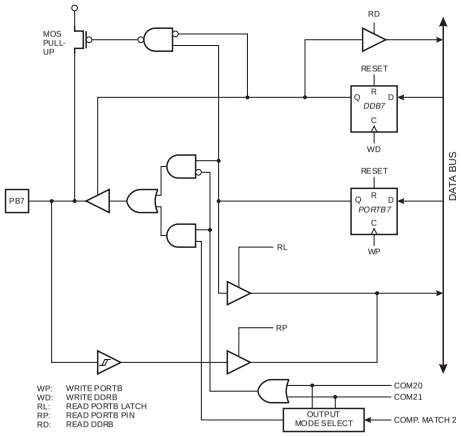


Figure 60. Port B Schematic Diagram (Pin PB7)



Port C

PORT C is an 8-bit Output port.

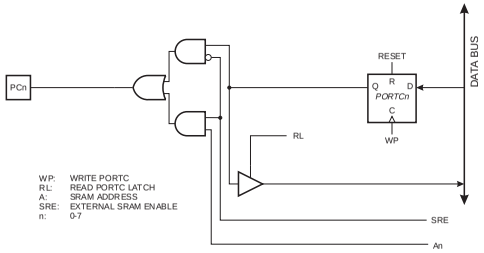
The Port C pins have alternate functions related to the optional external data SRAM. When using the device with exterr SRAM, Port C outputs the high-order address byte during accesses to external data memory. When a reset conditi becomes active, the port pins are not tristated, but the pins will assume their initial value after two stable clock cycles.

The Port C Data Register - PORTC

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port C Schematics

Figure 61. Port C Schematic Diagram (Pins PC0 - PC7)



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), Da Direction Register - DDRD, $11($31) and the Port D Input Pins - PIND, $10($30). The Port D Input Pins address is rea 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 pu up resistors are activated.

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

Table 32. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	$\overline{\text{INT0}}$ (External Interrupt0 Input)
PD1	$\overline{\text{INT1}}$ (External Interrupt1 Input)
PD2	$\overline{\text{INT2}}$ (External Interrupt2 Input)
PD3	$\overline{\text{INT3}}$ (External Interrupt3 Input)
PD4	IC1 (Timer/Counter1 Input Capture Trigger)
PD6	T1 (Timer/Counter1 Clock Input)
PD7	T2 (Timer/Counter2 Clock Input)

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

Port D Data Register - PORTD

Bit	7	6	5	4	3	2	1	0	
$12	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

ITALIAN TECHNOLOGY grifo® abaco bus

Port D Data Direction Register - DDRD

Bit	7	6	5	4	3	2	1	0	
$11	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port D Input Pins Address - PIND

Bit	7	6	5	4	3	2	1	0	
$10	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port D Input Pins address - PIND - is not a register, and this address enables access to the physical value on ea Port D pin. When reading PORTD, the Port D Data Latch is read, and when reading PIND, the logical values present on t 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 co figured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when configur as an input pin the MOS pull up resistor is activated. To switch the pull up resistor off the PDn has to be cleared (zero) the pin has to be configured as an output pin.The port pins are tri-stated when a reset condition becomes active, even if t clock is not running.

Table 33. DDDn Bits on Port D Pins

DDDn	PORTDn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PDn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

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

Alternate Functions of Port D

$\overline{\text{INT0}}$ .. $\overline{\text{INT3}}$ - 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. Tl internal pull up MOS resistors can be activated as described above. See the interrupt description for further details, ar 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 co tents of Timer/Counter1 is transferred to the Timer/Counter1 Input Capture Register. The pin has to be configured as input to serve this function. See the Timer/Counter1 description on how to operate this function. The internal pull up MC resistor can be activated as described above.

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.

Port D Schematics

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

Figure 62. Port D Schematic Diagram (Pins PD0, PD1, PD2 and PD3)

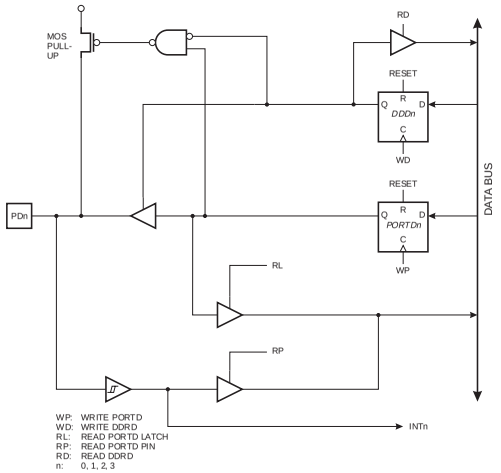
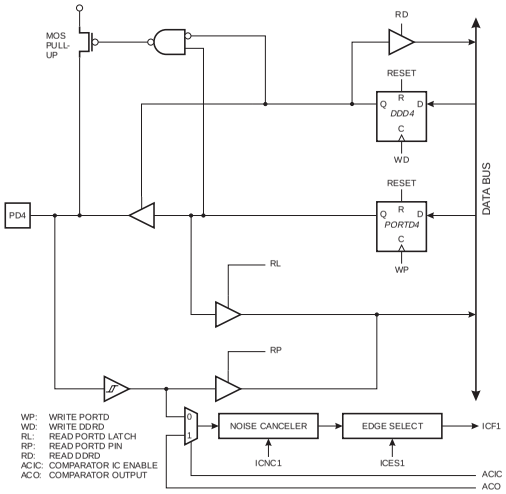


Figure 63. Port D Schematic Diagram (Pin PD4)



grifo® ITALIAN TECHNOLOGY

Figure 64. Port D Schematic Diagram (Pin PD5)

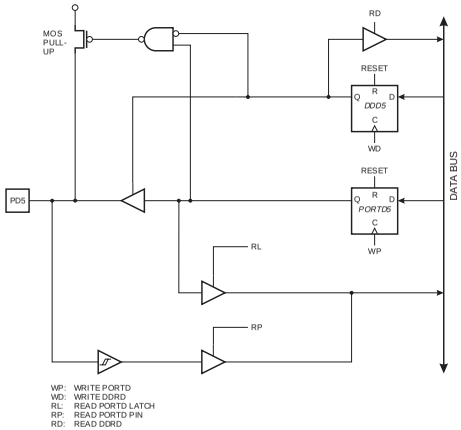
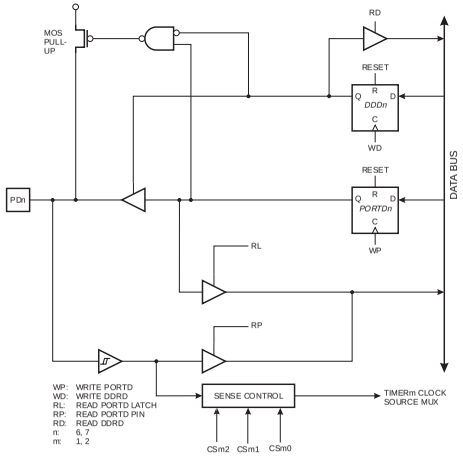


Figure 65. Port D Schematic Diagram (Pins PD6 and PD7)



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($23), Da Direction Register - DDRE, $02($22) and the Port E Input Pins - PINE, $01($21). The Port E Input Pins address is re 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 pu up resistors are activated.

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

Table 34. Port E Pins Alternate Functions

Port Pin	Alternate Function
PE0	PDI/RXD (Programming Data Input or UART Receive Pin)
PE1	PDO/TXD (Programming Data Output or UART Transmit Pin)
PE2	AC+ (Analog Comparator Positive Input)
PE3	AC- (Analog Comparator Negative Input)
PE4	INT4 (External Interrupt4 Input)
PE5	INT5 (External Interrupt5 Input)
PE6	INT6 (External Interrupt6 Input)
PE7	INT7 (External Interrupt7 Input)

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

Port E Data Register - PORTE

Bit	7	6	5	4	3	2	1	0	
$03 ($23)	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	PORTE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port E Data Direction Register - DDRE

Bit	7	6	5	4	3	2	1	0	
$02 ($22)	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	DDRE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port E Input Pins Address - PINE

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	PINE
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

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

Port E as general digital I/O

PEn, General I/O pin: The DDEn bit in the DDRE register selects the direction of this pin. If DDEn is set (one), PEn is co figured as an output pin. If DDEn is cleared (zero), PEn is configured as an input pin. If PEn is set (one) when configured an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off the PEn has to be cleared (zero) or tl pin has to be configured as an output pin.The port pins are tri-stated when a reset condition becomes active, even if tl clock is not running.

Table 35. DDEn Bits on Port E Pins

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

Note: n: 7,6...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 tl ATmega603/103.

RXD, UART Receive Pin. Receive Data (Data input pin for the UART). When the UART receiver is enabled this pin is co figured as an input regardless of the value of DDRD0. When the UART forces this pin to be an input, a logical one 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 tl 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 MC Interrupts can be triggered by low level or positive or negative edge on these pins. The internal pull up MOS resistors c 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 66. Port E Schematic Diagram, Pin PE0

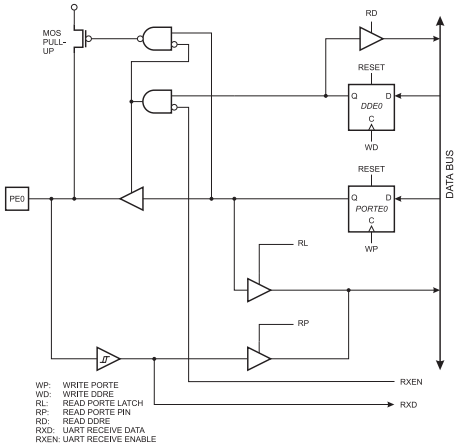
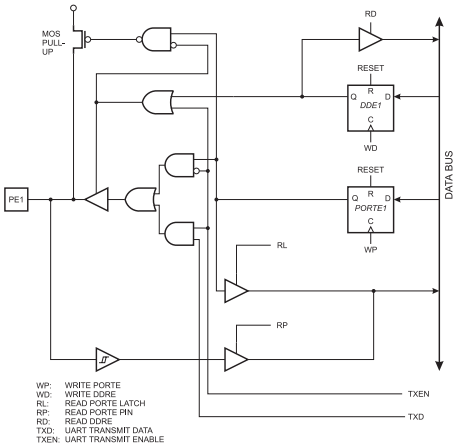


Figure 67. Port E Schematic Diagram (Pin PE1)



grifo® ITALIAN TECHNOLOGY

Figure 68. Port E Schematic Diagram (Pin PE2)

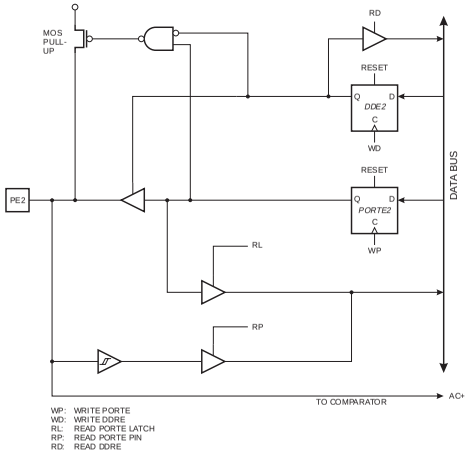


Figure 69. Port E Schematic Diagram (Pin PE3)

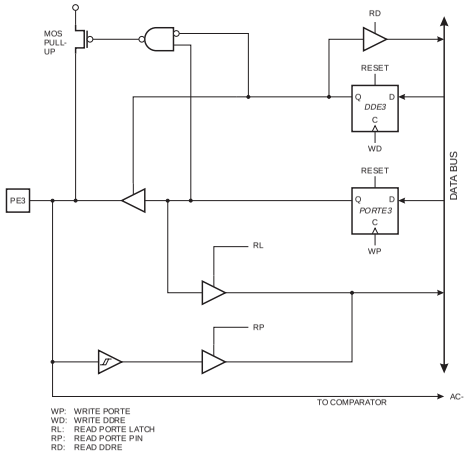
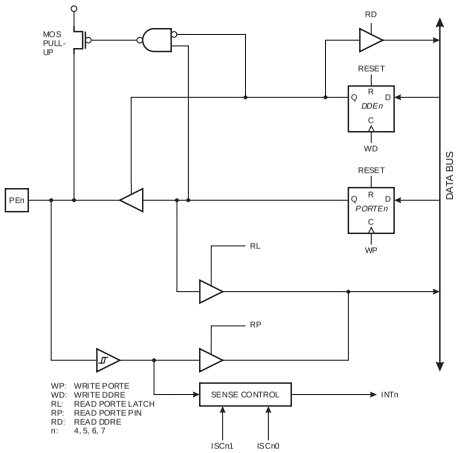


Figure 70. Port E Schematic Diagram (Pins PE4, PE5, PE6 and PE7)



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).

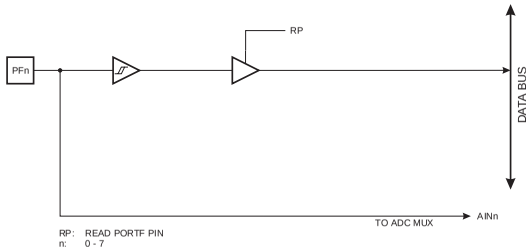
All Port F pins are connected to the analog multiplexer which further is connected to the A/D converter. The digital inp function of Port F can be used together with the A/D converter, allowing the user to use some pins of Port F and digi inputs and other as analog inputs, at the same time.

Port F Input Pins Address - PINF

Bit	7	6	5	4	3	2	1	0	
$00 ($20)	**PINF7**	**PINF6**	**PINF5**	**PINF4**	**PINF3**	**PINF2**	**PINF1**	**PINF0**	**PINF**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

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

Figure 71. Port F Schematic Diagram (Pins PF7 - PF0)



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') obtain the additional features listed in Table 36. The Lock bits can only be erased to '1' with the Chip Erase command..

Table 36. Lock Bit Protection Modes

Memory Lock Bits			Protection Type
Mode	LB1	LB2	
1	1	1	No memory lock features enabled.
2	0	1	Further programming of the Flash and EEPROM is disabled.(1)
3	0	0	Same as mode 2, and verify is also disabled.

Note: 1. In Parallel mode, programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bit

Fuse Bits

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

- When the SPIEN Fuse is programmed ('0'), Serial Program and Data Downloading is enabled. Default value is programmed ('0'). The SPIEN Fuse is not accessible in serial programming mode.
- When EESAVE is programmed, 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

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	page 21
$3E ($5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	page 21
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	page 21
$3C ($5C)	XDIV	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	page 23
$3B ($5B)	RAMPZ	-	-	-	-	-	-	-	RAMPZ0	page 22
$3A ($5A)	EICR	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	page 31
$39 ($59)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	page 30
$38 ($58)	EIFR	INTF7	INTF6	INTF5	INTF4	-	-	-	-	page 30
$37 ($57)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	page 31
$36 ($56)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	page 32
$35 ($55)	MCUCR	SRE	SRW	SE	SM1	SM0	-	-	-	page 22
$34 ($54)	MCUSR	-	-	-	-	-	-	EXTRF	PORF	page 29
$33 ($53)	TCCR0	-	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	page 38
$32 ($52)	TCNT0	Timer/Counter0 (8 Bit)								page 39
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register								page 40
$30 ($50)	ASSR	-	-	-	-	AS0	TCN0UB	OCR0UB	TCR0UB	page 41
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10	page 45
$2E ($4E)	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	page 46
$2D ($4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								page 47
$2C ($4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								page 47
$2B ($4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								page 48
$2A ($4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								page 48
$29 ($49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								page 48
$28 ($48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								page 48
$27 ($47)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								page 48
$26 ($46)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								page 48
$25 ($45)	TCCR2	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	page 38
$24 ($44)	TCNT2	Timer/Counter2 (8 Bit)								page 39
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								page 40
$21 ($47)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	page 51
$1F ($3F)	EEARH	-	-	-	-	EEAR11	EEAR10	EEAR9	EEAR8	page 52
$1E ($3E)	EEARL	EEPROM Address Register L								page 52
$1D ($3D)	EEDR	EEPROM Data Register								page 53
$1C ($3C)	EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	page 53
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	page 75
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	page 75
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	page 75
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	page 77
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	page 77
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 77
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	page 82
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	page 83
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	page 84
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	page 84
$0F ($2F)	SPDR	SPI Data Register								page 58
$0E ($2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	page 58
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	page 57
$0C ($2C)	UDR	UART I/O Data Register								page 62
$0B ($2B)	USR	RXC	TXC	UDRE	FE	OR	-	-	-	page 62
$0A ($2A)	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	page 63
$09 ($29)	UBRR	UART Baud Rate Register								page 65
$08 ($28)	ACSR	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	page 65
$07 ($27)	ADMUX	-	-	-	-	-	MUX2	MUX1	MUX0	page 69
$06 ($26)	ADCSR	ADES	ABSY	ADRF	ADIF	ADIE	ADPS2	ADPS1	ADPS0	page 70
$05 ($25)	ADCH	-	-	-	-	-	-	ADC9	ADC8	page 71
$04 ($24)	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	page 71
$03 ($23)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	page 87
$02 ($22)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	page 87
$01 ($21)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	page 87
$00 ($20)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	page 91

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 and SBI instructions work with registers $00 to $1F only.

ITALIAN TECHNOLOGY

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd - Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd - K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
ELPM(1)		Extended Load Program Memory	R0 ← (Z+RAMPZ)	None	3
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	3
WDR		Watchdog Reset	(see specific descr. for WD timer)	None	1

Note: 1. Not in ATmega603.

APPENDIX C: ELECTRIC DIAGRAMS

In this appendix are available some electric diagrams of the most frequently used **GPC® AM4** interfaces. All these interface can be yourself produced and some of them are standard **grifo®** cards and, if required, they can be directly ordered.

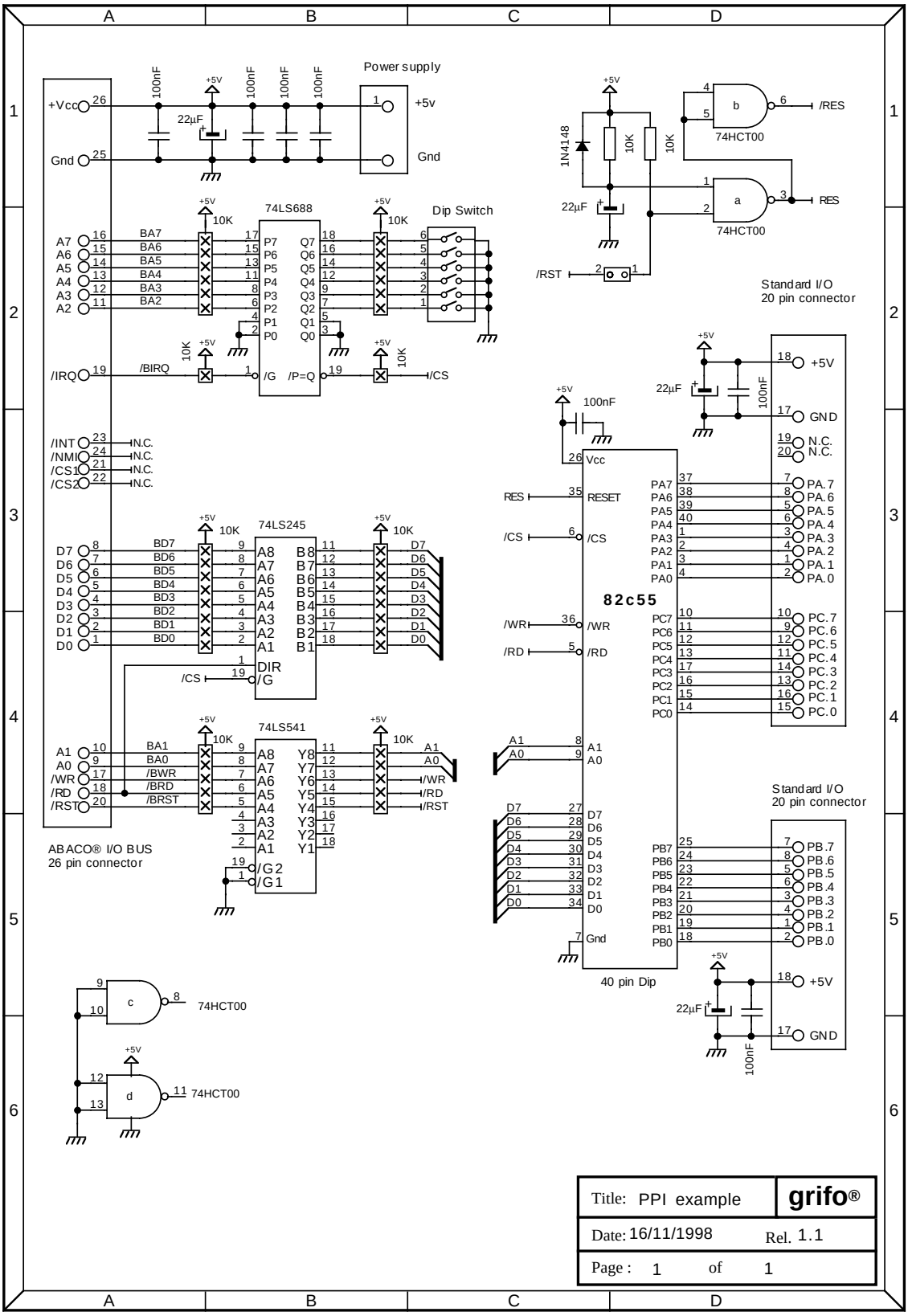


FIGURE C1: PPI EXPANSION ELECTRIC DIAGRAM

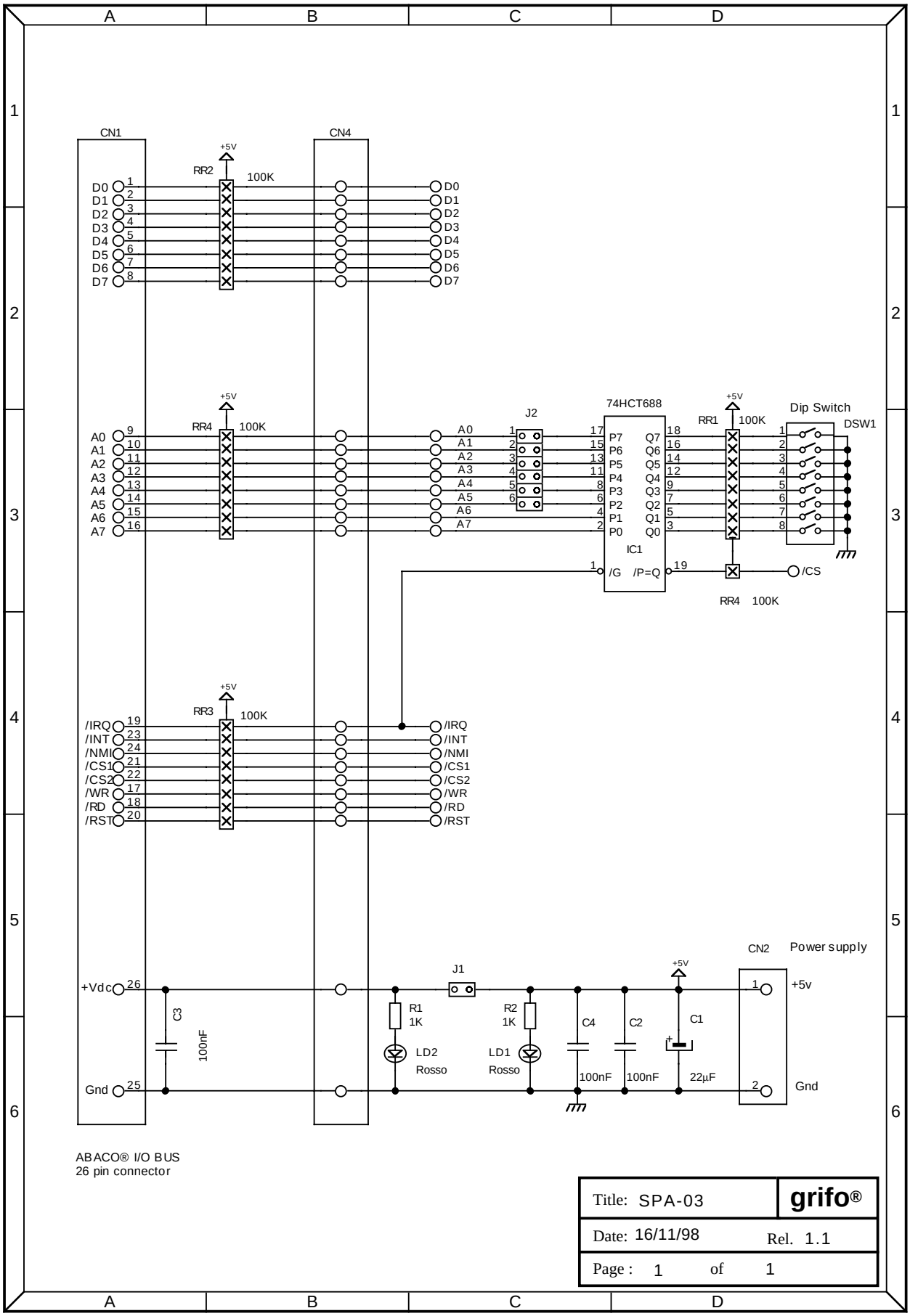


FIGURE C2: SPA 03 ELECTRIC DIAGRAM

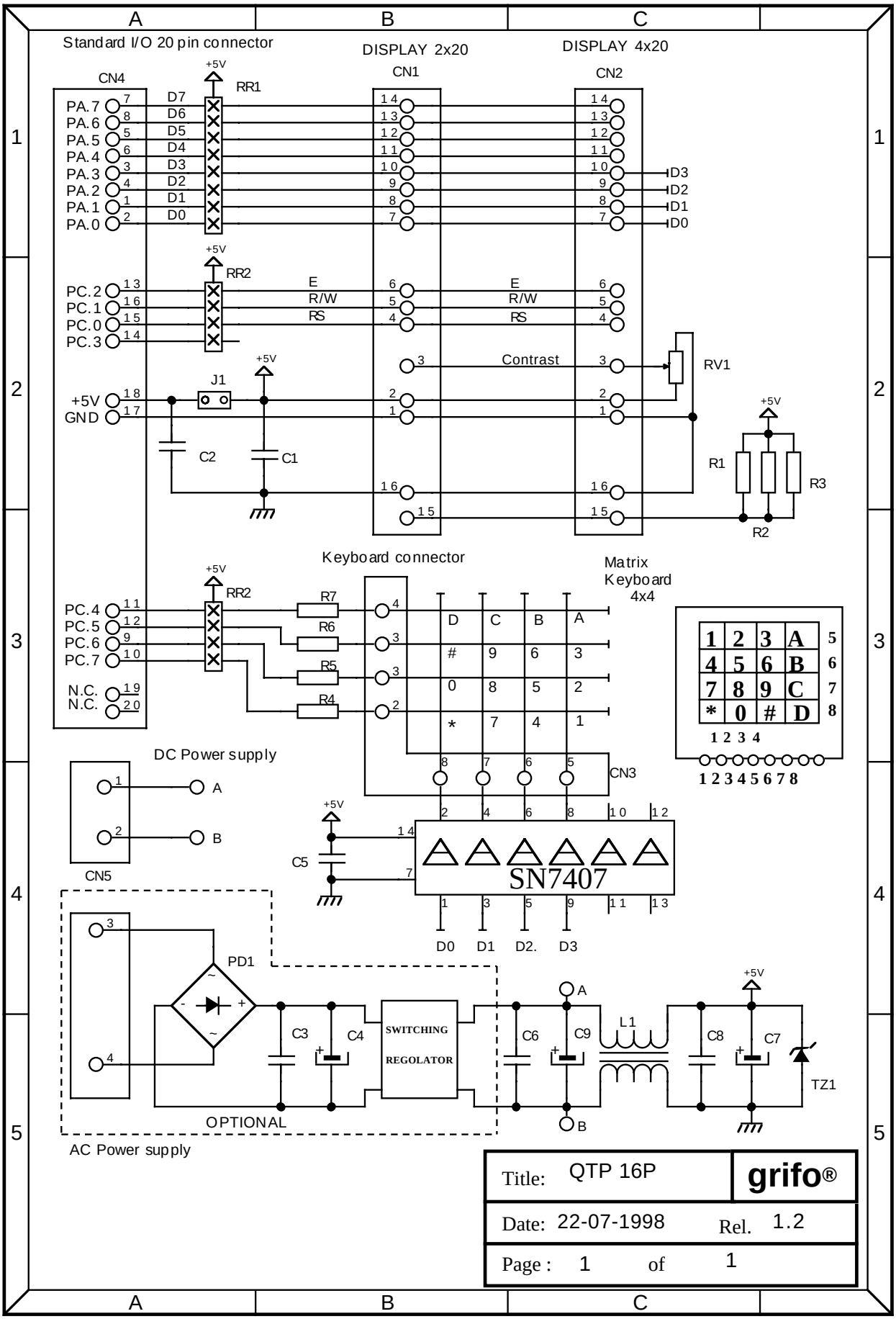


FIGURE C3: QTP 16P ELECTRIC DIAGRAM

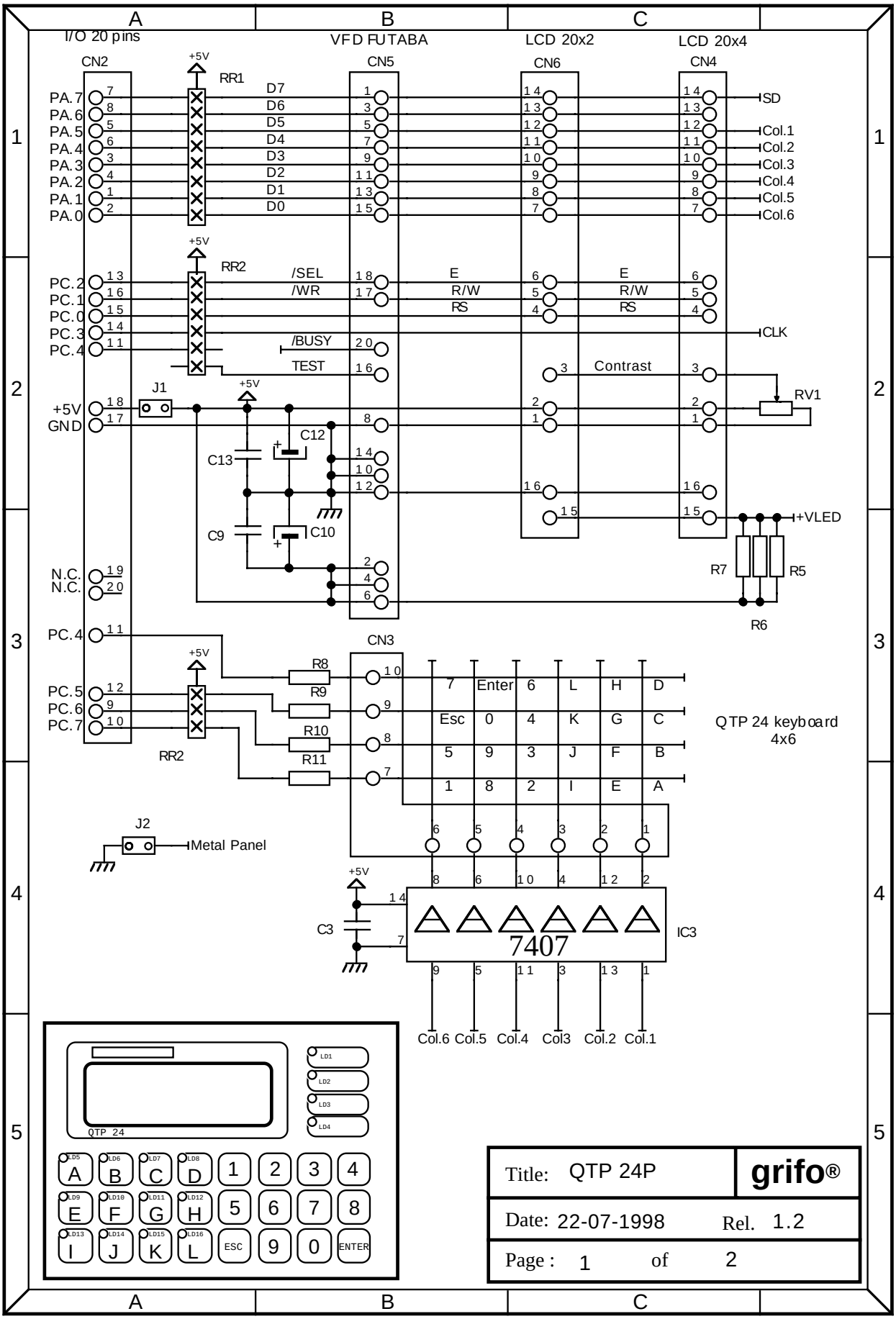


FIGURE C4: QTP 24P ELECTRIC DIAGRAM (1 OF 2)

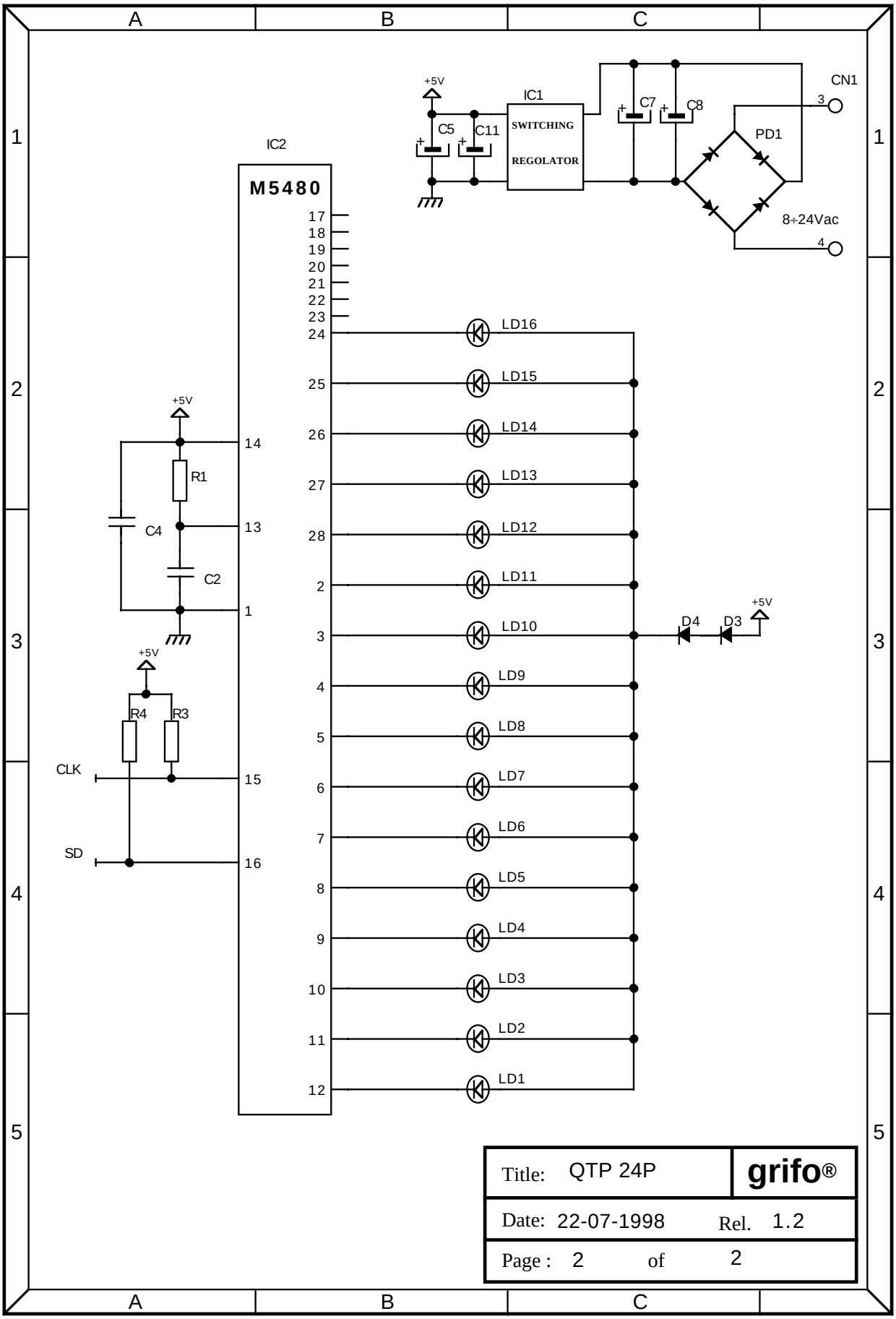


FIGURE C5: QTP 24P ELECTRIC DIAGRAM (2 OF 2)

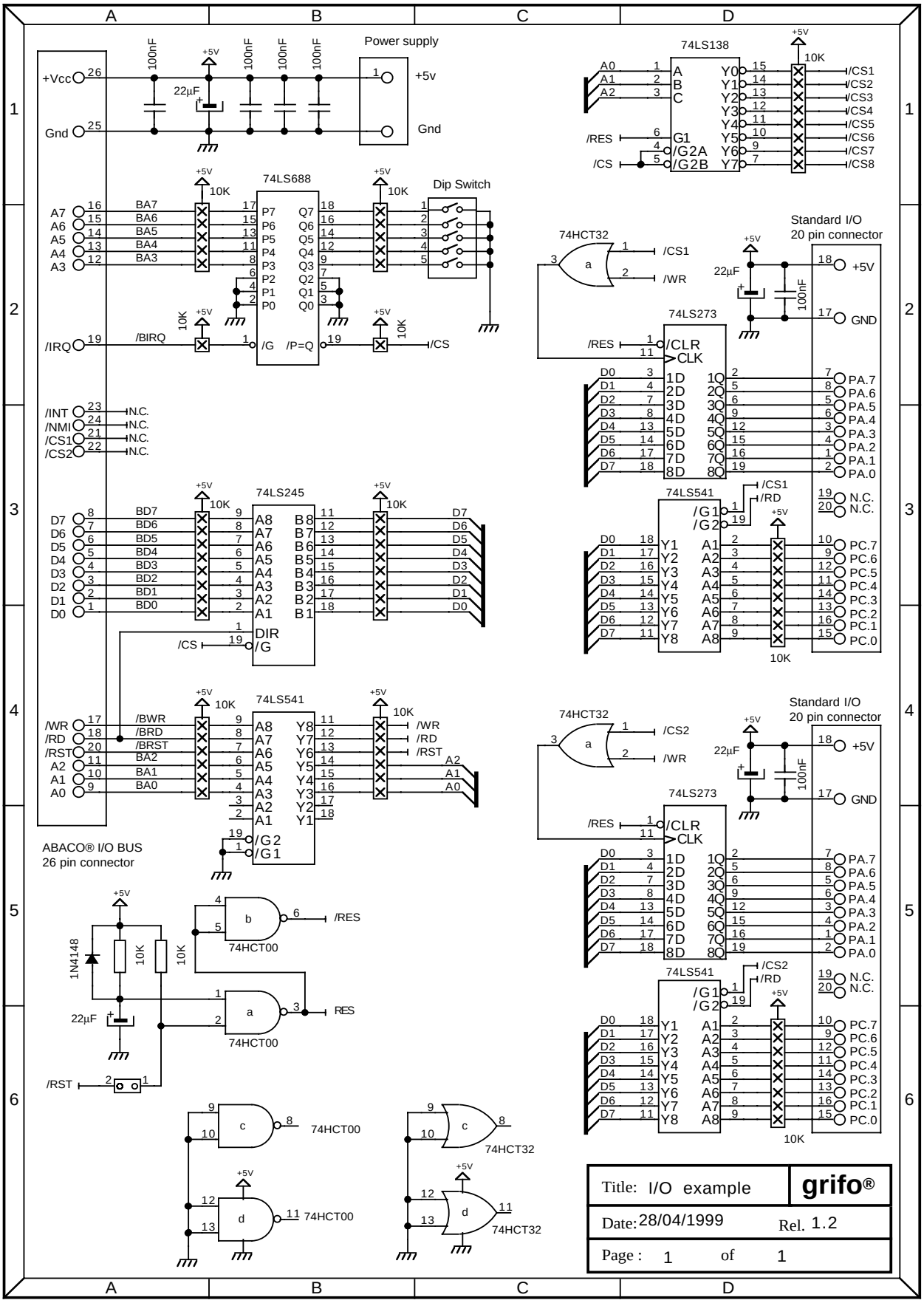


FIGURE C6: ABACO® I/O BUS INPUT OUTPUT ELECTRIC DIAGRAM

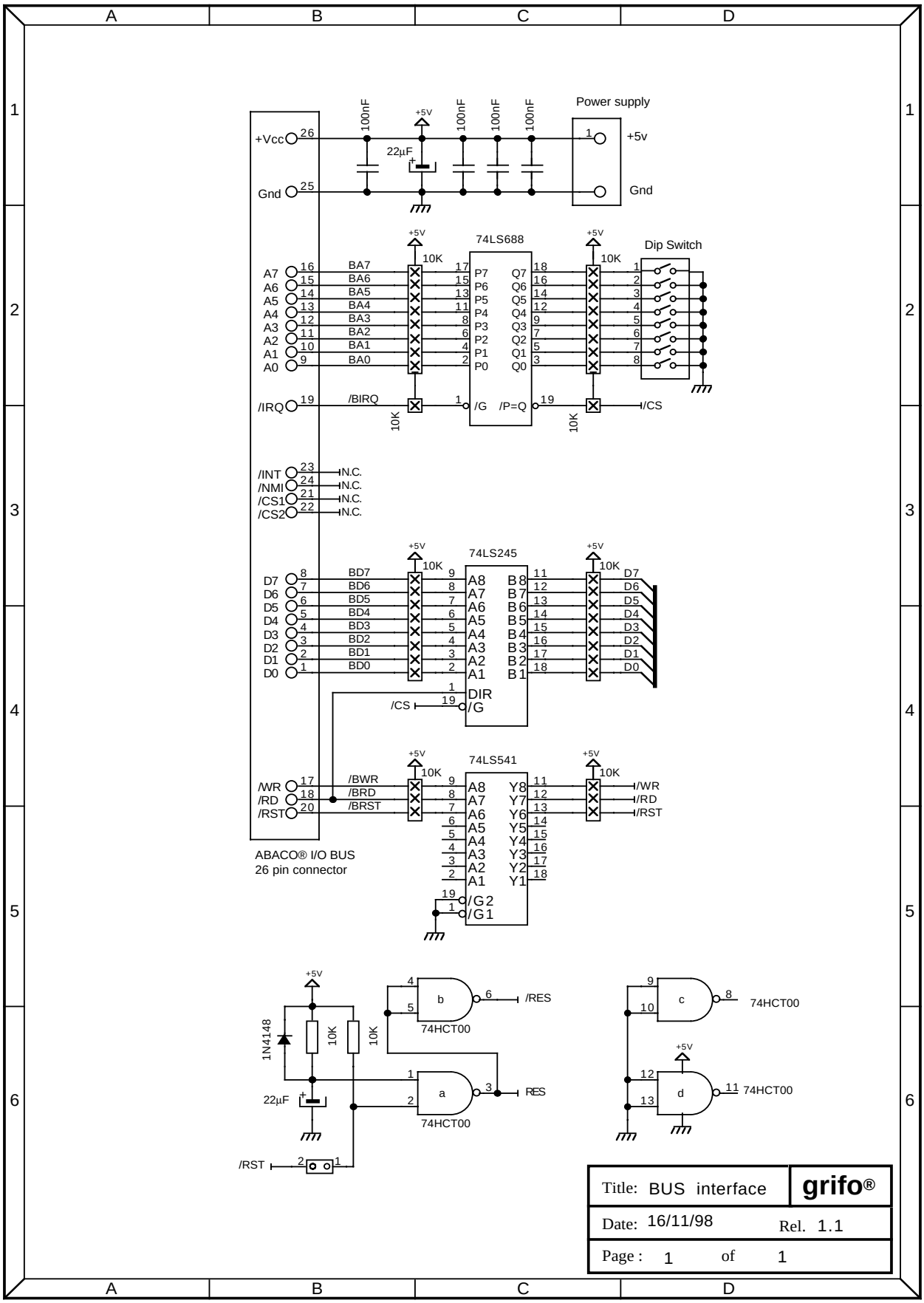


FIGURE C7: BUS INTERFACE ELECTRIC DIAGRAM

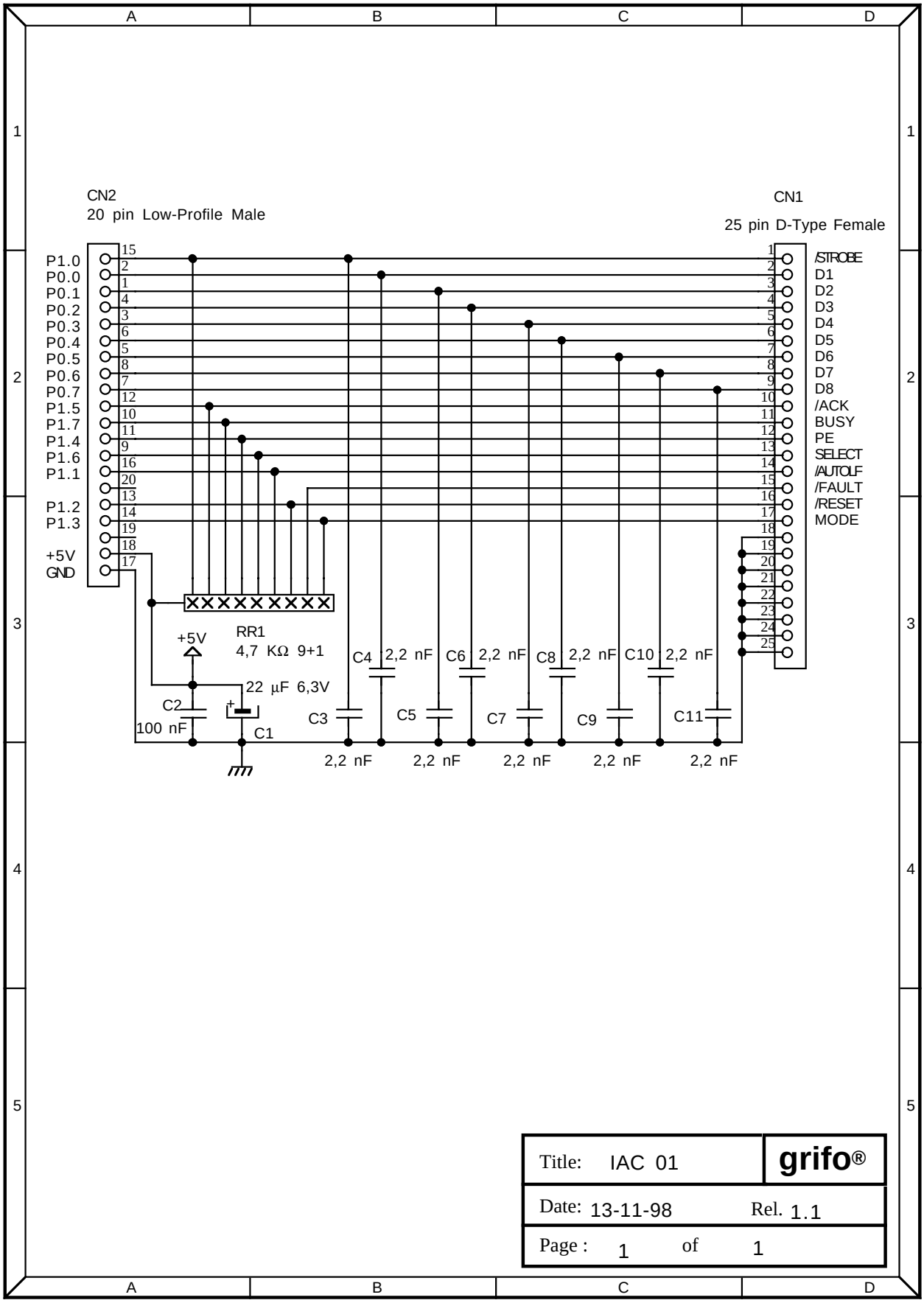


FIGURE C8: IAC 01 ELECTRIC DIAGRAM

APPENDIX D: ALPHABETICAL INDEX

A

A/D
  conversion time **8**
  diagram **21**
  inpedance **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**
Istrunctions **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**