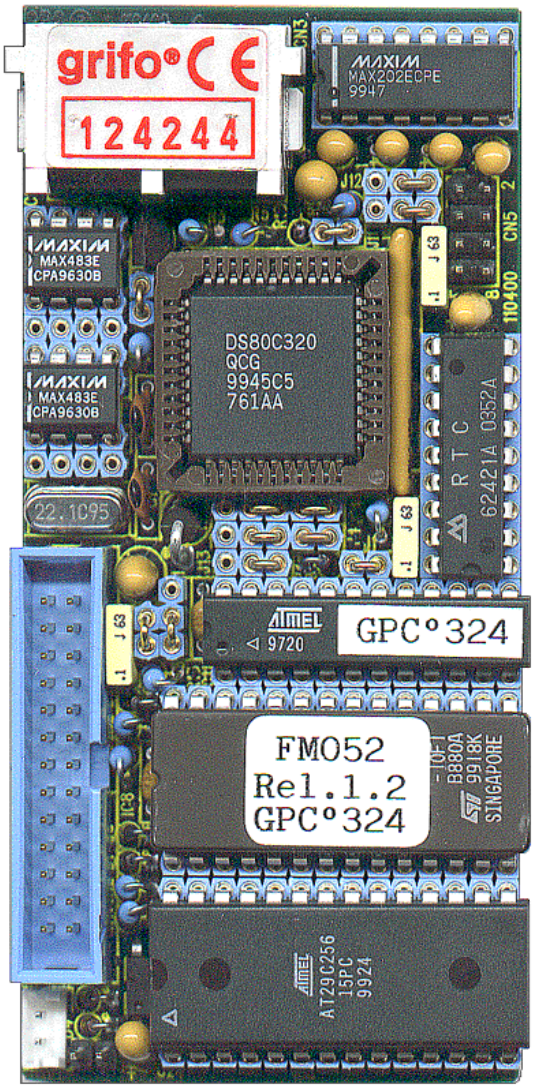
GPC® 324

General Purpose Controller 80C32,320 etc. 4 type

TECHNICAL MANUAL



grifo®
ITALIAN TECHNOLOGY

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

http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (a. r.) FAX: +39 051 893.661

GPC® 324 Edition 5.20 *Rel. 05 July 2000*

GPC® 324

General Purpose Controller 80C32,320 etc. 4 type

TECHNICAL MANUAL

Intelligent module of the **ABACO® BLOCK** series, 100x50 mm size. Optional plastic mount for connection to **DIN 46277-1** and **DIN 46277-3** Ω rails. **CPU 80c32** or **80c320**,at **22 MHz** and all the compatibles models. Moreover the In System Programming of the **PHILIPS 89CRx+/2** is completely supported. Maximun memory configuration of 104K Bytes: **32K SRAM**, socket for **32K EPROM**, socket for **32K** EPROM, SRAM, **EEPROM** or **FLASH EPROM** and serial **EEPROM**, from 256 to 8192 Bytes. Complete **Real Time Clock** capable to generate **INT**errupt. **Back up** circuit for **SRAM** and **RTC**, through on board and external **LITHIUM** battery. **Watch dog** software and hardware settable. **5 TTL I/O** lines, three 16 Bits **timer counters**; jumper for **RUN/DEBUG** mode. 2 **RS232** serial lines, one configurable in **RS422**, **RS485** or **current loop**. Standard 26 pins expansion connector for **ABACO® I/O BUS** interface. Numerous interrupt sources, including an efficient **Power Failure** circuitry. Single **5Vdc**, **115 mA** power supply, with different power saving modes. On board logic protected against transients by **TransZorb™**. Wide range of developement software such as: Monitor Debugger (**MDP**, **FMO52**, **NOICE**); Assembler (**MCA51**); **GET 51**; C compilers (**MCC51**, **HTC51**, **SYS51CW**, **DDS Micro C51**); BASIC (**BASIC 324**,**BXC51**, **BASCOM 8051**); PASCAL compiler (**SYS51PW**); etc.

grifo®
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
E-mail: grifo@grifo.it
http://www.grifo.it http://www.grifo.com
Tel. +39 051 892.052 (a. r.) FAX: +39 051 893.661

DOCUMENTATION COPYRIGHT BY grifo®, ALL RIGHTS RESERVED

IMPORTANT

Although all the information contained herein have been carefully verified, **grifo**® assumes no responsability for errors that might appear in this document, or for damage to things or persons resulting from technical errors, omission and improper use of this manual and of the related software and hardware.
grifo® reserves the right to change the contents and form of this document, as well as the features and specification of its products at any time, without prior notice, to obtain always the best product.
For specific informations on the components mounted on the card, please refer to the Data Book of the builder or second sources.

SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

Attention: Generic danger

Attention: High voltage

Trade Marks

abaco®, **GPC**®, **grifo**® : are trade marks of **grifo**®.
Other Product and Company names listed, are trade marks of their respective companies.

GENERAL INDEX

FIGURE INDEX

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® 324** card release **110400** 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 near the CN5 connector and IC14 socket on the component side).

GENERAL FEATURES

The **GPC® 324** belongs to the CPUs **4 Serie** 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® 324** module can be secured in a plastic mount for connection to **Omega** rails **DIN 46277-1** and **DIN 46277-3**, thereby dispensing with the need of a rack and allowing a less costly mounting direct to the electrical control panel. Thanks to this small size, the **GPC® 324** put into the same plastic rails that contains the peripheral I/O card, i.e **ZBR xxx**, **ZBT xxx**, forming in this way an unique **BLOCK** element. The **GPC® 324** can also be mounted as a macro CPU module on a card developed by the end user, in **Piggy Back** (stack through) mode; cards as **SPA 03** and **SPA 04** can be used to realize a quick prototype in this modality. Finally the presence of the **ABACO® I/O BUS** allows the connection to cards as **CAN 14**, **ADC 812**, **DAC 212**, etc, and through **ABB 03** and **ABB 05** mother boards even to **ABACO® BUS** peripheral cards.
The most interesting characteristic of **GPC® 324** module is that it can be equipped with a wide range of **µP**. It is in fact possible to get it by the standard **80C32**; by the fast **DALLAS 80C320**, by the new **PHILIPS 89CRx+/2** and many of the compatible models. The characteristics of the **GPC® 324** module remain basically the same, but its performance changes according to the built in **µP**.
At present there are some developping software tools which allow the card to be used as developping system of itself both in asselmber and high level languages. Noteworthy among these are the numerous **C** compilers, **PASCAL**, the powerfull **BASIC** compilers and the handy **BASIC 324**. The latter is compatible with the widespread **MCS® BASIC 52** of **INTEL** to which have been added new commands. Among these new ones is dutiful to mention some of those referred to the serial **EEPROM** and 2nd serial line. By adding an external **82c55** a **Fluorescent** or **LCD displays** and a **matrix keyboard** are managed; for an immediate use of this new commands, **KDx x24** boards are available, or if you need a finished object, there is the operator panel **QTP xxP**. This operator panel, offered in the open frame version, bears the same aesthetic as **QTP xx**, but, as the local intelligence is not furnished, it is driven directly by **GPC® 324**, allowing a notable cost reduction.
The **BASIC 324** and **BASCOM 8051 + FMO52** affords truly notable debug facilities, and allows to program directly the on board **EEPROM** or **FLASH EPROM** with the user program.

- Intelligent module of the **ABACO® BLOCK** series, 100x50 mm size.
- Optional plastic mount for connection to **DIN 46277-1** and **DIN 46277-3** Ω rails.
- **CPU 80c32** or **80c320**,at **22 MHz** and all the compatibles models. Moreover the In System Programming of the **PHILIPS 89CRx+/2** is completely supported.
- Maximun memory configuration of 104K Bytes: **32K SRAM**, socket for **32K EPROM**, socket for **32K** EPROM, SRAM, **EEPROM** or **FLASH EPROM** and serial **EEPROM**, from 256 to 8192 Bytes.
- Complete **Real Time Clock** capable to generate **INT**errupt.
- **Back up** circuit for **SRAM** and **RTC**, through on board and external **LITHIUM** battery.
- **Watch dog** software and hardware settable.
- **5 TTL I/O** lines, three 16 Bits **timer counters**; jumper for **RUN/DEBUG** mode.
- 2 **RS232** serial lines, one configurable in **RS422**, **RS485** or **current loop**.
- Standard 26 pins expansion connector for **ABACO® I/O BUS** interface.
- Numerous interrupt sources, including an efficient **Power Failure** circuitry.
- Single **5Vdc**, **115 mA**power supply, with different power saving modes.
- On board logic protected against transients by **TransZorb™**
- Wide range of developement software such as: Monitor Debugger (**MDP**, **FMO52**, **NOICE**); Assembler (**MCA51**); **GET 51**; C compilers (**MCC51**, **HTC51**, **SYS51CW**,

DDS Micro C51); BASIC (**BASIC 324**,**BXC51**, **BASCOM 8051**); PASCAL compiler (**SYS51PW**); 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 2.

CPU

The **GPC® 324** can use many of '51 microprocessors family as 80C32, 80C52, 87C52, 89C52 (from INTEL and other secound sources), 89S8252 (from ATMEL), 89CRx+/2 (from PHILIPS) 80C320, 87C320 (manufactured by DALLAS) and all the interchangeable ones. These 8 bit microprocessors are code compatible with the world wide used 8051 INTEL and they have an extended instruction set, fast execution time, easy use of all kind of memory and an efficient interrupt management. The most important features of the described microprocessor, are:

Microprocessor	**80C32**	**89S8252**	**89CRx+/2**	**80C320**
Data BUS width	8	8	8	8
Clock system cicle	12	12	6	4
Internal RAM (bytes)	256	256	256	256
Internal ROM (kbytes)	8	8	64	8
Internal EEPROM (kbytes)	0	2	0	0
External code area (kbytes)	64	64	64	64
External data area (kbytes)	64	64	64	64
I/O ports	4	4	4	4
16 bits Timer/Counters	3	3	3	3
Interrupt sources	6	9	7	13
Interrupt priority level	2	2	4	3
A/Syncronous serial line	1	1	1	2
Idle mode or Power down mode	Yes	Yes	Yes	Yes
Power monitor and control section	No	Yes	No	Yes
Internal watch dog timer	No	Yes	No	Yes
In system programming	No	Yes	Yes	No
In application programming	No	No	Yes	No

FIGURE 1: MICROPROCESSOR FEATURES

For further information, please refer to specific documentation of the manufacturing company or to appendix B of this manual. Please remember that the previous table describes the general microprocessor features and some of them can be not supported by the card.
The user must specify the requested microprocessor in the order phase and in absence of any indication the card is supplied in its default condition with 80C32. The version with DALLAS 80C320 is instead denoted by the D suffix = **GPC® 324D**.

ABACO® I/O BUS

One of the most important features of **GPC® 324** is its possibility to be interfaced to many others industrial cards.Thanks to its standard **ABACO® I/O BUS** 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 (A/D), cards for analog signals generation (D/A), cards for digital I/O signals management, cards with timers and counters, cards for temperature controls and even custom boards designed to satisfy specific needs of the end user.
Using **ABB 03** or **ABB 05** mother boards it is possible manage all the BUS **ABACO®** single EURO cards. So **GPC® 324** 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.

SERIAL COMMUNICATION

An hardware serial line is always available on **GPC® 324** (named serial **A**) and a second serial line (named serial **B**) is managed as below described:

µP 80C32: software serial line driven through two microprocessor I/O lines;
µP 80C320: hardware serial line driven by proper hardware section;

The hardware serial communication lines are completely software configurable for physical protocol and by simply programming some microprocessor registers, the user can set the baud rate, stop bits number, lenght of character and parity. For software serial line the physical protocol is completely defined by the software procedures. Some software tools (**BASIC 324**, **BASCOM 8051**, etc.) directly manages the software lines through high level instructions.
The serial line B is always RS 232 buffered, while the serial line A can be configured in RS 232, RS 422, RS 485 or passive current loop thanks to confortable on board jumpers.
For further information about serial communication please refer to chapter "SERIAL COMMUNICATION SELECTION" and to the technical documentation of the microprocessor.

CONTROL LOGIC

The addresses of all peripheral device's registers and of memory devices on **GPC® 324** are assigned from a specific control logic that allocates all these devices in the microprocessor addressing space. For further information please refer to paragraph "I/O ADDRESSES" and "MEMORY ADDRESSES" of this manual.

RESET CONTACT

P1 reset contact of the **GPC® 324** allows the user to reset the board and restarting it in a general clearing condition. The main purpose of this contact is to come out of infinite loop conditions, useful especially during debug or to grant a particular initial stat. Please see figure 7 for an easy localization of this contact.

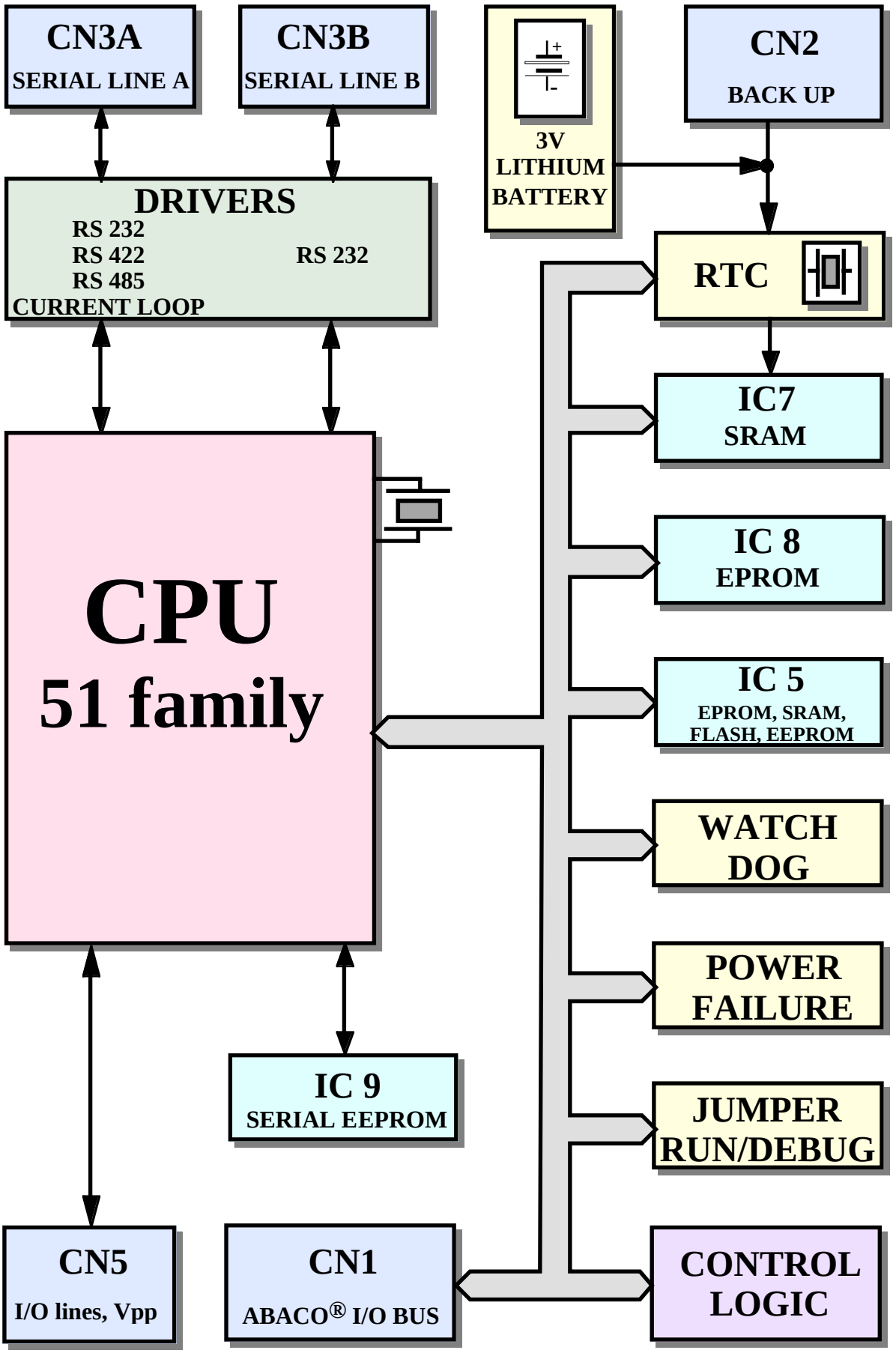


FIGURE 2: BLOCK DIAGRAM

PERIPHERAL DEVICES

GPC® 324 is the right card to solve many control problems in automation fields, in fact it is equipped with some peripheral components that facilitate the connection and the management to external system. These peripherals are:

- **external watch dog** it is an astable circuit that can reset the card at programmable time intervals of about 1.4 sec, if not retriggered. By software the user can retrigger the circuit to prevent card reset, using specific register allocated in microprocessor addressing space. The external watch dog is used when the user want to exit from endless loops or to reset anomalous conditions not estimated by application program.

- **serial EEPROM** With the IC9 EEPROM module (range 256÷8192 bytes), there is the possibility to keep data also when power supply is failed; in this way the card is always able to maintain parameters, few logged data, system status and configuration, etc. without using backed RAM. This component has a default size of 512 bytes.

- **board configuration** Jumper J1 has been introduced expressely to make the board and in particualr the application program confugrable. The possibility to read by software the status of this jumper gives the user the ability to manage several working conditions through an unique program, without no need to employ more input singnals (typical applications are: language choice, program parameters definition, operational mode selection, etc.). Some software tools developed for the **GPC® 324** board use jumper J1 to select between the operation modalities RUN and DEBUG, as described in the manuals of the tools themselves.

- **real time clock** **GPC® 324** 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 addressing space by the control logic.The RTC section can generate interrupts at a software programmable rate, for diverting the CPU from its normal tasks or awakening it from one of its low consumption working modes.

- **I/O lines** Five lines of the four 8 bits ports are connected to CN5 connector. The lines direction is software settable at bit level and interrupts can be generated. In this way an external status can obtain CPU control in any condition, with a fast response time. The ports are completely driven by software by programming the proper microprocessor internal registers.

For further information about peripheral devices please refer to chapter "PERIPHERAL DEVICE SOFTWARE DESCRIPTION" and APPENDIX B.

MEMORY DEVICES

On the card can be mounted 104K of memory divided with **32K** EPROM on socket, **32K** SRAM, **32K** SRAM, EPROM, EEPROM or FLASH EPROM on socket and up to **8K** of serial EEPROM. The **GPC® 324** memory configuration must be chosen considering the application to realize or the specific requirements of the user. Normally the card is equipped with 32K byte of static RAM and 512 bytes of serial EEPROM and all different configurations must be specified from the user, at the moment of the order. With the on board back up circuit there is the possibility to keep 32K SRAM data (IC7), 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 based on the on board and external battery, as described in "BACK UP" paragraph. By mounting a proper device on IC5 socket the user can improve the on board backed memory capacity.
The addressing of memory devices is controlled by a specific control logic, that provides to allocate the devices in the microprocessor address space, this control logic automatically manages the different addressing mode and it satisfy the requests of each **GPC® 324** software tools.
For further information about memory configuration, sockets description and jumpers connection, please refer to paragraphs "MEMORY ADDRESSES" and "MEMORY SELECTION".

CLOCK

GPC® 184 is provided with a circuitery that generates the CPU clock frequence (22.1184 MHz); this frequence is used also to generate the frequencies needed to the other sections of the board (Timer/ Counters, serial lines, etc.). On some microprocessors it is possible to divide the clock frequency by software to reduce the power consumption.

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 and idle mode are available) and for increasing the electrical noise immunity. An intersting power failure circuitry capable to detect the imminent power black out is installed on the board, so it can start a software intervent through a generated interrupt. Please remember that on board there is a protection circuit aganist voltage peaks by **TransZorb™**.

TECHNICAL FEATURES

GENERAL FEATURES

Devices:
5 programmable TTL input/output lines
3 timer counters (16 bits)
1 bidirectional RS 232 serial line (software/hardware)
1 bidirectional RS 232, RS 422, RS 485, current loop serial line
1 watch dog
1 local contact for reset
1 software readable user inputs
1 **ABACO® I/O BUS** expansion interface
1 backed Real Time Clock
1 power failure circuit

Memory:
IC 8: 32K x 8 EPROM
IC 7: 32K x 8 SRAM
IC 5: 32K x 8 SRAM, EPROM, EEPROM, FLASH EPROM
IC 9: serial EEPROM from 256 bytes to 8192 bytes

Memories acces time: 70 nsec

CPU:
INTEL 80C32 and compatible ones
ATMEL 89S8252 and compatible ones
PHILIPS 89CRx+/2 and compatible ones
DALLAS 80C320 and compatible ones

Clock Frequency: 22.1184 MHz

External watch dog reset time: from 940 msec to 2060 msec (typical 1420 msec)

PHYSICAL FEATURES

Size:
100 x 50 x 25 mm (without container)
110 x 60 x 60 mm (with plastic container)

Weight:
75 g (without container)
130 g (with plastic container)

Connectors:
CN1: 26 pins, male, vertical, low profile connector
CN2: 2 pins, male, vertical, low profile connector
CN3A: 6 pins PLUG connector
CN3B: 6 pins PLUG connector
CN5: 4+4 pins, male, vertical, strip connector

Temperature range: 0÷50 °C

Relative humidity: 20%÷90% (without condense)

ELECTRIC FEATURES

Power supply voltage:	+5 Vdc
Consumption on 5 Vdc:	115 mA (default configuration) 160 mA (full and higher configuration)
On board back up battery:	3.0 Vdc; 180 mAh
External back up battery:	3.6÷5 Vdc
Back up current:	4.2 µA (on board battery) 5.5 µA (external 3.6 V battery)
RS 422, RS 485 termination network:	line termination = 120 Ω Positive pull-up resistor = 3.3 KΩ Negative pull-down resistor = 3.3 KΩ
Power failure threshold:	52 mV before reset activation

FIGURE 3: CARD PHOTO

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, jumpers and some explanatory diagrams.

CONNECTIONS

The **GPC®324** module has 5 connectors that can be linkeded to other devices or directly to the field, according to system requirements. In this paragraph there are connectors pin out, a short signals description (including the signals direction), connectors location (see figure 7) and some electrical diagrams that shows the on board circuit of each connector.

CN2 - EXTERNAL BACK UP BATTERY CONNECTOR

CN2 is a 2 pins, male, vertical, low profile connector with 2.54 mm pitch.
Through CN2 the user back up the IC7 SRAM and the real time clock through an external battery when the power supply is switched off (for further information please refer to paragraphs "ELECTRIC FEATURES", "BACK UP" and "MEMORY SELECTION").

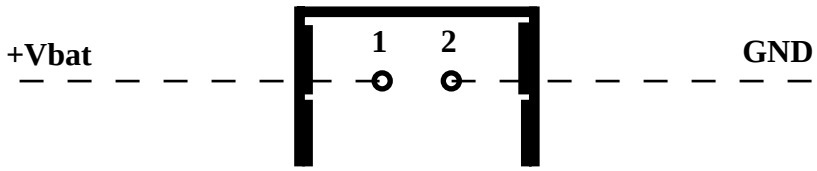


FIGURE 4: CN2 - EXTERNAL BACK UP BATTERY CONNECTOR

Signals description:

+Vbat	=	I	- External back up battery positive pin
GND	=		- External back up battery negative pin

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 some of the numerous **grifo®** boards,both intelligent and not. For example the user can directly use cards for analog signals acquisition (A/D), cards for analog signals generation (D/A), cards for digital I/O signals management, cards with timers and counters, cards for temperature controls, etc. All this connector signals are at TTL level.

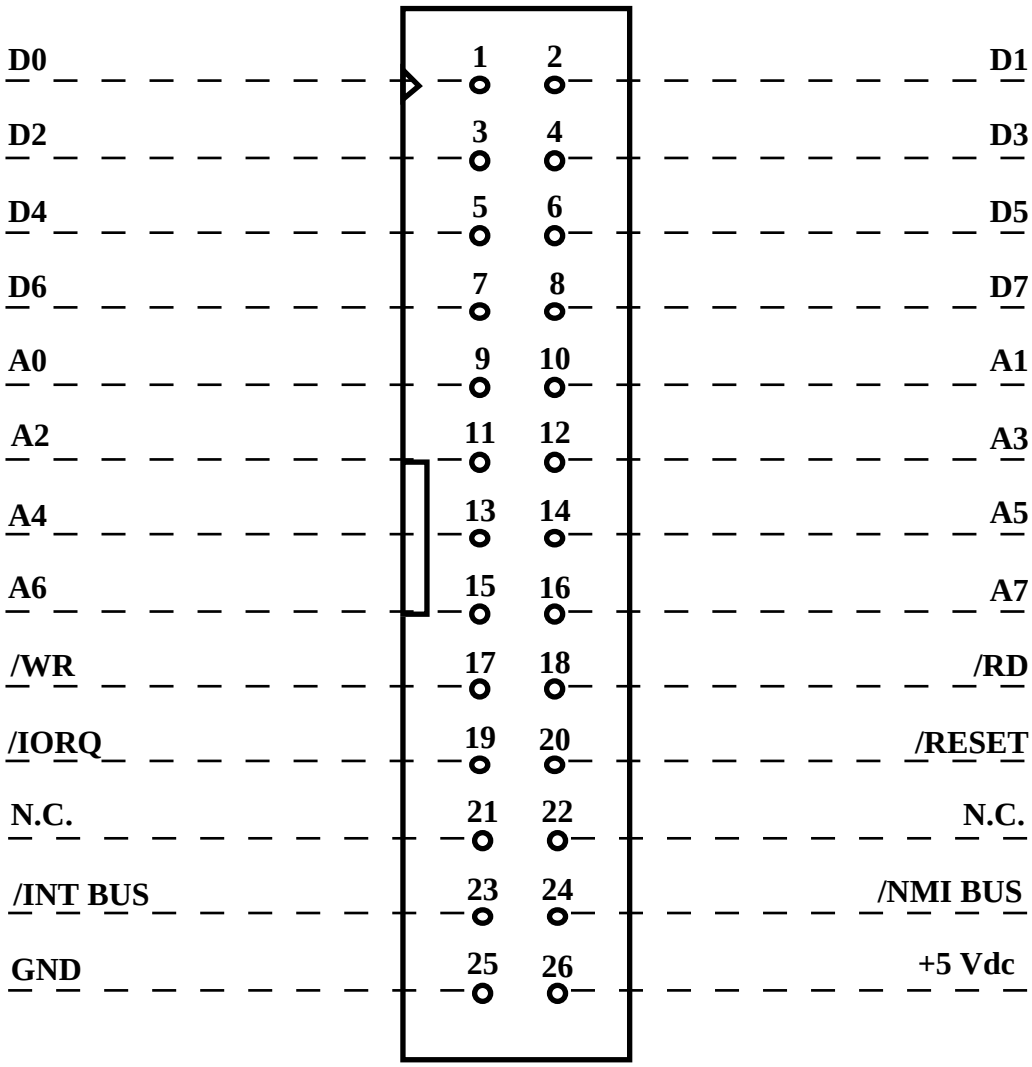


FIGURE 5: CN1 - ABACO® I/O BUS CONNECTOR

Signals description:

A0÷A7	= O	- Address BUS.
D0÷D7	= I/O	- Data BUS.
/INT BUS	= I	- Interrupt request (open collector type).
/NMIBUS	= I	- Non mascable interrupt.
/IORQ	= O	- Input output request.
/RD	= O	- Read cycle status.
/WR	= O	- Write cycle status.
/RESET	= O	- Reset.
+5 Vdc	= I	- +5 Vdc power supply.
GND	=	- Ground signal.
N.C.	=	- Not connected.

CN3A - SERIAL LINE A CONNECTOR

CN3A is a 6 pins, female PLUG connector.
On CN3A are available the buffered signals for RS 232, RS 422, RS 485, current loop serial line A that is phisically connected to the hardware serial line 0 of the microprocessor. The electric protocol follows the CCITT normative and all the signals are placed in order to reduce interference and electrical noise and in order to simplify connection with other systems.

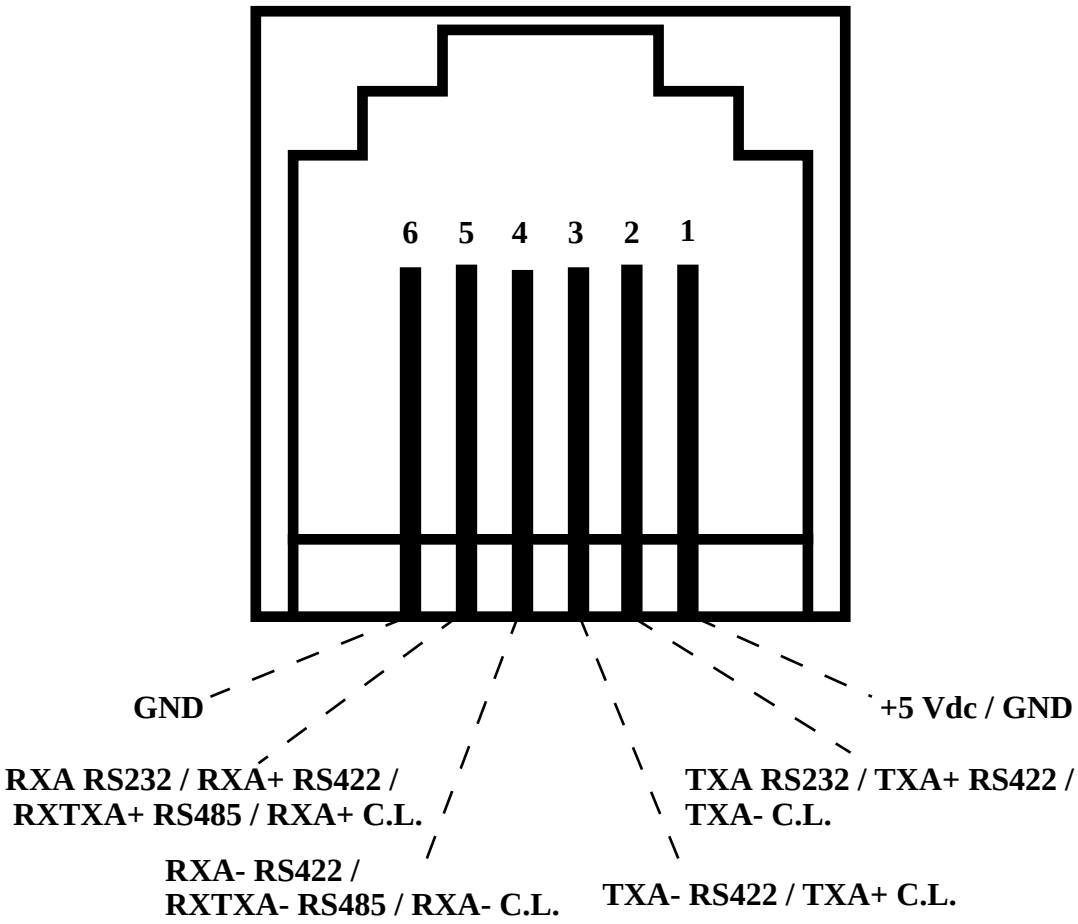


FIGURE 6: CN3A - SERIAL LINE A CONNECTOR

Signals description:

RXA RS 232	= I -	Serial line A RS 232 Receive Data.
TXA RS 232	= O -	Serial line A RS 232 Transmit Data.
RXA- RS 422	= I -	Receive Data Negative: Serial line A negative signal for RS 422 serial differential receive.
RXA+ RS 422	= I -	Receive Data Positive: Serial line A positive signal for RS 422 serial differential receive.
TXA- RS 422	= O -	Transmit Data Negative: Serial line A negative signal for RS 422 serial differential transmit.
TXA+ RS 422	= O -	Transmit Data Positive: Serial line A positive signal for RS 422 serial differential transmit.
RXTXA- RS 485	=I/O-	Receive Transmit Data Negative: Serial line A negative signal for RS 485 serial differential receive and transmit.

RXTXB+ RS 485 =I/0- Receive Transmit Data Positive: Serial line A positive signal for RS 485 serial differential receive and transmit.

RXB- C.L. = I - Receive Data Negative: Serial line A negative signal for Current Loop serial bipolar receive.

RXB+ C.L. = I - Receive Data Positive: Serial line A positive signal for Current Loop serial bipolar receive.

TXB- C.L. = O - Transmit Data Negative: Serial line A negative signal for Current Loop serial bipolar transmit.

TXB+ C.L. = O - Transmit Data Positive: Serial line A positive signal for Current Loop serial bipolar transmit.

+5 Vdc = I - +5 Vdc or ground signal.

GND = - Ground signal.

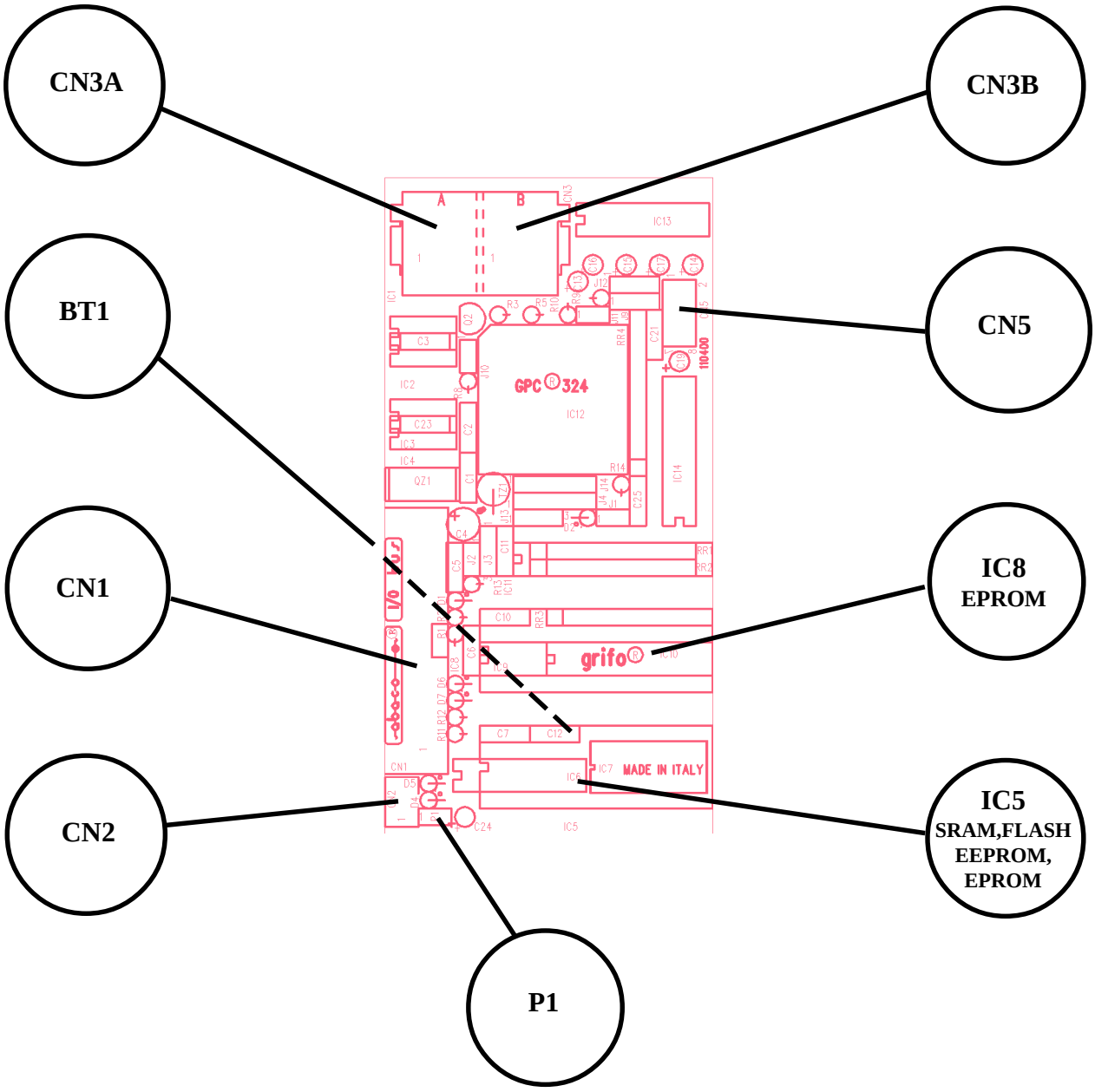


FIGURE 7: P1, MEMORIES, BATTERY, CONNECTORS LOCATION

CN3B - SERIAL LINE B CONNECTOR

CN3B is a 6 pins, female PLUG connector.
On CN3B are available the buffered signals for RS 232 serial line B that is phisically connected to the hardware serial line 1 of the microprocessor or to the software serial line. The electric protocol follows the CCITT normative and all the signals are placed in order to reduce interference and electrical noise and in order to simplify connection with other systems.

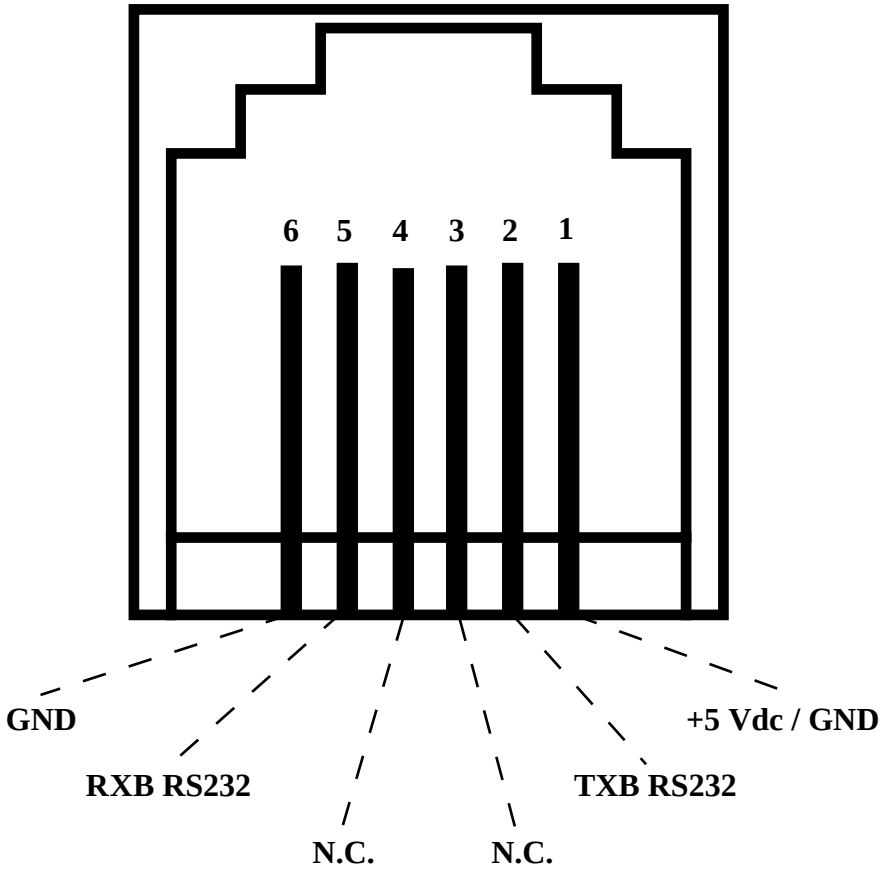


FIGURE 8: CN3B - SERIAL LINE B CONNECTOR

Signals description:

RXB RS 232	= I -	Serial line B RS 232 Receive Data.
TXB RS 232	= O -	Serial line B RS 232 Transmit Data.
+5 Vdc	= I -	+5 Vdc or ground signal.
GND	= -	Ground signal.
N.C.	= -	Not connected.

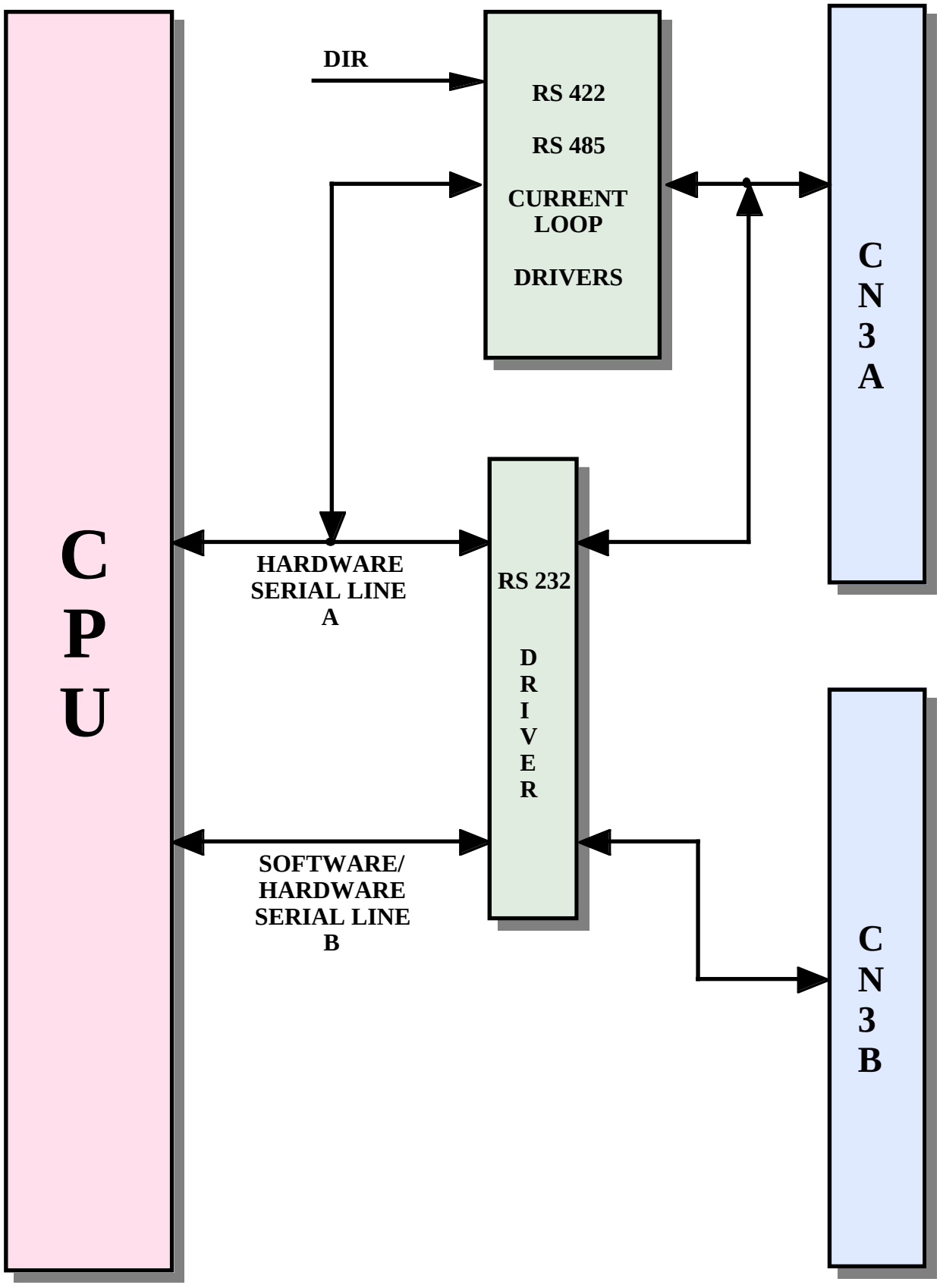


FIGURE 9: SERIAL COMMUNICATION DIAGRAM

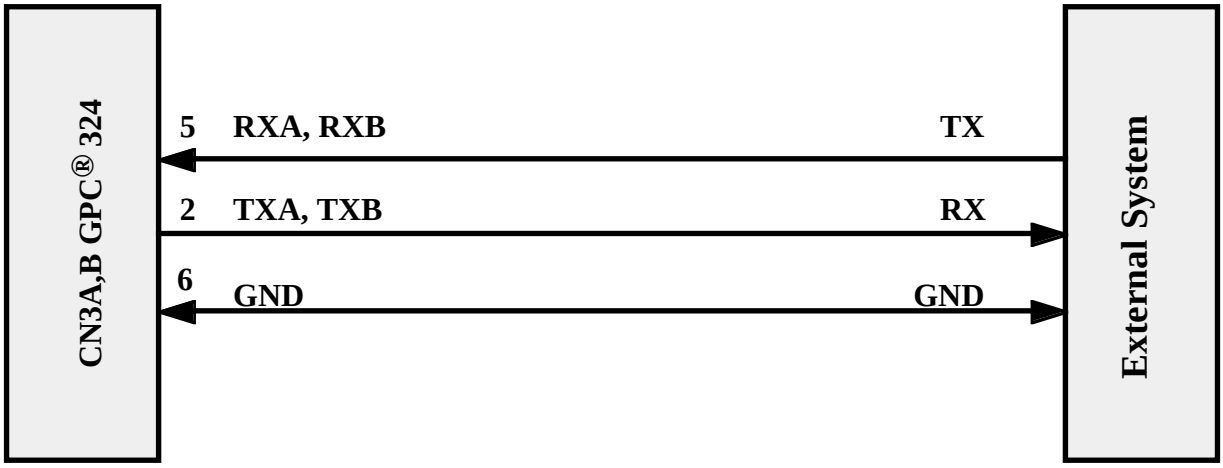


FIGURE 10: RS 232 POINT TO POINT CONNECTION EXAMPLE

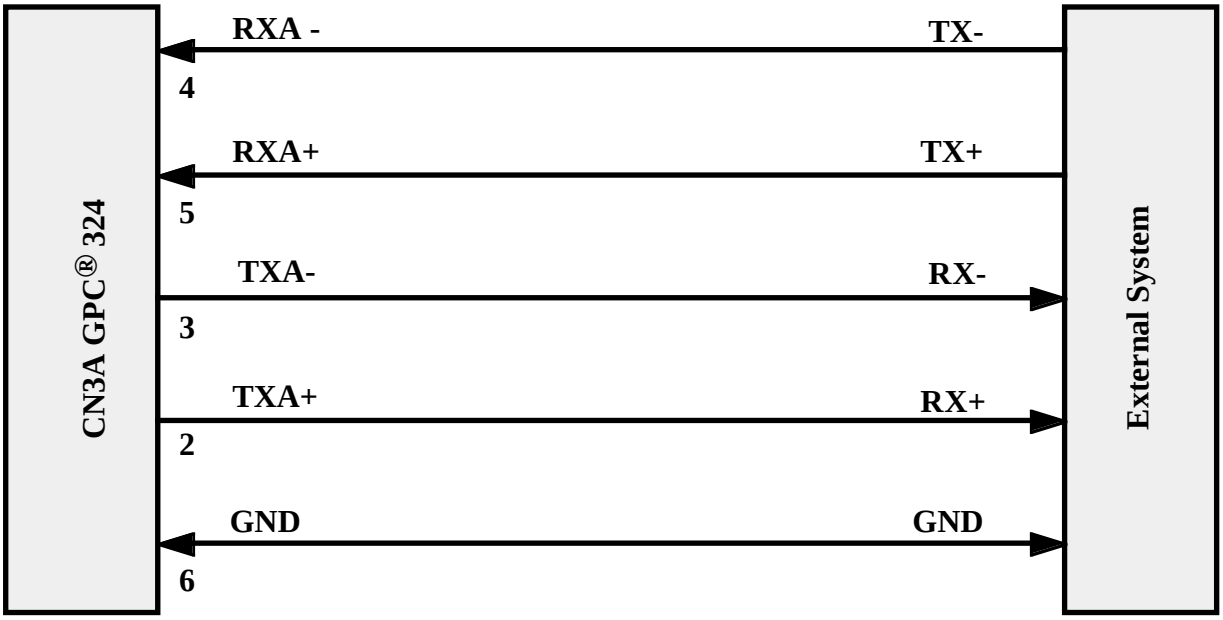


FIGURE 11: RS 422 POINT TO POINT CONNECTION EXAMPLE

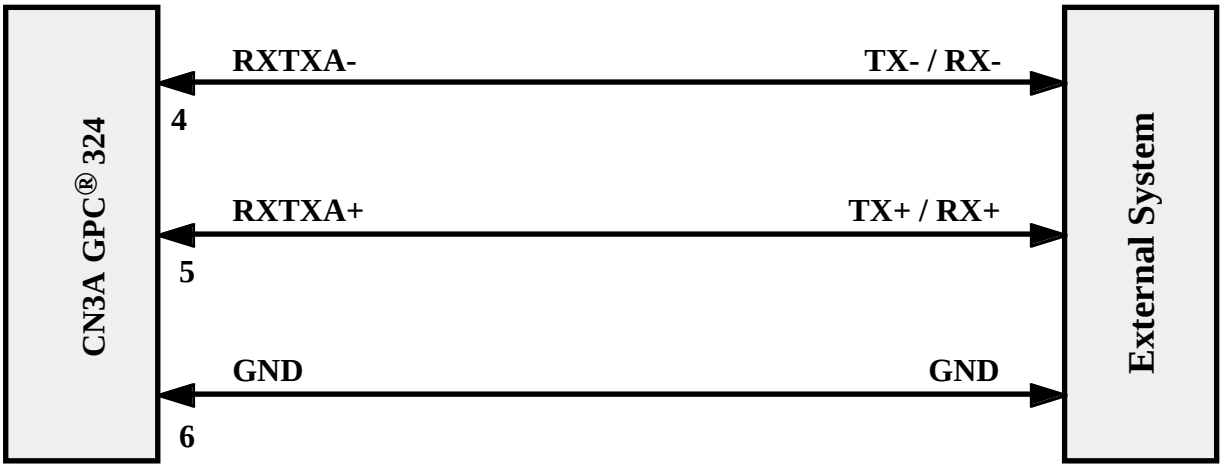


FIGURE 12: RS 485 POINT TO POINT CONNECTION EXAMPLE

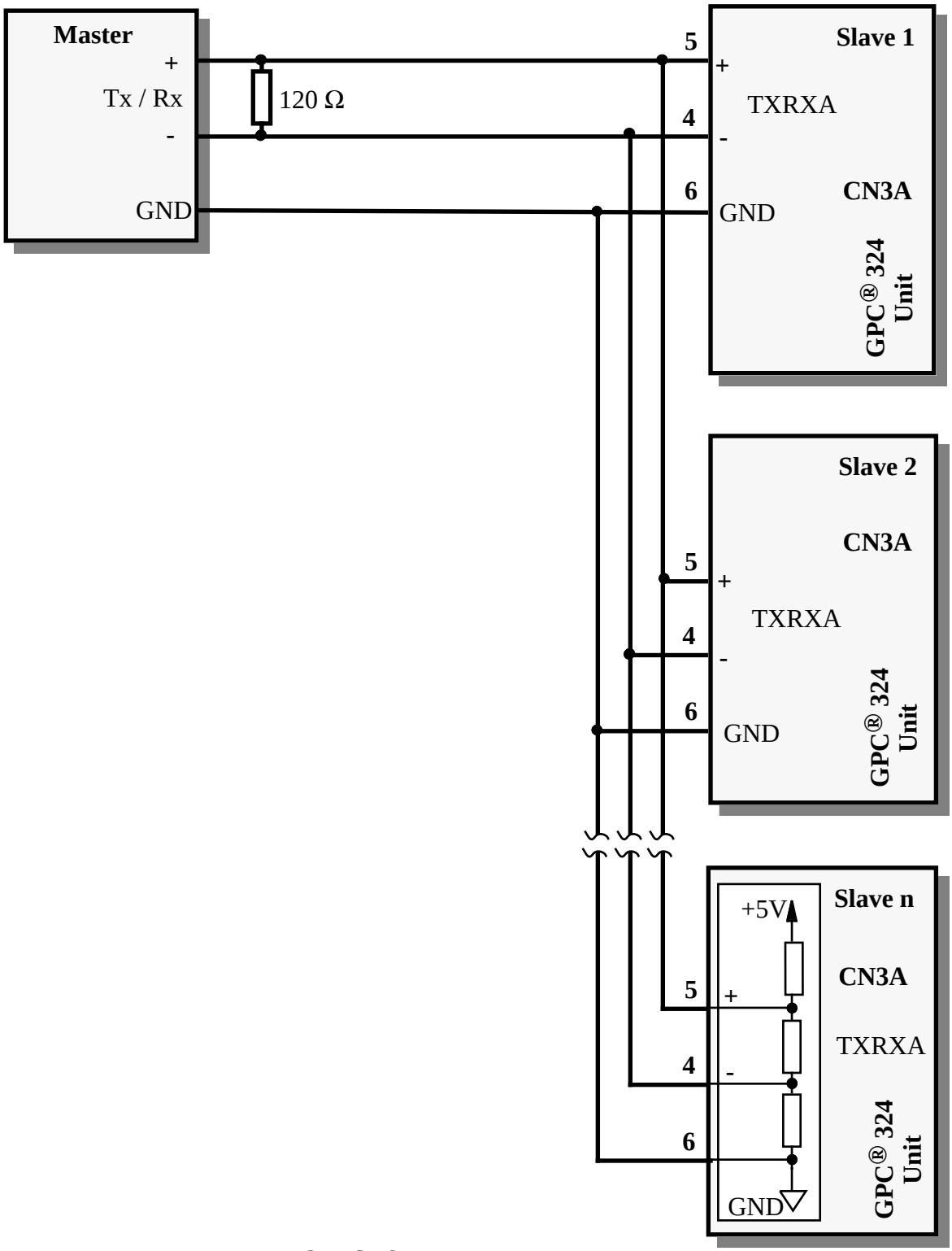


FIGURE 13: RS 485 NETWORK CONNECTION EXAMPLE

Please remark that in a RS 485 network two forcing resistors must be connected across the net and two termination resistors (120 Ω)must be placed at its extrems, respectevely near the master unit and the slave unit at the greatest distance from the master.
Forcing and terminating circuitry is installed on **GPC® 324** board; it can be enabled or disabled through specific jumers, as explained later.
For further informations please refr to TEXAS INSTRUMENTS data book, "*RS 422 and RS 485 Interface Cicuits*", the introduction about RS 422-485.

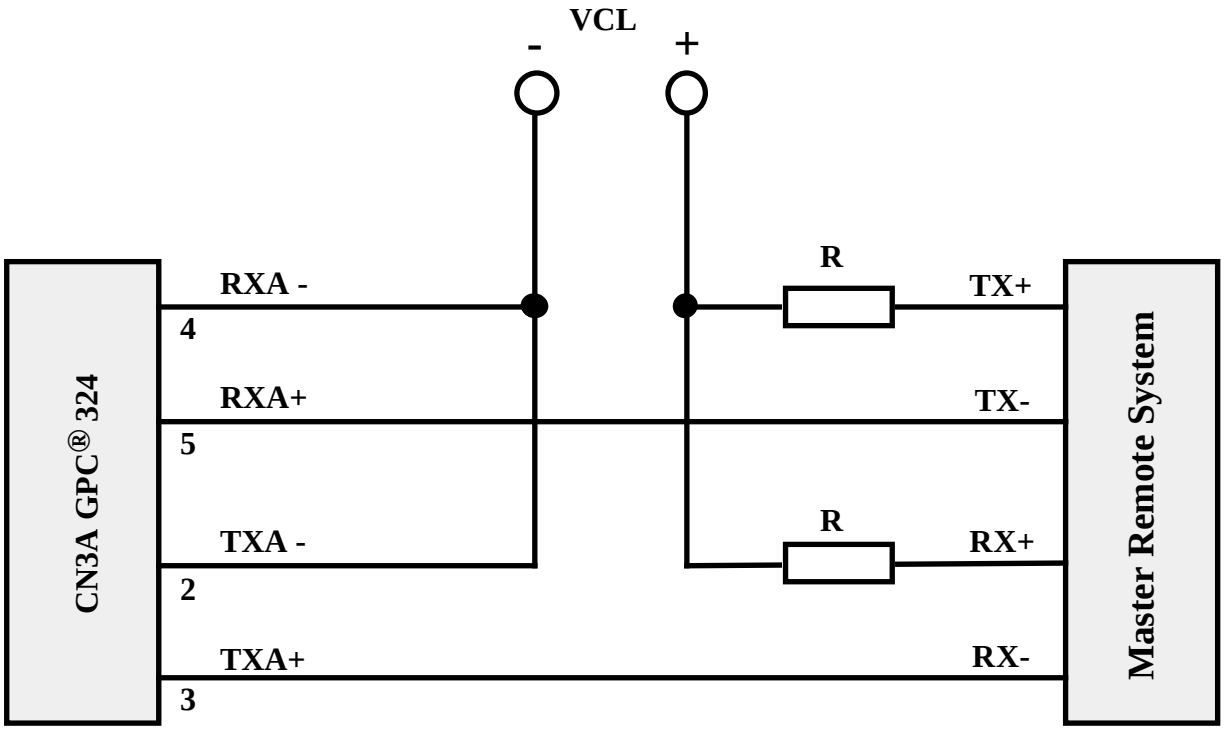


FIGURE 14: 4 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

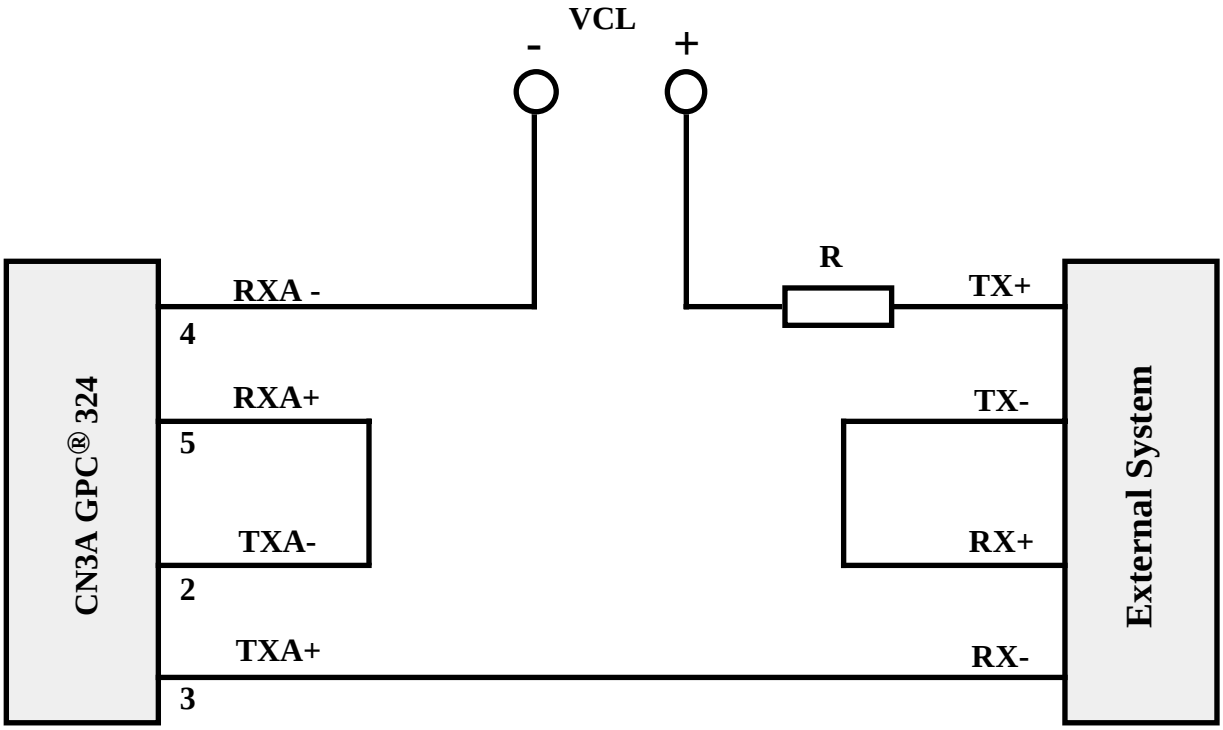


FIGURE 15: 2 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPL

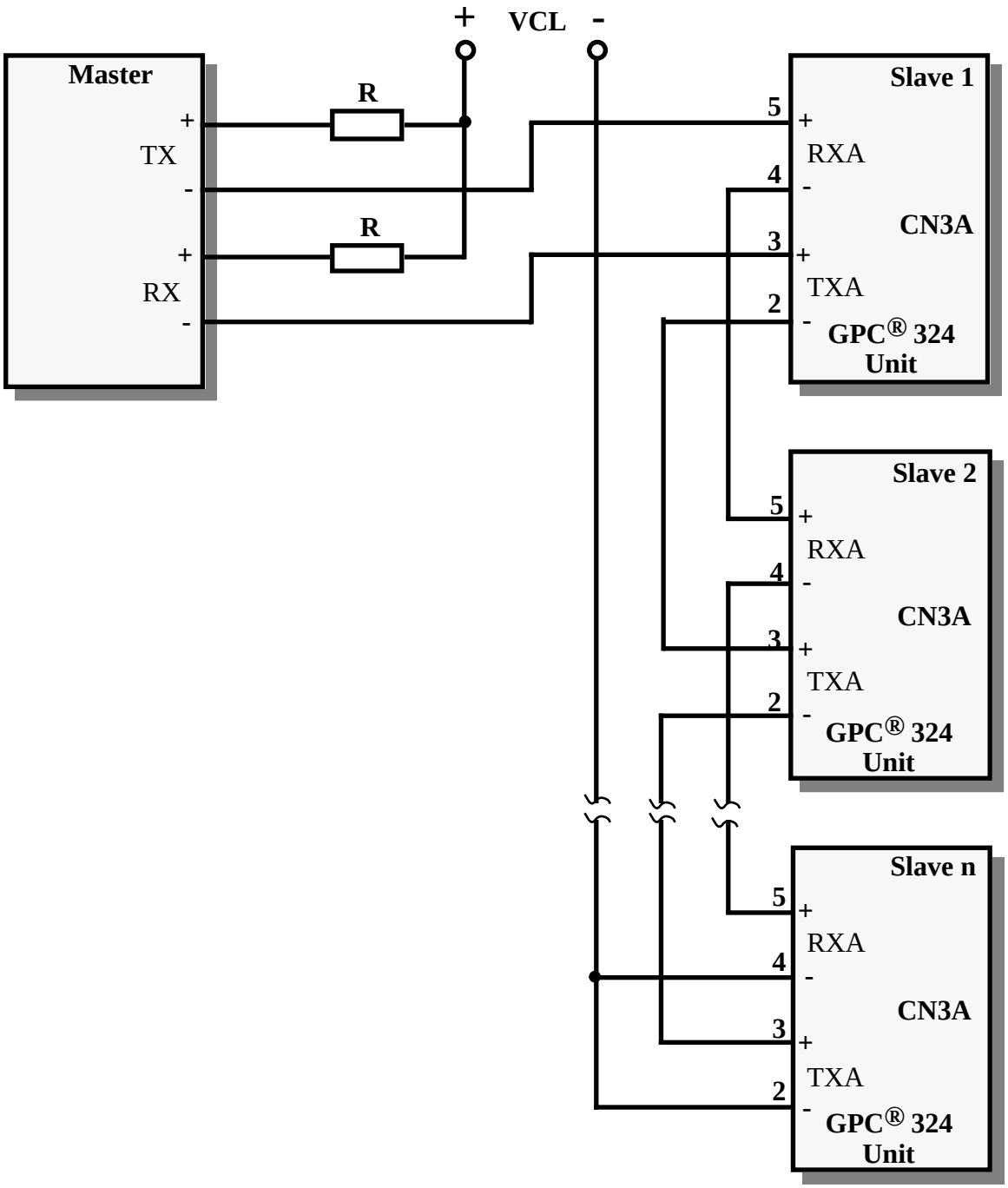


FIGURE 16: CURRENT LOOP NETWORK CONNECTION EXAMPLE

Possible Current Loop connections are two: 2 wires and 4 wires. These connections are shown in figures 14÷16 where it is possible to see the voltage for **VCL** and the resistances for current limitation (**R**). The supply voltage varies in compliance with the number of connected devices and voltage drop on the connection cable.
The choice of the values for these components must be done cosidering that:
- circulation of a **20 mA** current must be guaranteed;
- potential drop on each transmitter is about **2,35 V** with a 20 mA current;
- potential drop on each receiver is about **2,52 V** cwith a 20 mA current;
- in case of shortciruit each transmitter must dissipate at most **125 mW**;
- in case of shortciruit each receiver must dissipate at most **90 mW**.

For further info please refer to HEWLETT-PACKARD Data Book, (**HCPL 4100** and **4200** devices).

CN5 - CPU I/O LINES AND PROGRAMMING VOLTAGE CONNECTOR

CN5 is a 4+4 pins, male, vertical, strip connector with 2.54 mm pitch.
On CN5 are available some signals of microprocessor ports 1 and 3 that can be connected to external devices and the possible input for the voltage required by PHILIPS in system programmable microprocessors. Please note and remember that the lines P3.3 can be used only when jumper J2 is not connected.

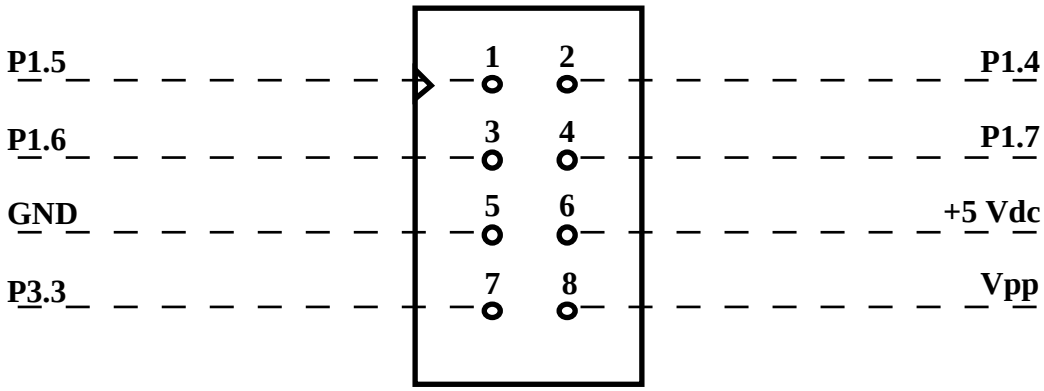


FIGURE 17: CN5 - CPU I/O LINES AND PROGRAMMING VOLTAGE CONNECTOR

Signals description:

P1.n	= I/O	- Digital line n of the CPU port 1.
P3.3	= I/O	- Digital line 3 of the CPU port 3 = /INT1.
Vpp	= I	- +12 Vdc or +5 Vdc programming voltage (for further information read paragraph "IN SYSTEM PROGRAMMING").
+5 Vdc	= I	- +5 Vdc power supply.
GND	=	- Ground signal.

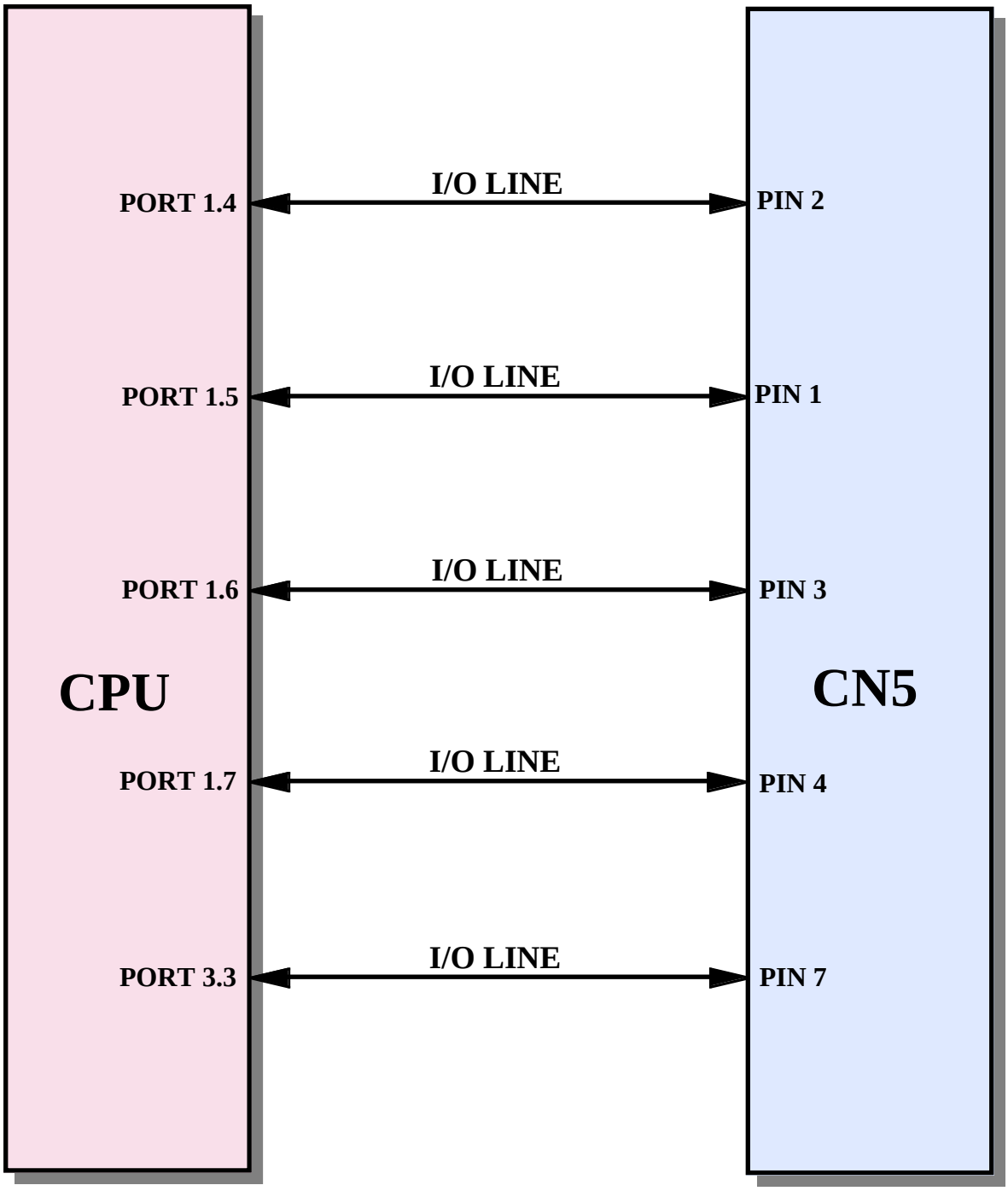


FIGURE 18: I/O LINES CONNECTION DIAGRAM

I/O CONNECTION

To prevent possible connecting problems between **GPC® 324** 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 and 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 and if an electric insulation is necessary, then an opto coupled interface must be connected. The 0 Vdc level corresponds to logic state "0", while 5Vdc level corrisponds to logic state "1".

DIGITAL I/O INTERFACES

Through CN1 (**ABACO®** I/O BUS standard connector) the **GPC® 324** card can be connected to all the numerous **grifo®** boards featuring the same standard pin out. It is really interesting the connection to **ETI 324** card that expands the digital I/O lines and that allows a direct connection to many digital I/O interfaces, ready to use. Installation of these modules is very easy; in fact only a 20 pins flat cable (code FLT20+20) is required, while the software management of these interfaces is as easy; in fact most of the software packages available for **GPC® 324** card are provided with the necessary procedures. Remarkable interfaces are:

- **QTP 16P, QTP 24P**, **KDL x24**, **KDF 224**, **DEB 01**, etc. that solve all the local operator interfacing problems. These modules are provided with all the resources needed to obtain a high level human machine interface (they feature alphanumeric displays, matrix keyboard and visualization LEDs) at a short distance from **GPC® 324** card. The available software drivers allow to manage the operator interface resources directly through the high level instructions for console management.

- **IAC 01**, **DEB 01**: it is an interface for CENTRONICS parallel printer that can be connected with a standard printer cable. About software the developed drivers provide procedures to read and write data at a specified address, for the selected programming language.

- **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 a complete file system interfacing allowing to access the informations stored in the memory card directly through the high level file management instructions.

- **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. Some of the above mentioned interfaces can be connected directly to CN4.

For more informations refer to "EXTERNAL CARDS" chapter and the software tools documentation.

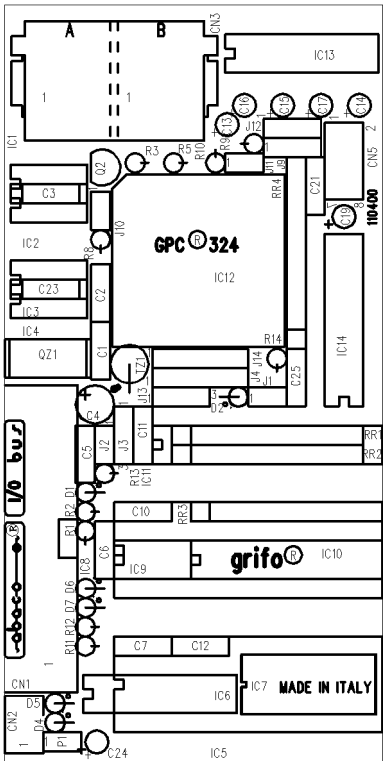


FIGURE 19: COMPONENTS MAPS (COMPONENTS SIDE)

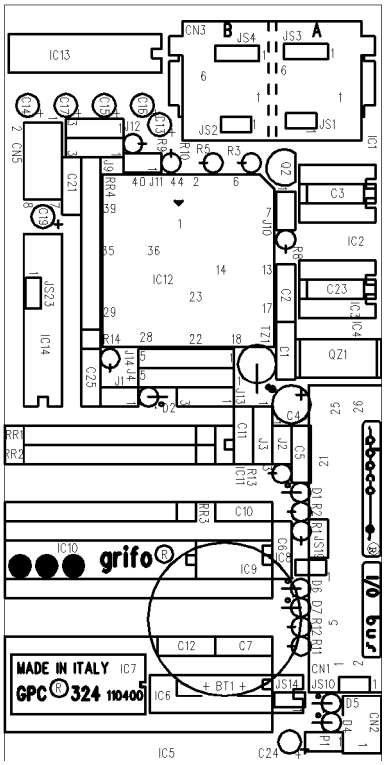


FIGURE 20: COMPONENTS MAPS (SOLDERING SIDE)

JUMPERS

On **GPC® 324** there are 19 jumpers for card configuration. Connecting these jumpers, the user can define for example the memory type and size, the peripheral devices functionality and so on. Here below is the jumpers list, location and function:

JUMPERS	PIN N°	FUNCTION
J1	2	It selects the connection for RUN/DEBUG user input.
J2	2	It selects the memory map.
J3	3	It selects the memory map.
J4	5	It selects IC5 memory type.
J9	3	It selects the electric interface for the serial communication line A.
J10, J11	2	They connect the serial line A, buffered in RS 232, to CN3A connector.
J12	3	It selects the direction and operating mode for RS 422 and RS 485 serial line A communication line.
J13	3	It selects the connection for pin 32 (/PSEN) of the CPU.
J14	5	It selects the connection for pin 35 (/EA) of the CPU.
JS1, JS2	2	They connect termination and force circuit to RS 422 and RS 485 serial line A communication line.
JS3	3	It selects the connection for pin 1 of CN3A.
JS4	3	It selects the connection for pin 1 of CN3B.
JS10	2	It enables/disables the external watch dog circuit.
JS14	2	It connects the on board battery BT1 to the back up circuit.
JS19	2	It connects the power failure circuit to microprocessor interrupt.
JS23	2	It connects the real time clock to microprocessor interrupt.

FIGURE 21: JUMPERS SUMMARIZING TABLE

The following tables describe all the right connections of **GPC® 324** jumpers with their relative functions. To recognize these valid connections, please refer to the board printed diagram (serigraph) or to figures 19,20 of this manual, where the pins numeration is listed, while for recognizing jumpers location, please refer to figures 22,23.
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.

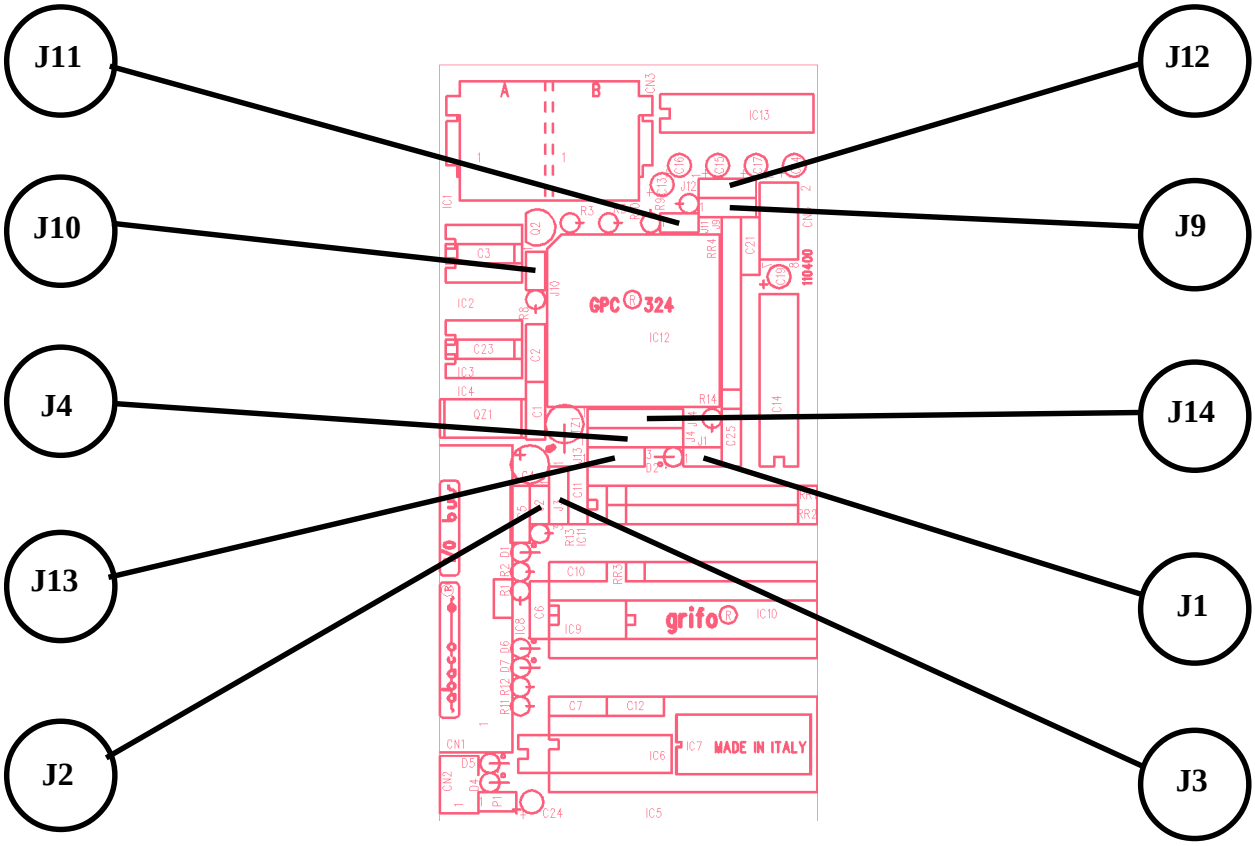


FIGURE 22: JUMPERS LOCATION (COMPONENT SIDE)

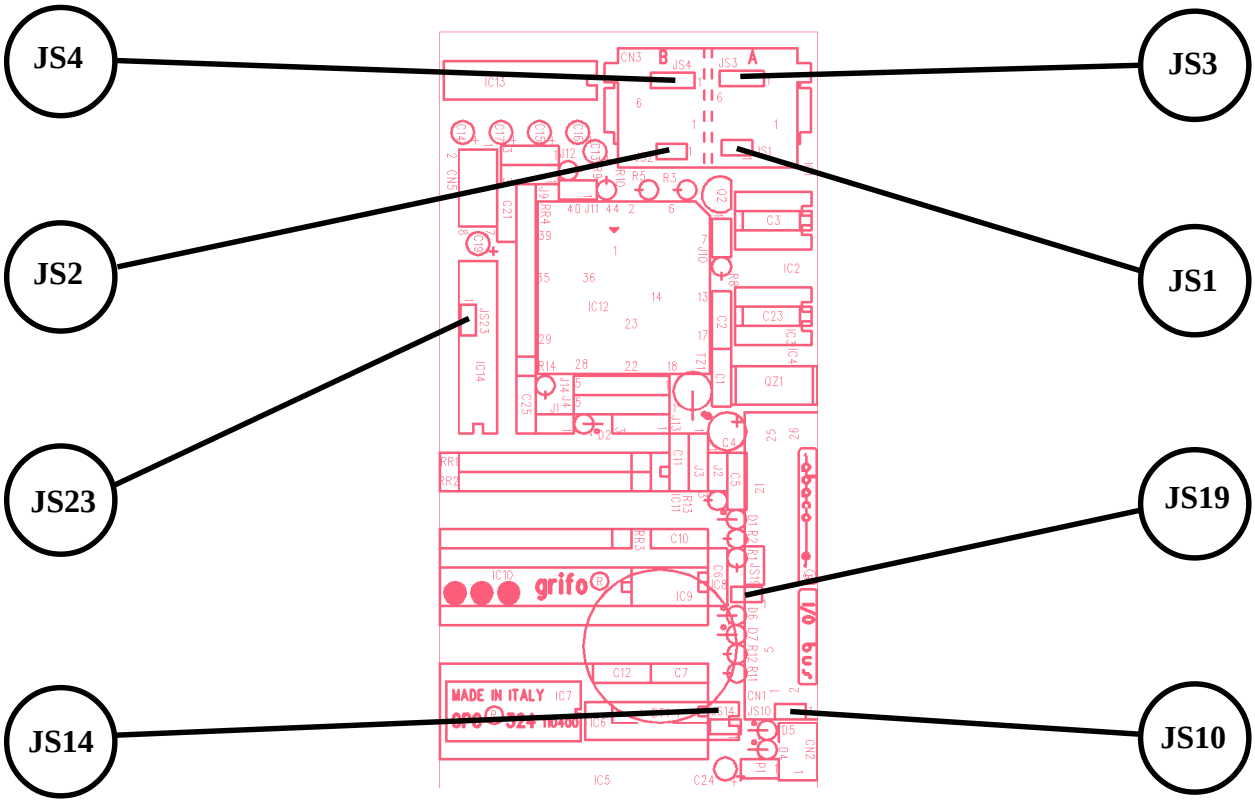


FIGURE 23: JUMPERS LOCATION (SOLDERING SIDE)

2 PINS JUMPERS

JUMPERS	CONNECTION	FUNCTION	DEF.
J1	not connected	It sets RUN/DEBUG user input at logic level 1 (DEBUG mode).	
	connected	It sets RUN/DEBUG user input at logic level 0 (RUN mode).	*
J2	not connected	This jumper is used with J3 for the memory map selection. For further information please refer to chapter "MEMORY ADDRESSES".	*
	connected		
J10, J11	not connected	The RS 232 serial line A is not connected to proper pins on CN3A.	
	connected	The RS 232 serial line A is connected to proper pins on CN3A.	*
JS1, JS2	not connected	The termination and force circuit is not connected to RS 422 and RS 485 serial line A.	*
	connected	The termination and force circuit is connected to RS 422 and RS 485 serial line A.	
JS10	not connected	It disables the watch dog circuit by hardware.	*
	connected	It enables the watch dog circuit by hardware.	
JS14	not connected	It doesn't connect the on board battery BT1 to the back up circuit.	*
	connected	It connects the on board battery BT1 to the back up circuit.	
JS19	not connected	It doesn't connect the power failure circuit to the microprocessor /INT0 interrupt.	*
	connected	It connects the power failure circuit to the microprocessor /INT0 interrupt.	
JS23	not connected	It doesn't connect the real time clock to the microprocessor /INT1 interrupt.	*
	connected	It connects the power failure circuit to the microprocessor /INT1 interrupt.	

FIGURE 24: 2 PINS JUMPERS TABLE

3 PINS JUMPERS

JUMPERS	CONNECTION	FUNCTION	DEF.
J3	not connected position 1-2 position 2-3	This jumper is used with J2 for the memory map selection. For further information please refer to chapter "MEMORY ADDRESSES".	*
J9	position 1-2	It configures serial line A for RS 232 standard.	*
	position 2-3	It configures serial line A for RS 422, RS 485 or current loop standard.	
J12	position 1-2	It configures serial line A for RS 485 electric standard (2 wires).	
	position 2-3	It configures serial line A for RS 422 electric standard (4 wires).	*
J13	position 1-2	It connects pin 32 (/PSEN) of CPU to GND, to enable the microprocessor FLASH programming.	
	position 2-3	It connects pin 32 (/PSEN) of CPU to control logic of the card.	*
JS3	position 1-2	It connects pin 1 of CN3A to GND.	*
	position 2-3	It connects pin 1 of CN3A to +5 Vdc.	
JS4	position 1-2	It connects pin 1 of CN3B to GND.	*
	position 2-3	It connects pin 1 of CN3B to +5 Vdc.	

FIGURE 25: 3 PINS JUMPERS TABLE

5 PINS JUMPERS

JUMPERS	CONNECTION	FUNCTION	DEF.
J4	position 1-2 and 3-4	It configures IC5 for 32K FLASH EPROM.	
	position 2-3 and 4-5	It configures IC5 for 32K SRAM or EEPROM.	*
	position 3-4	It configures IC5 for 32K EPROM.	
J14	position 1-2	It connects pin 35 (/EA) of CPU to Vpp from CN5, to provide the microprocessor FLASH programming voltage.	
	position 2-3 or 3-4	It connects pin 35 (/EA) of CPU to GND, to disable the microprocessor internal ROM.	*
	position 4-5	It connects pin 35 (/EA) of CPU to +5Vdc, to enable the microprocessor internal ROM.	

FIGURE 26: 5 PINS JUMPERS TABLE

MEMORY SELECTION

On **GPC® 324** can be mounted 104K bytes of memory divided in several configurations, as described in the following table:

IC	DEVICE	SIZE	JUMPERS CONFIGURATION
5	SRAM, EEPROM	32K Bytes	J4 in 2-3 and 4-5
	FLASH EPROM	32K Bytes	J4 in 1-2 and 3-4
	EPROM	32K Bytes	J4 in 3-4
7	SRAM	32K Bytes	-
8	EPROM	32K Bytes	-
9	EEPROM	256÷8192 Bytes	-

FIGURE 27: MEMORY SELECTION TABLE

The sockets IC 5 and IC 8 follow the JEDEC standard, so the mounted memory devices must have JEDEC pin outs. IC 9 is a serial memory device that must be specified at the moment of the order and can be mounted only by **grifo®** technicians. The jumpers configurations described on figure 27 only set the sockets for the indicated memory device, but there are some other jumpers that set the memory addressing map; for this information, please refer to "MEMORY ADDRESSES" paragraph. Normally **GPC® 324** is supplied in its default configuration with 32K RAM on IC 7 and 512 byte EEPROM on IC 9; each different configurations can be defined during order phase or self mounted by the user. Below are reported the abbrevation of the possible memory options:

.32K	->	32K x 8 SRAM
.32KMOD	->	32K x 8 backed SRAM
.32EE	->	32K x 8 parallel EEPROM
.32KF	->	32K x 8 parallel FLASH EPROM
.EE02	->	2K bit (256 byte) serial EEPROM
.EE08	->	8K bit (1K byte) serial EEPROM
.EE16	->	16K bit (2K byte) serial EEPROM
.EE32	->	32K bit (4K byte) serial EEPROM
.EE64	->	64K bit (8K byte) serial EEPROM

For further information and prices please contact directly **grifo®**.

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 MANAGEMENT

One of the most important **GPC® 324** 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.

- CPU inside devices	->	Possible sources of internal interrupt events are: timer 0÷2; serial port 0, 1; External interrupts 0÷5; internal watch dog.
- I/O signals	->	The P3.3 signal of CN5 is connected to pin /INT1 = P3.3 of CPU.
- Real Time Clock	->	It is wire anded to pin /INT1 = P3.3 of CPU, according to JS23 connection.
- Power failure	->	It is wire anded to pin /INT0 = P3.2 of CPU, according to JS19 connection.
- **ABACO®** I/O BUS	->	The /INT BUS signal of CN1 is connected to pin /INT0 = P3.2 of CPU. The /NMI BUS signal of CN1 is connected to pin T2 = P1.0 of CPU.

The last described connection is really important for two different reasons: each activation of /NMI BUS signal can generate an interrupt or each /NMI BUS signal change can be counted. The /NMI BUS signal management is defined by software programmation of timer 2 , so the user can select the favourite mode. This feature is really important especially when **GPC® 324** is connected to external card as **ZBT xxx** and **ZBR xxx**, in fact optocoupled digital signals can be counted or they can generate standard interrupts.
Themicroprocessor features a programmable priority structure that manages the case of contemporary interrupts. The addresses of the interrupt response subroutines can be software programmed by the user placing them on the proper code areas while the interrupts priority level and activation are sofware programmable through internal CPU registers. So the user program has always the possibility to react promptly to every external event, deciding also the priority of interrupts.

RESET AND WATCH DOG

The watch dog circuit of **GPC® 324** is really efficient and provided of easy software management. In details the most important features of this circuit are:
- astable functionality;
- intervent time of about 1420 msec;
- hardware enable;
- software retrigger;

With the astable mode when the intervent time elapses, the circuit becomes active, it stay active till the end of reset time (about200 msec) and after it is deactivated. Jumper JS10 connects the watch dog circuit to reset circuit so when it is connected the watch dog is enabled and viceversa. The watch dog retrigger operation is described in chapter "WATCH DOG".
After an activation and following deactivation of /RESET signal, the card resumes execution of the program saved on IC8 (at address 0000H) starting from a global reset status of all the on board peripheral devices.
Please remember that the /RESET signal is connected to CN1 connector and that on **GPC® 324** are available other reset sources as the power good circuit 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.

SERIAL COMMUNICATION SELECTION

Please remember that if not differently specified during the order phase, the card is delivered in its default configuration with two RS 232 serial line.
The serial line A is available on connector CN3A and 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 table). By software the serial line can be programmed to operate with 8, 9 bits per character, no parity, 1 stop bits at standard or no standard baud rates, through some some CPU internal register setting..
Some components necessary for RS 422 and RS 485 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 A CONFIGURED IN RS 232 (default configuration)

			IC 1	=	no component
J9	=	position 1-2	IC 2	=	no component
J10, J11	=	connected	IC 3	=	no component
J12	=	don't care	IC 4	=	no component
JS1, JS2	=	not connected	IC 13	=	MAX 202 driver

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

			IC 1	=	no component
J9	=	position 2-3	IC 2	=	HCPL 4200 driver
J10, J11	=	not connected	IC 3	=	no component
J12	=	don't care	IC 4	=	HCPL 4100 driver
JS1, JS2	=	not connected	IC 13	=	don't care

Please remark that current loop serial interface is passive, so it must be connected an active Current Loop serial line, that is a line provided with its own power supply. Current loop interface can be employed to make both point to point and multi point connections through a 2 wires or a 4 wires connection as described in figures 14÷16.

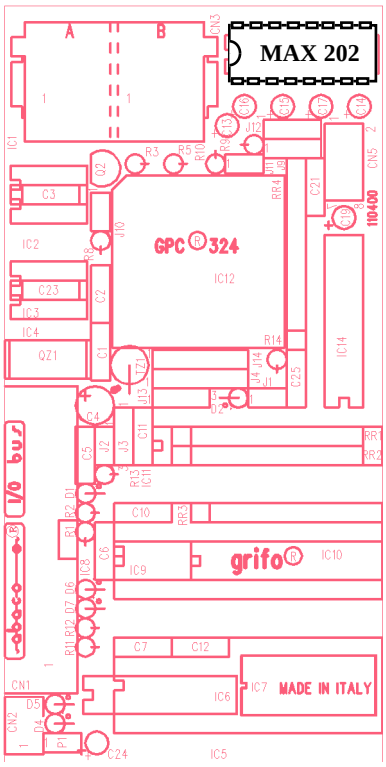
- SERIAL LINE A CONFIGURED IN RS 422 (.RS422 option)

			IC 1	=	SN 75176 driver
J9	=	position 2-3	IC 2	=	no component
J10, J11	=	not connected	IC 3	=	SN 75176 driver
J12	=	position 1-2	IC 4	=	no component
JS1, JS2	=	(*)	IC 13	=	don't care

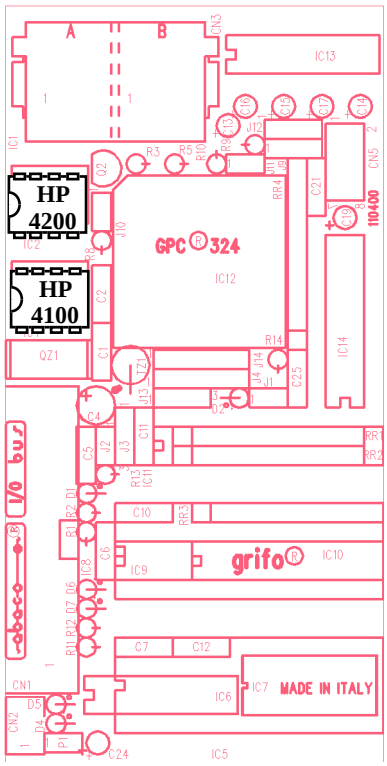
Status of signal DIR, which is software managed, allows to enable or disable the transmitter as follows:

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

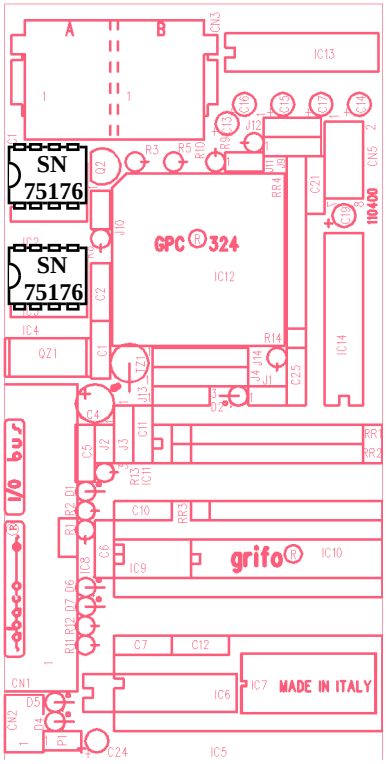
In point to point connections, signal DIR can be always kept low (trasnmitter always enabled), while in multi point connections transmitter must be enabled only when a transmission is requested.



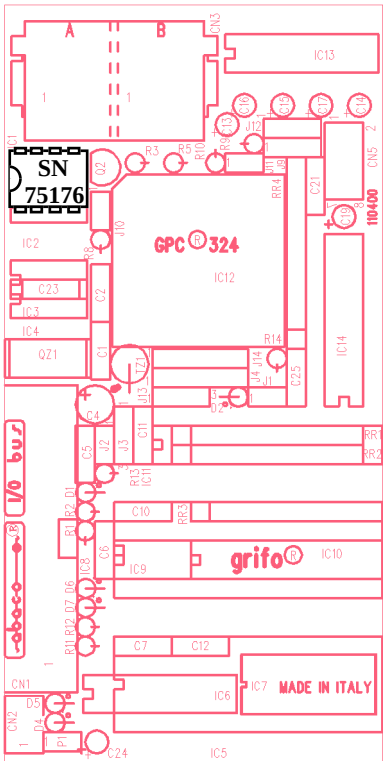
Serial A in RS 232



Serial A in Current Loop



Serial A in RS 422



Serial A in RS 485

FIGURE 28: SERIAL COMMUNICATION DRIVERS LOCATION

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

			IC 1	=	SN 75176 driver
J9	=	position 2-3	IC 2	=	no component
J10, J11	=	not connected	IC 3	=	no component
J12	=	position 2-3	IC 4	=	no component
JS1, JS2	=	(*)	IC 13	=	don't care

In this modality the signals to use are pins 4 and 5 of connector CN3A, that become transmission or reception lines according to the status of signal DIR, managed by software, as follows:

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

This kind of serial communication can be used for multi point connections, in addition it is possible to listen to own transmission, so the user is allowed to verify the succes of transmission. In fact, any conflict on the linecan be recognized by testing the received character after each transmission.

(*) If using the RS 422 or RS 485 serial line, it is possible to connect the terminating and forcing circuit on the line by using JS1 and JS2. This circuit must be always connected in case of point to point connections, while in case of multi point connections it must be connected olny in the farest boards, that is on the edges of the commmunication line.

When a reset or a power on occour, signal DIR is kept to a logic level high, so in any of these two cases driver RS 485 is receiving or RS 422 driver is disabled, avoidng eventual conflicts in communication.

The serial line B is available on connector CN3B and it can be buffered only in RS 232. By mounting a MAX 202 driver on IC 13 this second serial line is hardware enabled and by software it can be managed as below described:

- µP 80C32 and compatible ones

The serial line B is a software serial line, managed through two I/O pins of the processor (pin 4 = P1.2 = RXB and pin 5 = P1.3 = TXB). The communication parameters (baud rate, stop bit, bit x char, etc.) are software defined through some timing and some sequence of management procedure. For further information, please refer to the software tools manuals.

- µP 80C320 and compatible ones

The serial line B is an hardware serial line, managed through the dedicated pins of the processor (pin 4 = RXB and pin 5 = TXB). The communication parameters (baud rate, stop bit, bit x char, etc.) are software defined through proper microprocessor registers setting. For further information, please refer to microprocessor data sheets.

For further informations about serial communication please refer to the examples of figures 10÷16 and paragraph "RS 422-485 DIRECTION".

BACK UP

GPC® 324 ha an on board lithium battery BT1 for the back up of SRAM and RTC content when power supply is switched off. Jumper JS14 connects phisically the battery so it can be disconnected to save its duration whenever back up is not needed. By CN2 connector itis possible to connect an external battery: configuration of jumper JS14 does not affect the working of this battery and it can replace BT1 completely. The user can order an external battery (2.1 Ah) ready to be connected to CN2 with the code:.LITIO.
Please refer to the paragraph “ELECTRIC FEATURES” to choose the type of the external back up battery, to easily locate see figure 7.

POWER FAILURE

In addition to the CPU controlled power management circuitery, **GPC® 324** card also features an efficent power failure circuitery. Through jumper JS19 this latter can be connected to the micropocessor /INT0 interrupt signal.
The task of this circuitry is to keep under control power supply voltage and activate on output to request a CPU action when this voltage reaches a value lower than a threshold of **52 mV** above the reset intervent.
Please remark that the time interval between power failure activation and reset activation changes according to the type of supply being used; it is however about 100 µsec, long enough only to execute a fast response routine (for example to save a flag in the backed SRAM).
Typical use of power failure is to inform the board about the imminent power supply black out, so the CPU can save appropriate informations.

IN SYSTEM PROGRAMMING

One of the most important features of **GPC® 324** is the possibility to use the new PHILIPS 89CRx+ /2 microprocessors that support the in system and in application programming (ISP). Below are listed the sequence of operations that must be performed by the user to use this features:

1) develop the application program through a proper software tools that generate an executable code;
2) connect jumper J13 in position 1-2 and J14 in position 1-2;
3) connect the programming voltage (according to used microprocessor: +12 V for 89CRx+ or +5 V for 89CRx2) to pin 8 of CN5, referred to pin 5 always of CN5;
4) connect RS 232 serial line A to a personal computer free COM line;
5) power on the card;
6) program the microprocessor internal FLASH EPROM by using the specific program supplied by PHILIPS: **WINISP**.
7) power off the card;
8) disconnect the programming voltage from CN5, connect J13 in position 2-3 and J14 in position 4-5;
9) power on the card: the programmed application programm will start execution from internal ROM.

The ISP reduce the total application cost, in fact it eliminates the requirements of EPROM, EPROM programmer, external FLASH EPROM, etc. For further informations on in system programming please refet to specific technical documentation from PHILIPS.

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 51 family, can be used.

BASIC 324: complete development tools for MCS BASIC (interpreted BASIC language for industrial application). It needs a P.C. for console and program saving operations, while the debug, test and program operations are performed on the card. Special instructions which manage the on board peripheral devices have been added, plus the possibility to save the application on EEPROM.

BXC51: cross compiler for source files written in BASIC 324. It allows a considerable speed increment of the starting MCS BASIC program and it is completely executed on external P.C.

FORTH: complete software development tools to program the card with FORTH high level language. It needs a P.C. for user interface and it is really interesting for its fast execution and small size, of the generated code.

MCC 51: integer cross compiler for source files in standard ANSI C. It produces a source assembly file compatible with MCA 51 or with Intel macro relocatable assembler MCS 51.

MCA 51: macro cross assembler available for MS-DOS operating system in absolute and relocatable version. In this relocatable version is supplied with a linker and a library manager.

MCS 51: source level debugger and simulator. It is executed on P.C. without any additional hardware and it allows loading of HEX and SYMBOLIC files, breakpoint setting, instruction execution in trace mode, registers and memory dump, etc.

MCK 51: it is the sum of MCC 51 and MCA 51 and it is a complete C compiler with an output file type compatible with MCS 51.

HI TECH C 51: cross compiler for C source program. It is a powerful software tool that includes editor, C compiler, assembler, optimizer, linker, library, and remote symbolic debugger, in one easy to use integrated development environment.

SYS51CW: cross compiler for C source program. It is a powerful software tool that includes editor, C compiler, assembler, optimizer, linker, library, simulator and remote symbolic debugger, included in an easy to use integrated development environment for Windows.

SYS51PW: cross compiler for PASCAL source program. It is a powerful software tool that includes editor, PASCAL compiler, assembler, optimizer, linker, library, simulator and remote symbolic debugger, included in an easy to use integrated development environment for Windows.

MDP: monitor debugger program able to load and debug each HEX Intel files for 51 microprocessor family. It is provided of the standard commands available on in circuit emulator and requires only an external P.C. connected through a serial line.

DDS MICRO C 51: low cost ross compiler for C source program. It is a powerful software tool that includes editor, C compiler (integer), assembler, optimizer, linker, library, and remote debugger, in one easy to use integrated development environment. There are also included the library sources and many utilities programs.

SOFTICE: It is a remote symbolic debugger with cross assembler. It is provided of the standard commands available on in circuit emulator and requires only an external P.C. connected through a serial line. There is an high level user interface that can visualize all the processor status in a multiwindow visualization.

NOICE 51: It is a PC hosted debugger consists of a target specific DOS program, NOICExxx.EXE, and a target resident monitor program. The two programs comunicate via RS 232. NOICE includes: source level debug; a disassembler; a file viewer; memory display and editing; a virtually unlimited number of breakpoints; hardware free single step; definition of symbols; the ability to record and play back files of commands; on line help.

OPEN 51/UNI: in circuit emulator for the 51 family. It is a powerfull hardware and software tool that includes: source level debugging and symbolic debugging; project management; built-in multi-file editor; execution of external compilers; debugging of several modules at the same time; built-in disassembler; source level step and trace functions; animate functions; inserting and deleting of breakpoints on the source level; watching and modifying variables on symbol and absolute level.

BASCOM 8051: cross compiler for BASIC source program. It is a powerful software tool that includes editor, BASIC compiler and simulator included in an easy to use integrated development environment for Windows. Many memory models, data types and direct use of hardware resource instructions are available.

GET 51: it is a complete program with editor, communication driver and mass memory management for all '51 family cards. This program developed by **grifo®** allows to operate in the best conditions when BASIC 324, BXC51, MDP, FMO 52 software tools are used. The program is menu driven and mouse driven. It is designed to run undr MS-DOS but can run also in MACINTOSH environment with VIRTUAL-PC. It is delivered in MS-DOS 3”1/2 floppy disks.

FM052: monitor debugger program able to load and debug each HEX Intel files for 51 microprocessor family. It is provided of the standard commands available on in circuit emulator and requires only an external P.C. connected through a serial line. It is preconfigured to work directly with BASCOM 8051 and GET51 software tools. It can also program the on board FLASH EPROM with the user application program and then execute it in autorun mode.

INTRODUCTION

In this chapter are reported all information about card use, related to hardware and software. For example, the registers addresses and the memory allocation are described below.

ADDRESSES

The card devices addresses are managed by a specific control logic, realized with programmable array logic. This control logic allocates memory and peripheral devices in a confortable mode for the user. The available microprocessors address 64K bytes of code memory and 64K bytes of data memory and the control logic maps the on board memory and peripheral devices, inside these addresses spaces. Control logic sets size, type and addresses of memory devices through jumpers J2, J3 and J4; at the same time it allocates I/O addresses always in the upper 256 bytes of microprocessor data memory. Summarizing the control logic allocates:

- 32K bytes of EPROM on IC 8;
- 32K bytes of SRAM on IC 7;
- 32K bytes of SRAM, EPROM, EEPROM, FLASH EPROM on IC 5;
- **ABACO®** I/O BUS;
- RUN/DEBUG selector (J1 status);
- External watch dog retrigger;
- Real time clock

The addresses of all these devices are described in the following paragraphs and can't be set with different value. The other device serial EEPROM of IC 9, is managed always by control logic but it is not allocated in memory space, in fact this device is drived through CPU I/O lines with a syncronous communication.

MEMORY ADDRESSES

On the **GPC® 324** three different memory configurations can be used. The configuration must be selected, with J2 and J3, both according to used software tools and user requests and/or application features. The following figures describe available memory configurations, with proper J2 and J3 setting.
The position 1-2 for jumper J3 is not described because it is provided for future expansion.

MEMORY CONFIGURATION 0

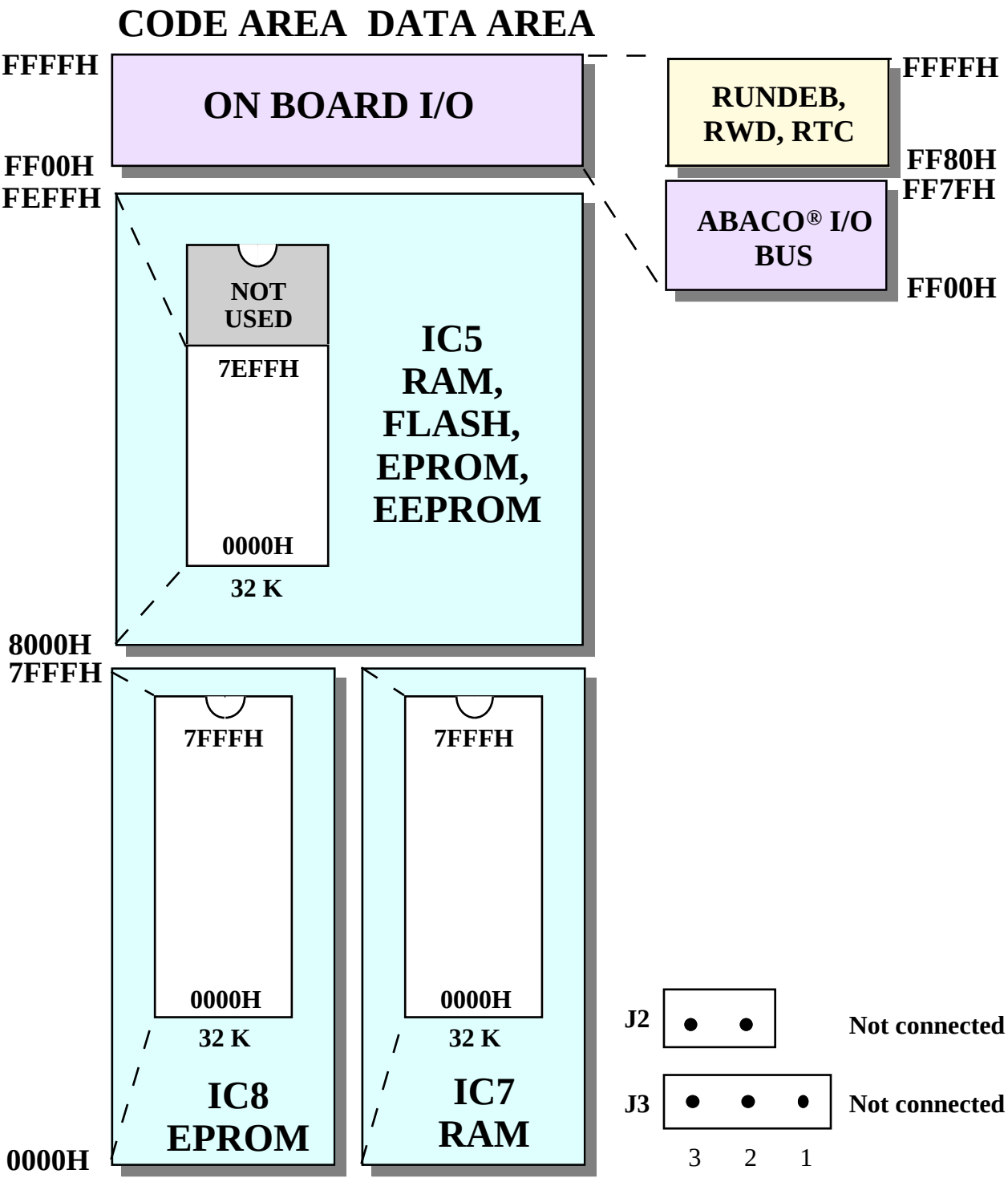


FIGURE 29: MODE 0 MEMORY CONFIGURATION

Used by software tools as: BASIC 324; BXC51; HI TECH C; DDS MICRO C; SYS51CW; SYS51CW; BASCOM 8051; SOFTICE (**J3** in **1-2**); etc.

MEMORY CONFIGURATION 1

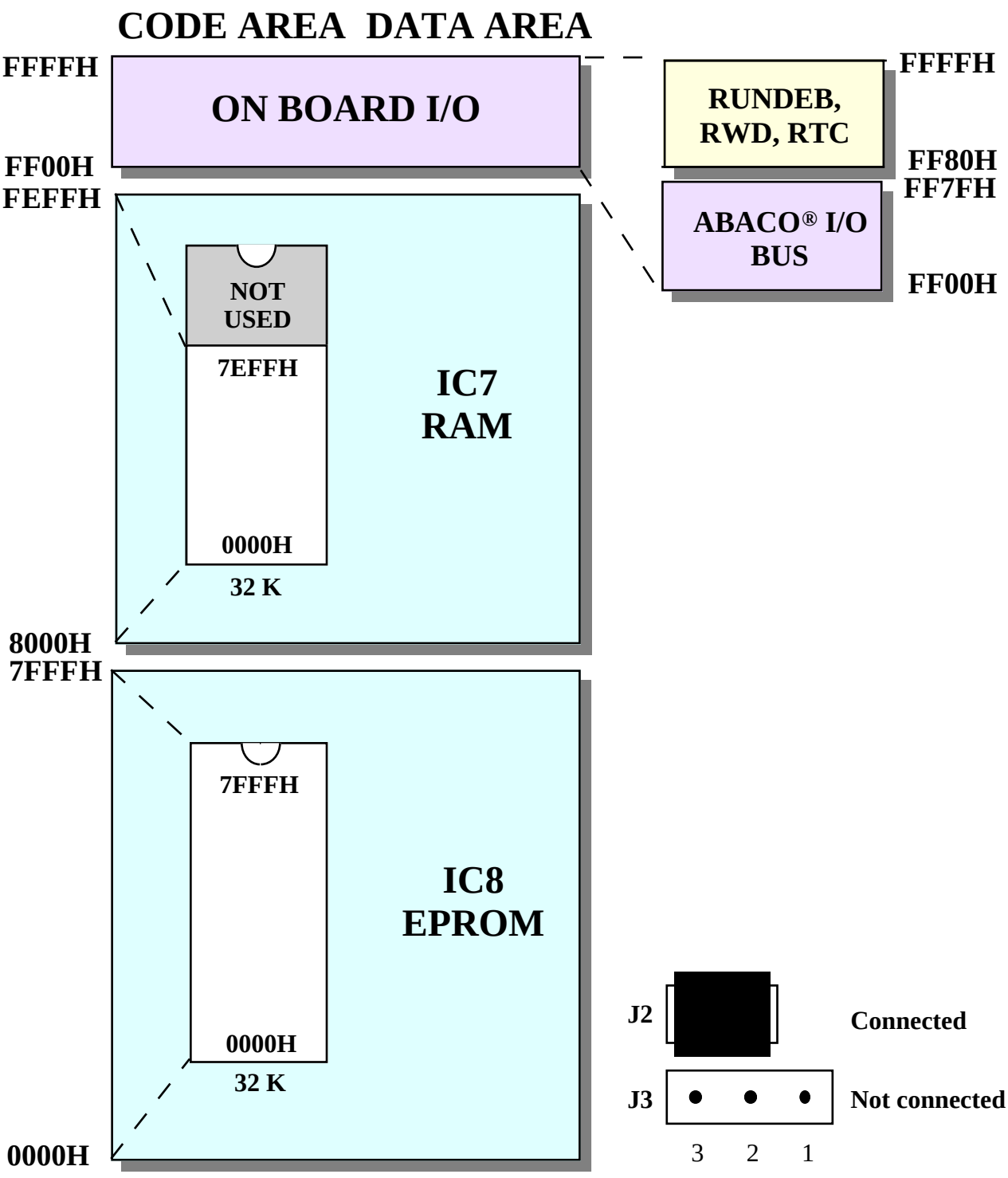


FIGURE 30: MODE 1 MEMORY CONFIGURATION

Used by software tools as: HI TECH C; DDS MICRO C; SYS51CW; SYS51CW; BASCOM 8051; etc.

MEMORY CONFIGURATION 3

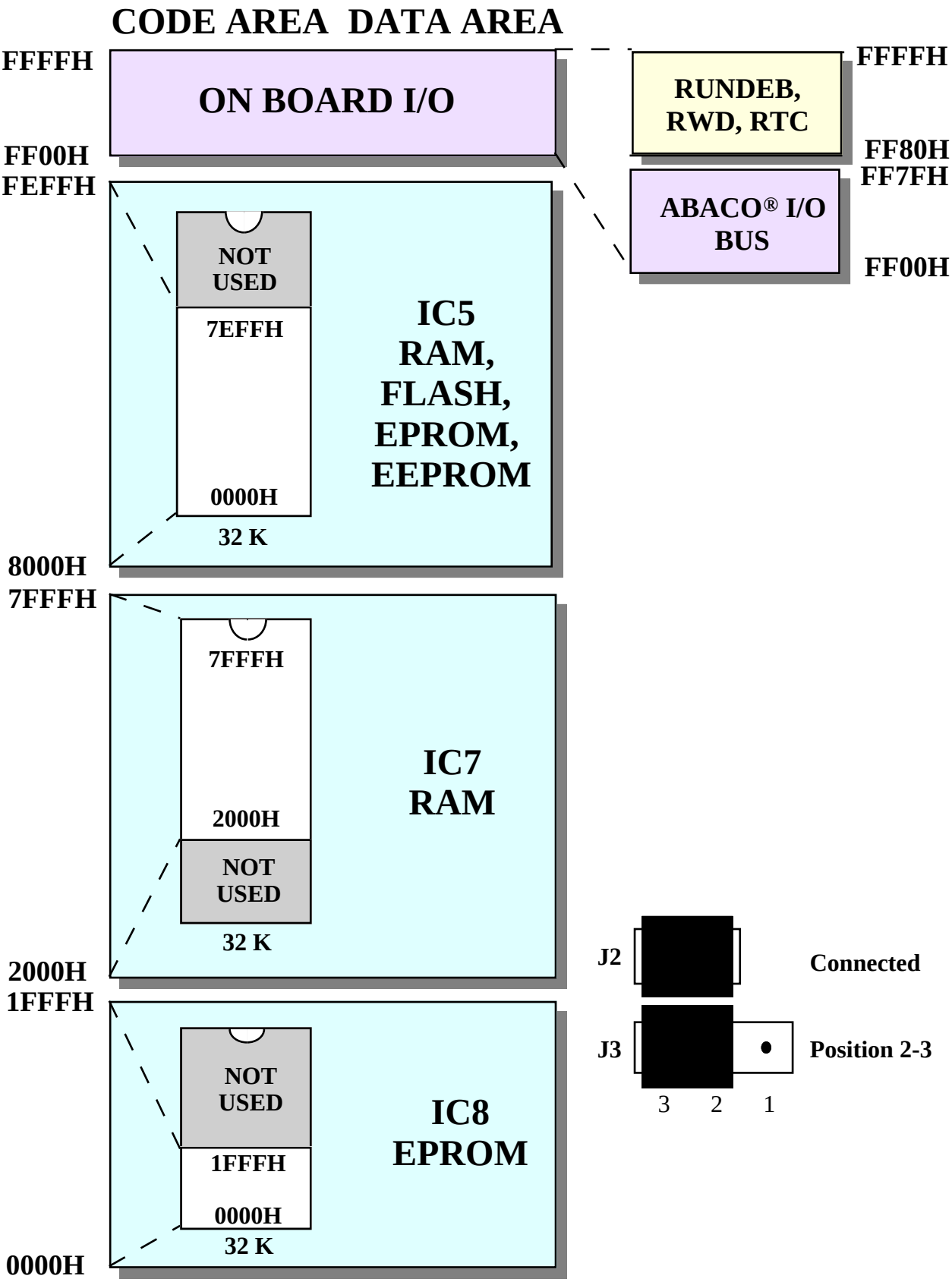


FIGURE 31: MODE 3 MEMORY CONFIGURATION

Used by software tools as: MDP; LUCIFER HI TECH C; DDS MICRO C; SYS51CW; SYS51CW; BASCOM 8051; FMO 52, NO ICE 51, etc.

I/O ADDRESSES

I/O addresses are located in the last 256 bytes (128 for **ABACO® I/O BUS** and 128 bytes for RUN/DEBUG reading, watch dog retrigger and real time clock) of the 64K bytes data microprocessor addressing space. Next table shows name, addresses, meanings and direction of peripheral devices registers (only the external ones to microprocessor):

DEVICES	REG.	ADDRESS	R/W	FUNCTION
ABACO® I/O BUS	I/O BUS	FF00H÷FF7FH	R/W	**ABACO®** I/O BUS addresses.
RUN/DEBUG	RUNDEB	FF80H÷FFFFH	R	J1 status reading register.
WATCH DOG	RWD	FFC0H÷FFFFH	R	Retrigger external watch dog.
REAL	SEC1	FFC0H	R/W	Data register for seconds units
TIME	SEC10	FFC1H	R/W	Data register for seconds decines
CLOCK	MIN1	FFC2H	R/W	Data register for minutes units
	MIN10	FFC3H	R/W	Data register for minutes decines
	HOU1	FFC4H	R/W	Data register for hours units
	HOU10	FFC5H	R/W	Data register for hours decines and AM/PM
	DAY1	FFC6H	R/W	Data register for day units
	DAY10	FFC7H	R/W	Data register for day decines
	MON1	FFC8H	R/W	Data register for month units
	MON10	FFC9H	R/W	Data register for month decines
	YEA1	FFCAH	R/W	Data register for year units
	YEA10	FFCBH	R/W	Data register for year decines
	WEE	FFCCH	R/W	Data register for week day
	REGD	FFCDH	R/W	Control register D
	REGE	FFCEH	R/W	Control register E
	REGF	FFCFH	R/W	Control register F

FIGURE 32: I/O ADDRESSES TABLE

For further information about register meanings, please refer to next chapter called "PERIPHERAL DEVICES SOFTWARE DESCRIPTION".

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 figure 32, for the registers name and addresses values). For microprocessor internal peripheral devices, not described in this paragraph, or for further information, please refer to manufacturing company documentation. In the following paragraphs the **D7÷D0** and **.0÷7** indications denote the eight bits of the combination used in I/O operations.

CONFIGURATION INPUT: J1 (RUN/DEBUG)

The on board J1 status can be obtained by software, through a simple read operation at the RUNDEB register address. The corrispondence between register bits and J1 status is as follows:

D0÷D6	->	Reserved
D7	->	J1 status (RUN/DEBUG in some software tools)

Remember that if the jumper is **connected** the logic status is **0** and if the jumper is **not connected** the logic status is **1**. This register shares the same address of other on board peripherals so the user must remember that the configuration input acquisition has effects also on other peripheral section. This jumper is normally used for system configuration (operating mode selection, card number programmation inside a network system, fimware configuration, etc.).

EXTERNAL WATCH DOG

Retrigger operation of **GPC® 324** external watch dog circuit is performed with a simple read operation at the address of register RWD. This register shares the same address of other on board peripherals, but no conflict are generated in fact retrigger operation is an input operation and the read data has no meaning. To avoid external watch dog activation is necessary to retrigger its circuit at regular time periods and the duration of these periods must be smaller than intervention time. If retrigger doesn't happen as before described and JS10 is connected, when intervention time is elapsed, the card is reset. The default intervention time is about 1420 msec.

RS 422-485 DIRECTION

To manage the RS 485 line direction or the RS 422 transmission driver enabling, on **GPC® 324** is used a proper digital I/O line of microprocessor, named DIR. This line is driven by CPU pin 3 (P1.1) and it has the following functionality:

- RS 485:	DIR = 0	->	RS 485 line transmitting
	DIR = 1	->	RS 485 line receiving
- RS 422:	DIR = 0	->	RS 422 transmitter driver enabled
	DIR = 1	->	RS 422 transmitter driver disabled

DIR signal is set (1) after reset or power on, maintaining enabled the RS 485 reception and mantaining disabled the RS 422 transmission; in this way each conflict is eliminated.

SERIAL EEPROM

For software management of serial EEPROM module of IC 9, please refer to specific documentation or to demo programs supplied with the card. The user must realize a serial communication with I^2C bus standard protocol, through two I/O microprocessor pins. The only necessary information is the electric connection:

DATA line (SDA)	->	pin 16 (P3.4) of the CPU
CLOCK line (SCL)	->	pin 17 (P3.5) of the CPU
A2 address line	->	GND (0 logic state)
A1 address line	->	GND (0 logic state)
A0 address line	->	GND (0 logic state)

The first 30 bytes of serial EEPROM are reserved for software tools use, so they can't be neither read nor written by user program.

REAL TIME CLOCK

This peripherial is allocated in 16 consecutives I/O 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

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 permitted 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.

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 -> 15^{th} 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.

SERIAL LINES; TIMER/COUNTER; I/O LINES; POWER MANAGEMENT

For further information on microprocessor internal peripheral device, please refer to specific documentation of the manufacturing company or to appendix B of this manual.

EXTERNAL DEVICES FOR GPC® 324

GPC® 324 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®** BUS connector. Even cards with **ABACO®** I/O BUS 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.

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.

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.

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 with Parallel interface

Intelligent user panel equipped with Fluorescent or LCD display, LEDs backlit, 20x2 or 20x4 characters; RS 232, RS 422, RS 485 or current loop serial line; serial E2 for set up and message. 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.

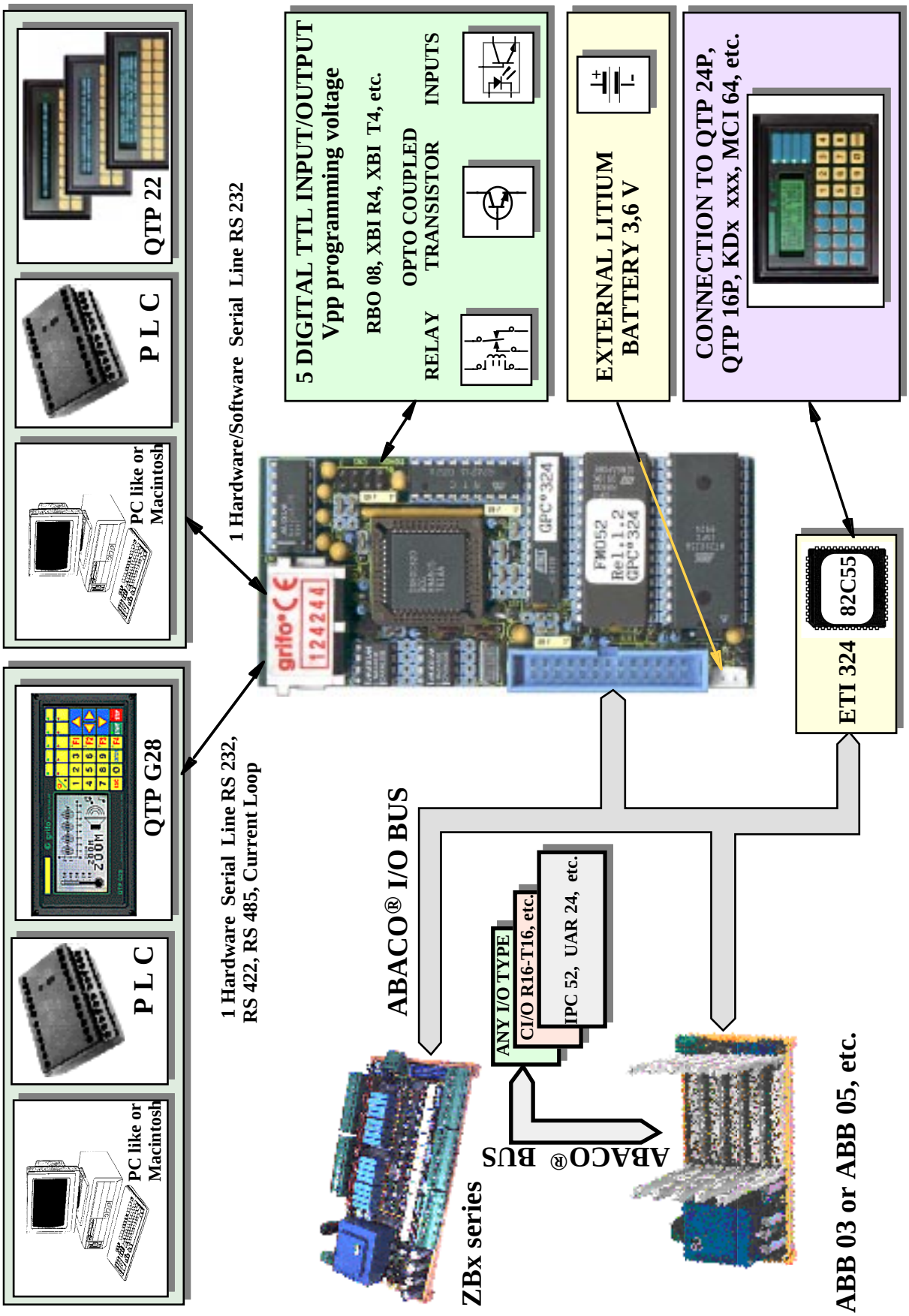


FIGURE 33: GPC®324 AVAILABLE CONNECTIONS DIAGRAM

QTP 16 - QTP 16P

Quick Terminal Panel 16 keys with Parallel interface

Intelligent user panel equipped with Fluorescent or LCD display, LEDs backlit, 20x2 or 20x4 characters; RS 232, RS 422, RS 485 or current loop serial line; serial E2 for set up and messages; buzzer manageable by software; 4 readable auxiliary opto in lines; power supply 5 Vdc. The QTP 16P is low cost no intelligent (passive) version. It is directly driven from 16 TTL I/O lines.

QTP G28

Quick Terminal Panel - LCD Graphic, 28 keys

LCD display 240x128 pixels, CFC backlit; Optocoupled RS 232 line and additional RS 232/422/485/ Current Loop line; CAN line controller; E^2 for set up; RTC and RAM lithium backed; primary graphic object; possibility of re-naming keys, LEDs and panel name; 28 keys and 16 LEDs with blinking attribute and buzzer manageable by software; Buzzer; built in power supply; reader of magnetic badge and relay option.

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.

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.

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.

DEB 01

Didactis Experimental Board

Supporting card for 16 TTL I/O lines use. It includes: 16 keys, 16 LEDs, 4 digits, 16 keys matrix keyboard, Centronics printer interface, LCD display and fluorescent display interface, **GPC® 68** I/O connector, field connection with screw terminal.

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; High level languages GDOS supported.

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.

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.

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.

IPC 52

Intelligent Peripheral Controller, 24 analogic input

This intelligent peripheral card acquires 24 indipendent analogic input lines: 8 PT 100 or PT 1000 sensors, 8 J,K,S,T termocouples, 8 analog input ±2Vdc or 4÷20mA; 16 bits + sign A/D section; 0.1 °C resolution; 32K RAM for local data logging; buzzer; 16 TTL I/O lines; 5 or 8 conversion per second; facility of networking up to 127 IPC 52 cards using serial line. BUS interfacing or through RS 232, RS 422, RS 485 or current loop line. Only 5Vdc power supply.

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® 324**.

Data book TEXAS INSTRUMENTS: *The TTL Data Book - SN54/74 Families*
Data book TEXAS INSTRUMENTS: *RS-422 and RS-485 Interface Circuits*

Data book NEC: *Memory Products*

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

Data book XICOR: *Data Book*

Data book PHILIPS: *80C51 - Based 8 Bit Microcontrollers*

Data book DALLAS SEMICONDUCTOR: *1992-1993 Product Data Book Supplement*

Data book INTEL: *8 Bit Embedded Microcontrollers*

Data book TOSHIBA: *Mos Memory Products*

Data sheet SEIKO EPSON: *RTC-62421Real Time Clock module*

For further information and upgrades please refer to specific internet web pages of the manufacturing companies.

APPENDIX A: CARD MECHANICAL MOUNTING

The **GPC® 324** can be physically mounted in two different manner. The first is the piggy back mounting (stack trough mode) that use the two connectors CN1 and CN5 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 (order code **BLOCK.100.50**); 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 long plastic container code is 414487 type RS/100 by Weidmuller and it can be ordered to **grifo®** as **.EXT-WMlll** options, where lll is the required lenght. By selecting this mounting the electric connection between **GPC® 324** and other peripheral cards is performed with flat cables that must be really short, as the **FLT.26+26 I/O** for **ABACO®** I/O BUS signals.
In the following figures are described the module dimensions with the connector positions and some immages that illustrate the connection modes.

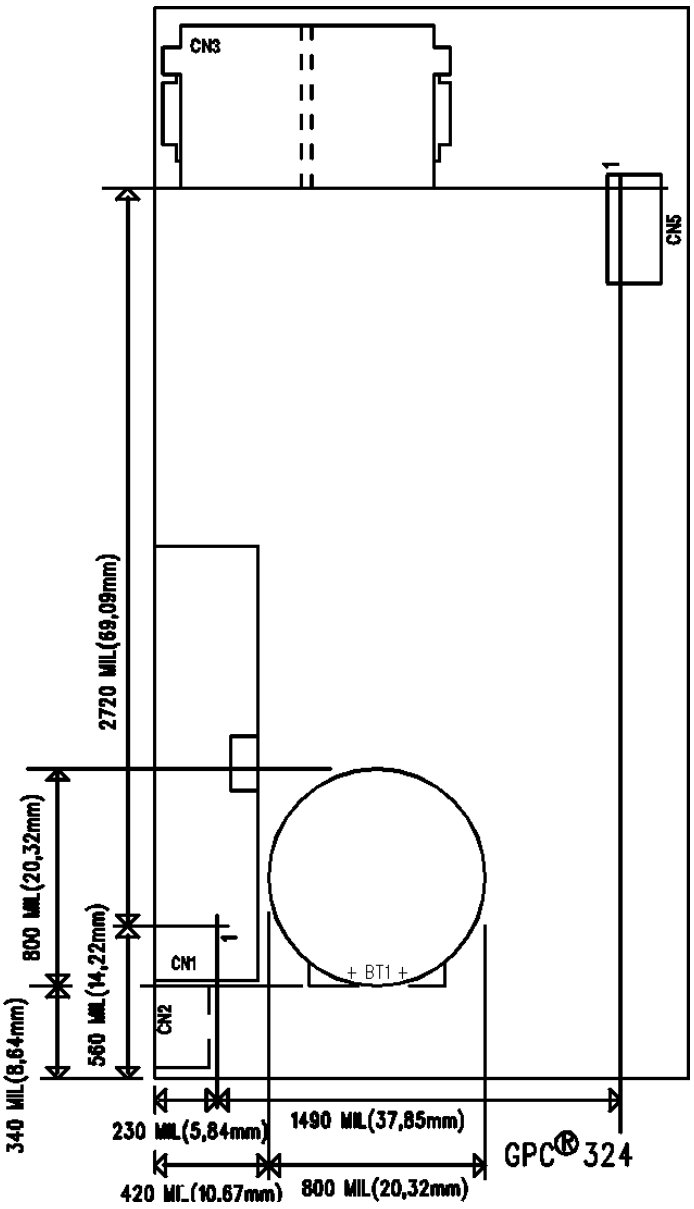


FIGURE A1: MODULE DIMENSION FOR PIGGY BACK MOUNTING

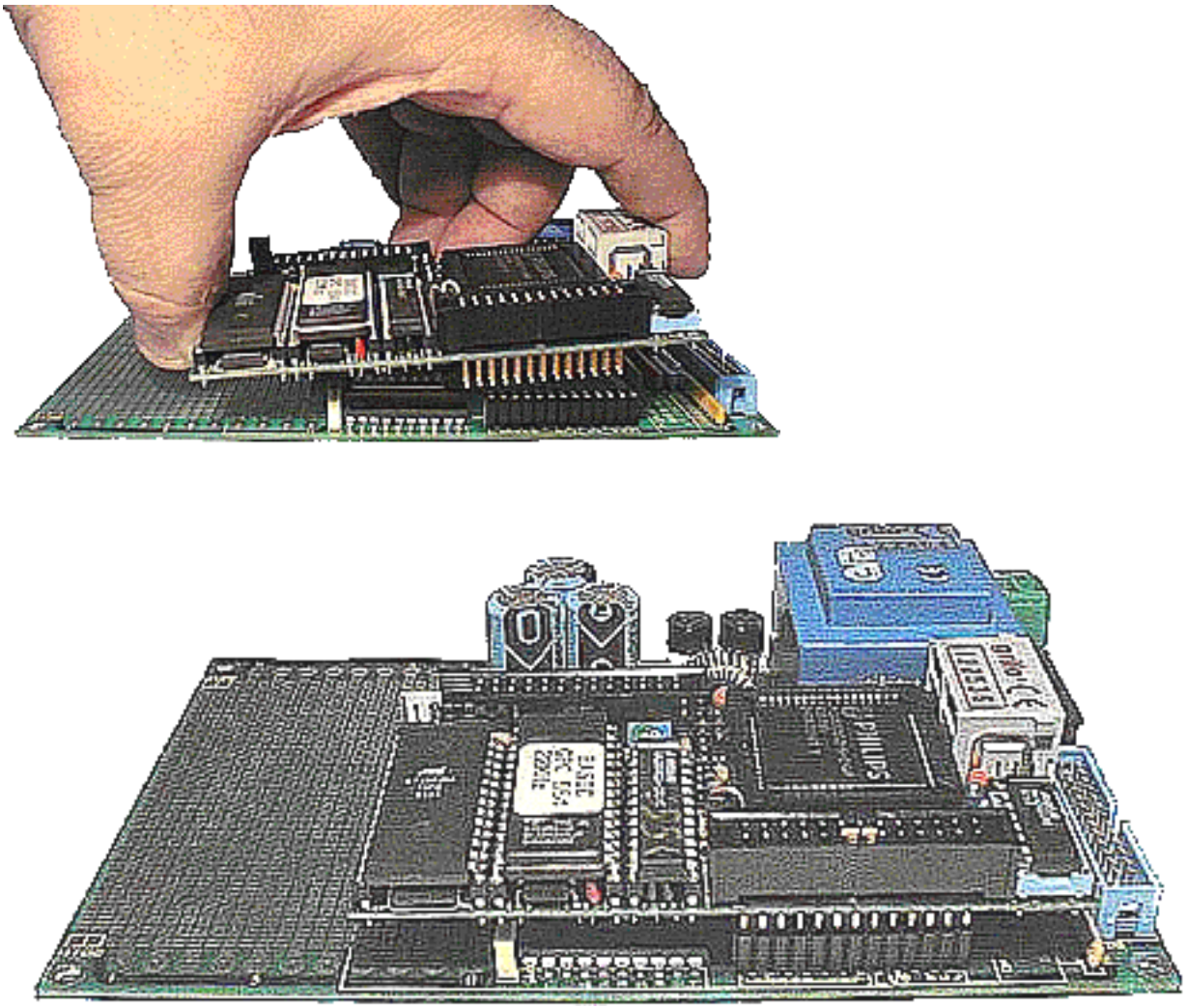


FIGURE A2: PIGGY BACK MOUNTING

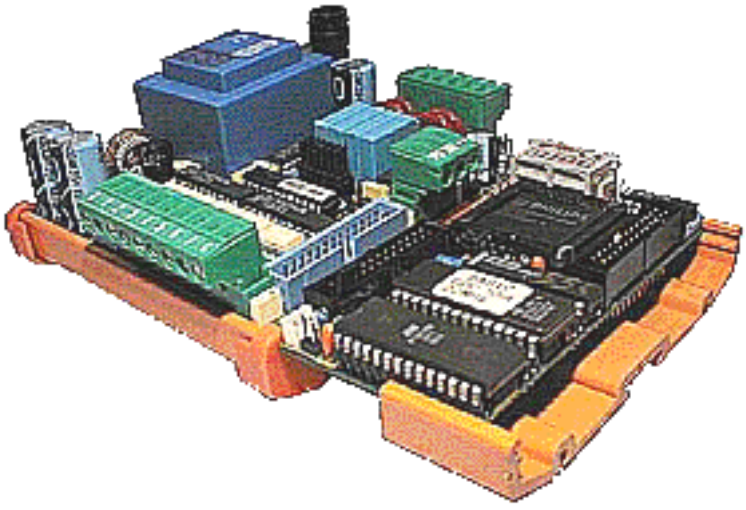


FIGURE A3: WEIDMULLER RAIL MOUNTING

APPENDIX B: ON BOARD DEVICES DESCRIPTION

µP 80C32

CMOS single-chip 8-bit microcontrollers 80C32/87C52

DESCRIPTION

The Philips 80C32/87C52 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity.

The 87C52 contains an 8k × 8 EPROM and the 80C32 is ROMless. Both contain a 256 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a six-source, two-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and on-chip oscillator and clock circuits.

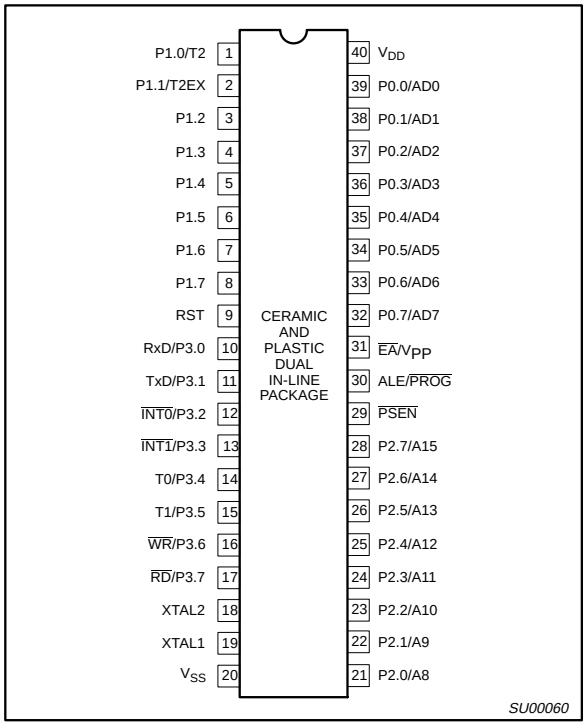
In addition, the 80C32/87C52 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

See 80C52/80C54/80C58 datasheet for ROM device specifications.

FEATURES

- 80C51 based architecture
- 8032 compatible
  - 8k × 8 EPROM (87C52)
  - ROMless (80C32)
  - 256 × 8 RAM
  - Three 16-bit counter/timers
  - Full duplex serial channel
  - Boolean processor
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- Three speed ranges:
  - 3.5 to 16MHz
  - 3.5 to 24MHz
  - 3.5 to 33MHz
- Five package styles
- Extended temperature ranges
- OTP package available

PIN CONFIGURATIONS



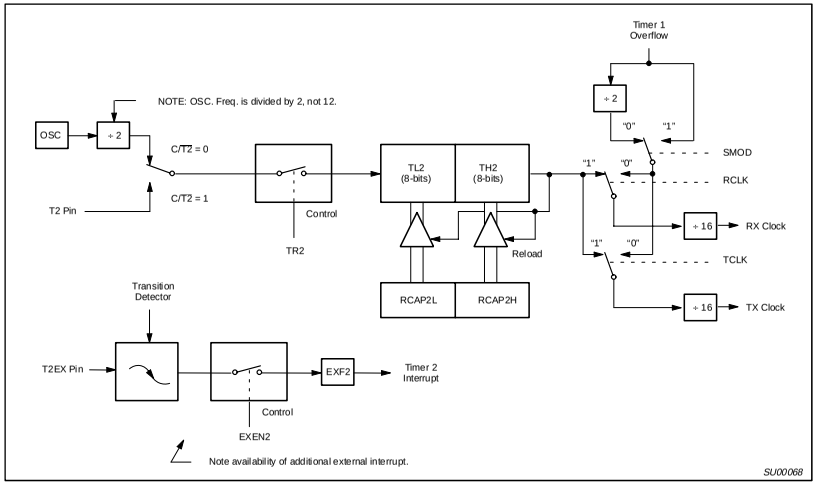


Figure 4. Timer 2 in Baud Rate Generator Mode

Table 2. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud rate generator
X	X	0	(off)

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK + TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the timer off (clear TR2) before accessing the Timer 2 or RCAP registers, in this case.

Timer/Counter 2 Set-up

Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set, separately, to turn the timer on. See Table 3 for set-up of timer 2 as a timer. See Table 4 for set-up of timer 2 as a counter.

Using Timer/Counter 2 to Generate Baud Rates

For this purpose, Timer 2 must be used in the baud rate generating mode. If Timer 2 is being clocked through pin T2 (P1.0) the baud rate is:

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

And if it is being clocked internally, the baud rate is:

$$\text{Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \quad [65536 \quad (\text{RCAP2H, RCAP2L})]}$$

To obtain the reload value for RCAP2H and RCA02L, the above equation can be rewritten as:

$$\text{RCAP2H, RCAP2L} \quad 65536 = \frac{\text{Oscillator Frequency}}{32 \quad \text{Baud Rate}}$$

BLOCK DIAGRAM

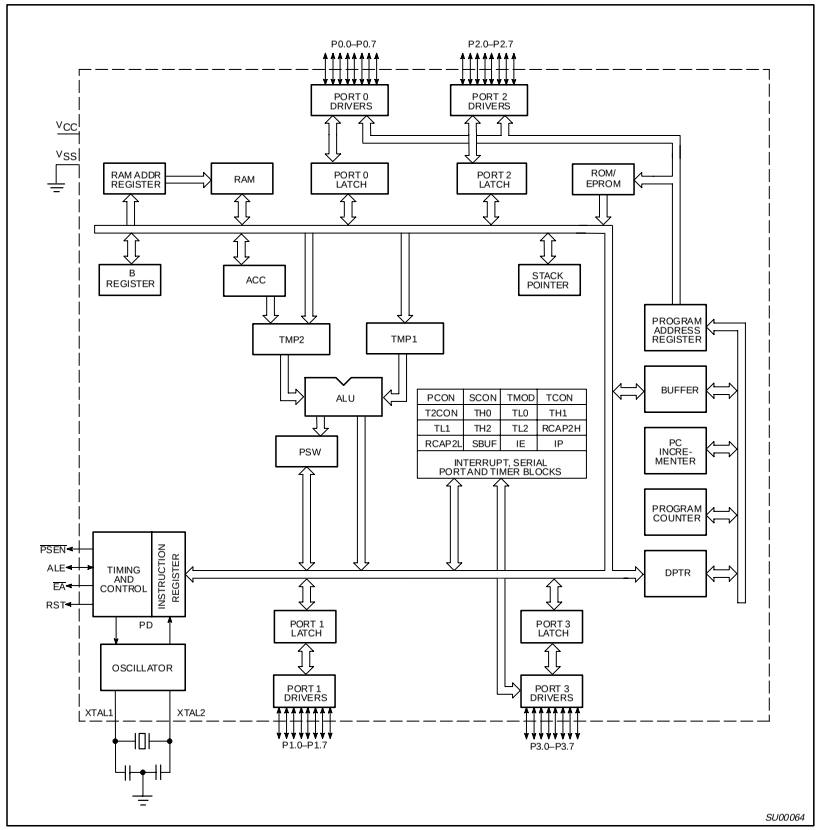


Table 1. 8XC52 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR:	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt enable	A8H	EA	–	ET2	ES	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt priority	B8H	–	–	PT2	PS	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	–	–	–	–	–	–	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	RxD	FFH
PCON[1]	Power control	87H	SMOD	–	–	–	GF1	GF0	PD	IDL	0xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	–	P	00H
RCAP2H#	Capture high	CBH									00H
RCAPL#	Capture low	CAH									00H
SBUF	Serial data buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*#	Timer 2 control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$	00H
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TH2#	Timer high 2	CDH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TL2#	Timer low 2	CCH									00H
TMOD	Timer mode	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	00H

* Bit addressable
\# SFRs are modified from or added to the 80C51 SFRs.
1. Bits GF1, GF0, PD, and IDL of the PCON register are not implemented in the NMOS 8XC52.

PIN DESCRIPTION

MNEMONIC	PIN NO. DIP	PIN NO. LCC	PIN NO. QFP	TYPE	NAME AND FUNCTION
V_{SS}	20	22	16	I	**Ground:** 0V reference.
V_{CC}	40	44	38	I	**Power Supply:** This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	**Port 0:** Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 87C52. External pull-ups are required during program verification.
P1.0–P1.7	1–8	2–9	40–44 1–3	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I_{IL}). Pins P1.0 and P1.1 also. Port 1 also receives the low-order address byte during program memory verification. Port 1 also serves alternate functions for timer 2:
	1	2	40	I	**T2 (P1.0):** Timer/counter 2 external count input.
	2	3	41	I	**T2EX (P1.1):** Timer/counter 2 trigger input.
P2.0–P2.7	21–28	24–31	18–25	I/O	**Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I_{IL}). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I_{IL}). Port 3 also serves the special features of the 80C51 family, as listed below:
	10	11	5	I	**RxD (P3.0):** Serial input port
	11	13	7	O	**TxD (P3.1):** Serial output port
	12	14	8	I	**$\overline{INT0}$ (P3.2):** External interrupt
	13	15	9	I	**$\overline{INT1}$ (P3.3):** External interrupt
	14	16	10	I	**T0 (P3.4):** Timer 0 external input
	15	17	11	I	**T1 (P3.5):** Timer 1 external input
	16	18	12	O	**$\overline{WR}$ (P3.6):** External data memory write strobe
	17	19	13	O	**$\overline{RD}$ (P3.7):** External data memory read strobe
RST	9	10	4	I	**Reset:** A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to V_{SS} permits a power-on reset using only an external capacitor to V_{CC}.
ALE/$\overline{PROG}$	30	33	27	I/O	**Address Latch Enable/Program Pulse:** Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input ($\overline{PROG}$) during EPROM programming.
$\overline{PSEN}$	29	32	26	O	**Program Store Enable:** The read strobe to external program memory. When the device is executing code from the external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory. $\overline{PSEN}$ is not activated during fetches from internal program memory.
$\overline{EA}$/V_{PP}	31	35	29	I	**External Access Enable/Programming Supply Voltage:** $\overline{EA}$ must be externally held low to enable the device to fetch code from external program memory locations 0000H to 1FFFH. If $\overline{EA}$ is held high, the device executes from internal program memory unless the program counter contains an address greater than 1FFFH. This pin also receives the 12.75V programming supply voltage (V_{PP}) during EPROM programming.
XTAL1	19	21	15	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	**Crystal 2:** Output from the inverting oscillator amplifier.

DIFFERENCES FROM THE 80C51

Special Function Registers

The special function register space is the same as the 80C51 except that the 80C32/87C52 contains the additional special function registers T2CON, RCAP2L, RCAP2H, TL2, and TH2. Since the standard 80C51 on-chip functions are identical in the 8XC52, the SFR locations, bit locations, and operation are likewise identical. The only exceptions are in the interrupt mode and interrupt priority SFRs (see Table 1).

Timer/Counters

In addition to timer/counters 0 and 1 of the 80C51, the 80C32/87C52 contains timer/counter 2. Like timers 0 and 1, timer 2 can operate as either an event timer or as an event counter. This is selected by bit C/T2 in the special function register T2CON (see Figure 1). It has three operating modes: capture, auto-load, and baud rate generator, which are selected by bits in the T2CON as shown in Table 2.

In the Capture Mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. (RCAP2L and RCAP2H are new special function registers in the 80C52.) In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2 like TF2 can generate an interrupt. The Capture Mode is illustrated in Figure 2.

In the auto-reload mode, there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. The auto-reload mode is illustrated in Figure 3.

The baud rate generation mode is selected by RCLK = 1 and/or TCLK = 1. It will be described in conjunction with the serial port.

Serial Port

The serial port of the 8XC52 is identical to that of the 80C51 except that counter/timer 2 can be used to generate baud rates.

In the 8XC52, Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (see Figure 1). Note that the baud rate for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows:

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The timer can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation (C/T2 = 0). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Normally, as a timer it would increment every machine cycle (thus at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (thus at 1/2 the oscillator frequency). In that case the baud rate is given by the formula:

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \quad [65536 \quad (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK or TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/T2	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

SU00065

Figure 1. Timer/Counter 2 (T2CON) Control Register

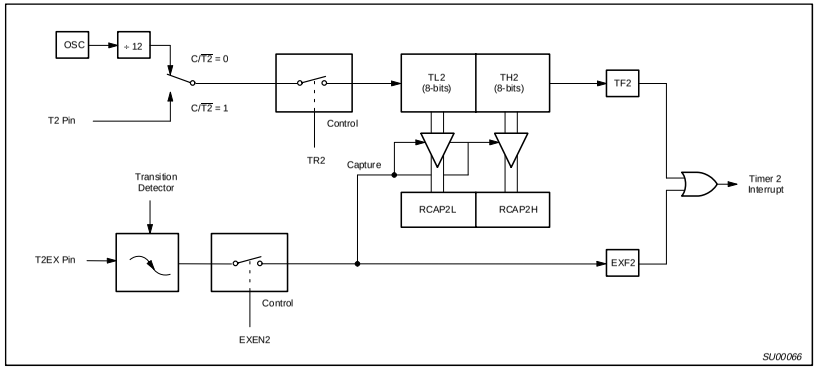


Figure 2. Timer 2 in Capture Mode

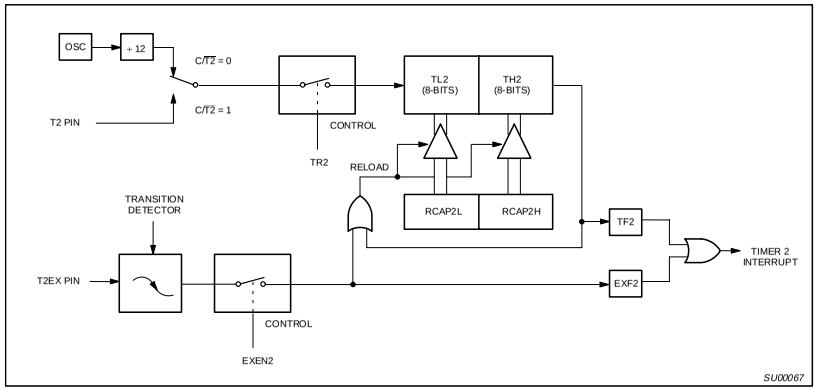


Figure 3. Timer 2 in Auto-Reload Mode

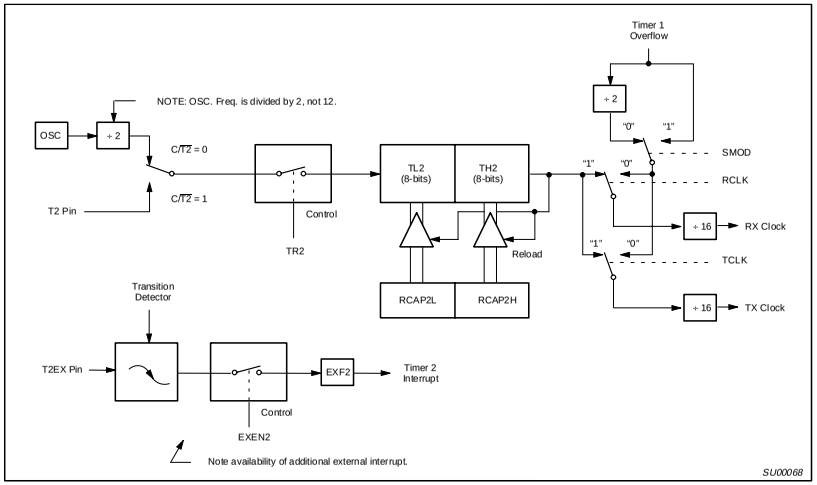


Figure 4. Timer 2 in Baud Rate Generator Mode

Table 2. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud rate generator
X	X	0	(off)

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK + TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the timer off (clear TR2) before accessing the Timer 2 or RCAP registers, in this case.

Timer/Counter 2 Set-up

Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set, separately, to turn the timer on. See Table 3 for set-up of timer 2 as a timer. See Table 4 for set-up of timer 2 as a counter.

Using Timer/Counter 2 to Generate Baud Rates

For this purpose, Timer 2 must be used in the baud rate generating mode. If Timer 2 is being clocked through pin T2 (P1.0) the baud rate is:

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

And if it is being clocked internally, the baud rate is:

$$\text{Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

To obtain the reload value for RCAP2H and RCA02L, the above equation can be rewritten as:

$$\text{RCAP2H, RCAP2L} = 65536 - \frac{\text{Oscillator Frequency}}{32 \times \text{Baud Rate}}$$

Interrupts

The 80C32/87C52 has 6 interrupt sources. All except TF2 and EXF2 are identical sources to those in the 80C51.

The Interrupt Enable Register and the Interrupt Priority Register are modified to include the additional 80C32/87C52 interrupt sources. The operation of these registers is identical to the 80C51.

In the 80C32/87C52, the Timer 2 Interrupt is generated by the logical OR of TF2 and EXF2. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it has been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software.

The interrupt vector addresses and the interrupt priority for requests in the same priority level are given in the following:

Source	Vector Address	Priority Within Level
1. IE0	0003H	(highest)
2. TF0	000BH	
3. IE1	0013H	
4. TF1	001BH	
5. RI + TI	0023H	
6. TF2 + EXF2	002BH	(lowest)

Note that they are identical to those in the 80C51 except for the addition of the Timer 2 (TF1 and EXF2) interrupt at 002BH and at the lowest priority within a level.

Table 3. Timer 2 as a Timer

MODE	T2CON	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit Auto-Reload	00H	08H
16-bit Capture	01H	09H
Baud rate generator receive and transmit same baud rate	34H	36H
Receive only	24H	26H
Transmit only	14H	16H

Table 4. Timer 2 as a Counter

MODE	TMOD	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit	02H	0AH
Auto-Reload	03H	0BH

NOTES:
1. Capture/reload occurs only on timer/counter overflow.
2. Capture/reload occurs on timer/counter overflow and a 1-to-0 transition on T2EX (P1.1) pin except when timer 2 is used in the baud rate generator mode.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the Logic Symbol, page 4.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure a good power-up reset, the RST pin must be high long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles.

IDLE MODE

In idle mode, the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

POWER-DOWN MODE

In the power-down mode, the oscillator is stopped and the instruction to invoke power-down is the last instruction executed. Only the contents of the on-chip RAM are preserved. A hardware reset is the only way to terminate the power-down mode. the control bits for the reduced power modes are in the special function register PCON.

DESIGN CONSIDERATIONS

At power-on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up.

Table 5 shows the state of I/O ports during low current operating modes.

As a precaution to coming out of an unexpected power down, $\overline{INT0}$ and $\overline{INT1}$ should be disabled prior to enterring power down.

Table 5. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

DALLAS SEMICONDUCTOR

DS80C320
High–Speed Micro

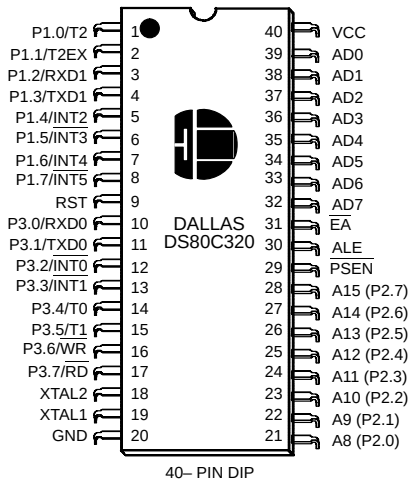
FEATURES

- 80C32–Compatible
  - Pin–compatible
  - Standard 8051 instruction set
  - Four 8–bit I/O ports
  - Three 16–bit timer/counters
  - 256 bytes scratchpad RAM
  - Multiplexed address/data bus
  - Addresses 64KB ROM and 64KB RAM
- High–speed architecture
  - 4 clocks/machine cycle (8032=12)
  - Wasted cycles removed
  - Runs DC to 33 MHz clock rates
  - Single–cycle instruction in 121 ns
  - Uses less power for equivalent work
  - Dual data pointer
  - Optional variable length MOVX to access fast/slow RAM /peripherals
- High integration controller includes:
  - Power–fail reset
  - Programmable Watchdog timer
  - Early–warning power–fail interrupt
- Two full–duplex hardware serial ports
- 13 total interrupt sources with six external
- Available in 40–pin DIP, 44–pin PLCC and TQFP

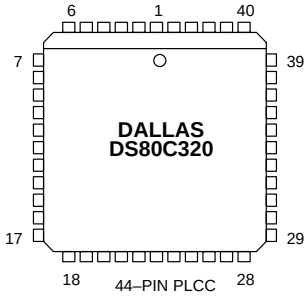
DESCRIPTION

The DS80C320 is a fast 80C31/80C32–compatible microcontroller. Wasted clock and memory cycles have been removed using a redesigned processor core. As a result, every 8051 instruction is executed between 1.5 and 3 times faster than the original for the same crystal speed. Typical applications will see a speed improvement of 2.5 times using the same code and same crystal. The DS80C320 offers a maximum crystal rate of 33 MHz, resulting in apparent execution speeds of 82.5 MHz (approximately 2.5X).

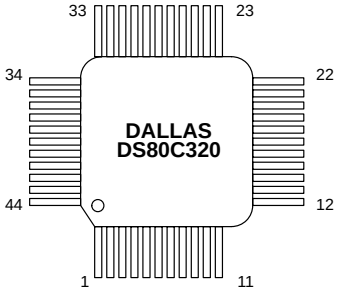
PIN ASSIGNMENT



40– PIN DIP



44–PIN PLCC



44–PIN TQFP

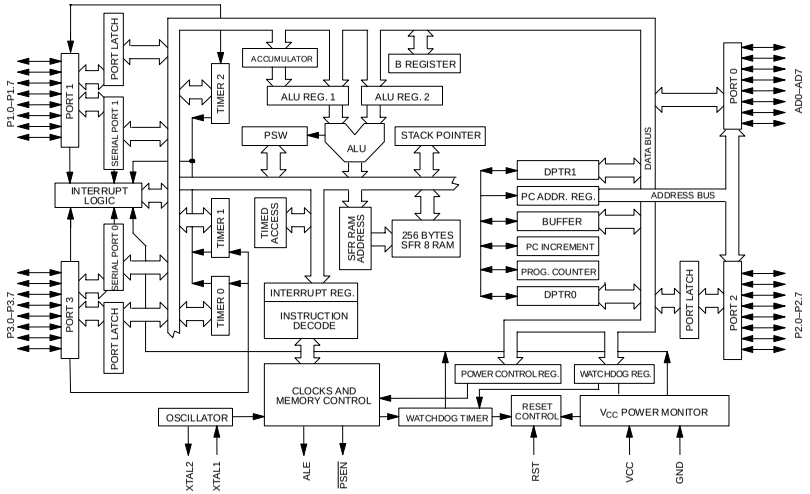
The DS80C320 is pin compatible with all three packages of the standard 80C32 and offers the same timer/counters, serial port, and I/O ports. In short, the DS80C320 is extremely familiar to 8051 users but provides the speed of a 16–bit processor.

The DS80C320 provides several extras in addition to greater speed. These include a second full hardware serial port, seven additional interrupts, programmable watchdog timer, power–fail interrupt and reset. The DS80C320 also provides dual data pointers (DPTRs) to speed block data memory moves. It can also adjust the speed of off–chip data memory access to between two and nine machine cycles for flexibility in selecting memory and peripherals.

ORDERING INFORMATION

PART NUMBER	PACKAGE	MAX CLOCK SPEED	TEMPERATURE RANGE
DS80C320–MCG	40–pin plastic DIP	25 MHz	0°C to +70°C
DS80C320–QCG	44–pin PLCC	25 MHz	0°C to +70°C
DS80C320–ECG	44–pin TQFP	25 MHz	0°C to +70°C
DS80C320–MNG	40–pin plastic DIP	25 MHz	–40°C to +85°C
DS80C320–QNG	44–pin PLCC	25 MHz	–40°C to +85°C
DS80C320–ENG	44–pin TQFP	25 MHz	–40°C to +85°C
DS80C320–MCL	40–pin plastic DIP	33 MHz	0°C to +70°C
DS80C320–QCL	44–pin PLCC	33 MHz	0°C to +70°C
DS80C320–ECL	44–pin TQFP	33 MHz	0°C to +70°C
DS80C320–MNL	40–pin plastic DIP	33 MHz	–40°C to +85°C
DS80C320–QNL	44–pin PLCC	33 MHz	–40°C to +85°C
DS80C320–ENL	44–pin TQFP	33 MHz	–40°C to +85°C

DS80C320 BLOCK DIAGRAM Figure 1



HIGH–SPEED OPERATION

The DS80C320 is built around a high speed 80C32 compatible core. Higher speed comes not just from increasing the clock frequency, but from a newer, more efficient design.

In this updated core, dummy memory cycles have been eliminated. In a conventional 80C32, machine cycles are generated by dividing the clock frequency by 12. In the DS80C320, the same machine cycle is performed in 4 clocks. Thus the fastest instruction, 1 machine cycle, is executed three times faster for the same crystal frequency. Note that these are identical instructions. A comparison of the timing differences is shown in Figure 2. The majority of instructions on the DS80C320 will see the full 3 to 1 speed improvement. Some instructions will get between 1.5 and 2.4 X improvement. Note that all instructions are faster than the original 80C51. Table 2 below shows a summary of the instruction set including the speed.

The numerical average of all opcodes is approximately a 2.5 to 1 speed improvement. Individual programs will be affected differently, depending on the actual instructions used. Speed sensitive applications would make the most use of instructions that are three times faster. However, the sheer number of 3 to 1 improved opcodes makes dramatic speed improvements likely for any code. When these architecture improvements are combined with 0.8 µm CMOS, the result is a single cycle instruction execution in 160 ns. The Dual Data Pointer feature also allows the user to eliminate wasted instructions when moving blocks of memory.

INSTRUCTION SET SUMMARY

All instructions in the DS80C320 perform the same functions as their 80C32 counterparts. Their affect on bits, flags, and other status functions is identical. However, the timing of each instruction is different. This applies both in absolute and relative number of clocks.

For absolute timing of real–time events, the timing of software loops will need to be calculated using the table below. However, counter/timers default to run at the older 12 clocks per increment. Therefore, while software runs at higher speed, timer–based events need no modification to operate as before. Timers can be set to run at 4 clocks per increment cycle to take advantage of higher speed operation.

The relative time of two instructions might be different in the new architecture than it was previously. For example, in the original architecture, the “MOVX A, @DPTR” instruction and the “MOV direct, direct” instruction used two machine cycles or 24 oscillator cycles. Therefore, they required the same amount of time. In the DS80C320, the MOVX instruction can be done in two machine cycles or eight oscillator cycles but the “MOV direct, direct” uses three machine cycles or 12 oscillator cycles. While both are faster than their original counterparts, they now have different execution times from each other. This is because in most cases, the DS80C320 uses one cycle for each byte. The user concerned with precise program timing should examine the timing of each instruction for familiarity with the changes. Note that a machine cycle now requires just four clocks, and provides one ALE pulse per cycle. Many instructions require only one cycle, but some require five. In the original architecture, all were one or two cycles except for MUL and DIV.

INSTRUCTION SET SUMMARY Table 2

Legends:

A	–	Accumulator
Rn	–	Register R7–R0
direct	–	Internal Register address
@Ri	–	Internal Register pointed–to by R0 or R1 (except MOVX)
rel	–	2’s complement offset byte
bit	–	direct bit–address
#data	–	8–bit constant
#data 16	–	16–bit constant
addr 16	–	16–bit destination address
addr 11	–	11–bit destination address

INSTRUCTION	BYTE	OSCILLATOR CYCLES	INSTRUCTION	BYTE	OSCILLATOR CYCLES
Arithmetic Instructions:					
ADD A, Rn	1	4	INC A	1	4
ADD A, direct	2	8	INC Rn	1	4
ADD A, @Ri	1	4	INC direct	2	8
ADD A, #data	2	8	INC @Ri	1	4
ADDC A, Rn	1	4	INC DPTR	1	12
ADDC A, direct	2	8	DEC A	1	4
ADDC A, @Ri	1	4	DEC Rn	1	4
ADDC A, #data	2	8	DEC direct	2	8
SUBB A, Rn	1	4	DEC @Ri	1	4
SUBB A, direct	2	8	MUL AB	1	20
SUBB A, @Ri	1	4	DIV AB	1	20
SUBB A, #data	2	8	DA A	1	4
Logical Instructions:					
ANL A, Rn	1	4	XRL A, Rn	1	4
ANL A, direct	2	8	XRL A, direct	2	8
ANL A, @Ri	1	4	XRL A, @Ri	1	4
ANL A, #data	2	8	XRL A, #data	2	8
ANL direct, A	2	8	XRL direct, A	2	8
ANL direct, #data	3	12	XRL direct, #data	3	12
ORL A, Rn	1	4	CLR A	1	4
ORL A, direct	2	8	CPL A	1	4
ORL A, @Ri	1	4	RL A	1	4
ORL A, #data	2	8	RLC A	1	4
ORL direct, A	2	8	RR A	1	4
ORL direct, #data	3	12	RRC A	1	4
			SWAP A	1	4
Data Transfer Instructions:					
MOV A, Rn	1	4	MOVC A, @A+DPTR	1	12
MOV A, direct	2	8	MOVC A, @A+PC	1	12
MOV A, @Ri	1	4	MOVX A, @Ri	1	8–36 *
MOV A, #data	2	8	MOVX A, @DPTR	1	8–36 *
MOV Rn, A	1	4	MOVX @Ri, A	1	8–36 *
MOV Rn, direct	2	8	MOVX @DPTR, A	1	8–36 *
MOV Rn, #data	2	8	PUSH direct	2	8
MOV direct, A	2	8	POP direct	2	8
MOV direct, Rn	2	8	XCH A, Rn	1	4
MOV direct1, direct2	3	12	XCH A, direct	2	8
MOV direct, @Ri	2	8	XCH A, @Ri	1	4
MOV direct, #data	3	12	XCHD A, @Ri	1	4
MOV @Ri, A	1	4			
MOV @Ri, direct	2	8			
MOV @Ri, #data	2	8			
MOV DPTR, #data 16	3	12			

*User Selectable

Bit Manipulation Instructions:					
CLR C	1	4	ANL C, bit	2	8
CLR bit	2	8	ANL C, $\overline{bit}$	2	8
SETB C	1	4	ORL C, bit	2	8
SETB bit	2	8	ORL C, $\overline{bit}$	2	8
CPL C	1	4	MOV C, bit	2	8
CPL bit	2	8	MOV bit, C	2	8
Program Branching Instructions:					
ACALL addr 11	2	12	CJNE A, direct, rel	3	16
LCALL addr 16	3	16	CJNE A, #data, rel	3	16
RET	1	16	CJNE Rn, #data, rel	3	16
RETI	1	16	CJNE Ri, #data, rel	3	16
AJMP addr 11	2	12	NOP	1	4
LJMP addr 16	3	16	JC rel	2	12
SJMP rel	2	12	JNC rel	2	12
JMP @A+DPTR	1	12	JB bit, rel	3	16
JZ rel	2	12	JNB bit, rel	3	16
JNZ rel	2	12	JBC bit, rel	3	16
DJNZ Rn, rel	2	12			
DJNZ direct, rel	3	16			

The table above shows the speed for each class of instruction. Note that many of the instructions have multiple opcodes. There are 255 opcodes for 111 instructions. Of the 255 opcodes, 159 are three times faster than the original 80C32. While a system that emphasizes those instructions will see the most improvement, the large total number that receive a 3 to 1 improvement assure a dramatic speed increase for any system. The speed improvement summary is provided below.

SPEED ADVANTAGE SUMMARY

#Opcodes	Speed Improvement
159	3.0 x
51	1.5 x
43	2.0 x
2	2.4 x
255	Average: 2.5

MEMORY ACCESS

The DS80C320 contains no on–chip ROM and 256 bytes of scratchpad RAM. Off–chip memory is accessed using the multiplexed address/data bus on P0 and the MSB address on P2. A typical memory connection is shown in Figure 3. Timing diagrams are provided in the Electrical Specifications. Program memory (ROM) is accessed at a fixed rate determined by the crystal frequency and the actual instructions. As mentioned above, an instruction cycle requires four clocks. Data memory (RAM) is accessed according to a variable speed MOVX instruction as described below.

DATA MEMORY CYCLE STRETCH VALUES Table 3

CKCON.2–0 MD2	MD1	MD0	MEMORY CYCLES	$\overline{RD}$ or $\overline{WR}$ STROBE WIDTH IN CLOCKS	STROBE WIDTH TIME @ 25 MHz
0	0	0	2	2	80 ns
0	0	1	3 (default)	4	160 ns
0	1	0	4	8	320 ns
0	1	1	5	12	480 ns
1	0	0	6	16	640 ns
1	0	1	7	20	800 ns
1	1	0	8	24	960 ns
1	1	1	9	28	1120 ns

DUAL DATA POINTER

Data memory block moves can be accelerated using the DS80C320 Dual Data Pointer (DPTR). The standard 8032 DPTR is a 16–bit value that is used to address off–chip data RAM or peripherals. In the DS80C320, the standard data pointer is called DPTR 0 and is located at SFR addresses 82h and 83h. These are the standard locations. No modification of standard code is needed to use DPTR. The new DPTR is located at SFR 84h and 85h and is called DPTR1. The DPTR Select bit (DPS) chooses the active pointer and is located at the LSB of the SFR location 86h. No other bits in register 86h have any effect and are set to 0. The user switches between data pointers by toggling the LSB of register 86h. The increment (INC) instruction is the fastest way to accomplish this. All DPTR–related instructions use the currently selected DPTR for any activity. Therefore only one instruction is required to switch from a source to a destination address. Using the Dual–Data Pointer saves code from needing to save source and destination addresses when doing a block move. Once loaded, the software simply switches between DPTR and 1. The relevant register locations are as follows.

DPL	82h	Low byte original DPTR
DPH	83h	High byte original DPTR
DPL1	84h	Low byte new DPTR
DPH1	85h	High byte new DPTR
DPS	86h	DPTR Select (LSB)

Sample code listed below illustrates the saving from using the dual DPTR. The example program was original code written for an 8051 and requires a total of 1869 machine cycles on the DS80C320. This takes 299 µs to execute at 25 MHz. The new code using the Dual DPTR requires only 1097 machine cycles taking 175.5 µs. The Dual DPTR saves 772 machine cycles or 123.5 µs for a 64 byte block move. Since each pass through the loop saves 12 machine cycles when compared to the single DPTR approach, larger blocks gain more efficiency using this feature.

64 BYTE BLOCK MOVE WITHOUT DUAL DATA POINTER

```
; SH and SL are high and low byte source address.
; DH and DL are high and low byte of destination address.
                                                              # CYCLES

MOV     R5, #64d        ; NUMBER OF BYTES TO MOVE                   2
MOV     DPTR, #SHSL     ; LOAD SOURCE ADDRESS                       3
MOV     R1, #SL         ; SAVE LOW BYTE OF SOURCE                   2
MOV     R2, #SH         ; SAVE HIGH BYTE OF SOURCE                  2
MOV     R3, #DL         ; SAVE LOW BYTE OF DESTINATION              2
MOV     R4, #DH         ; SAVE HIGH BYTE OF DESTINATION             2

MOVE:
; THIS LOOP IS PERFORMED THE NUMBER OF TIMES LOADED INTO R5, IN THIS EXAMPLE 64

MOVX    A, @DPTR        ; READ SOURCE DATA BYTE                     2
MOV     R1, DPL         ; SAVE NEW SOURCE POINTER                   2
MOV     R2, DPH         ;                                           2
```



```
MOV     DPL, R3         ; LOAD NEW DESTINATION                      2
MOV     DPH, R4         ;                                           2
MOVX    @DPTR, A        ; WRITE DATA TO DESTINATION                 2
INC     DPTR            ; NEXT DESTINATION ADDRESS                  3
MOV     R3, DPL         ; SAVE NEW DESTINATION POINTER              2
MOV     R4, DPH         ;                                           2
MOV     DPL, R1         ; GET NEW SOURCE POINTER                    2
MOV     DPH, R2         ;                                           2
INC     DPTR            ; NEXT SOURCE ADDRESS                       3
DJNZ    R5, MOVE        ; FINISHED WITH TABLE?                      3
```

64 BYTE BLOCK MOVE WITH DUAL DATA POINTER

```
; SH and SL are high and low byte source address.
; DH and DL are high and low byte of destination address.
; DPS is the data pointer select. Reset condition is DPS=0, DPTR0 is selected.
                                                              # CYCLES

EQU     DPS, #86h       ; TELL ASSEMBLER ABOUT DPS

MOV     R5, #64         ; NUMBER OF BYTES TO MOVE                   2
MOV     DPTR, #DHDL     ; LOAD DESTINATION ADDRESS                  3
INC     DPS             ; CHANGE ACTIVE DPTR                        2
MOV     DPTR, #SHSL     ; LOAD SOURCE ADDRESS                       2

MOVE:
; THIS LOOP IS PERFORMED THE NUMBER OF TIMES LOADED INTO R5, IN THIS EXAMPLE 64

MOVX    A, @DPTR        ; READ SOURCE DATA BYTE                     2
INC     DPS             ; CHANGE DPTR TO DESTINATION                2
MOVX    @DPTR, A        ; WRITE DATA TO DESTINATION                 2
INC     DPTR            ; NEXT DESTINATION ADDRESS                  3
INC     DPS             ; CHANGE DATA POINTER TO SOURCE             2
INC     DPTR            ; NEXT SOURCE ADDRESS                       3
DJNZ    R5, MOVE        ; FINISHED WITH TABLE?                      3
```

PERIPHERAL OVERVIEW

Peripherals in the DS80C320 are accessed using Special Function Registers (SFRs). The DS80C320 provides several of the most commonly needed peripheral functions in microcomputer–based systems. These functions are new to the 80C32 family and include a second serial port, Power–fail Reset, Power–fail Interrupt, and a programmable Watchdog Timer. These are described below, and more details are available in the High–Speed Microcontroller User's Guide.

SERIAL PORTS

The DS80C320 provides a serial port (UART) that is identical to the 80C32. Many applications require serial communication with multiple devices. Therefore the DS80C320 provides a second hardware serial port that is a full duplicate of the standard one. It optionally uses pins P1.2 (RXD1) and P1.3 (TXD1). This port has duplicate control functions included in new SFR locations. The second serial port operates in a comparable manner with the first. Both can operate simultaneously but can be at different baud rates.

The second serial port has similar control registers (SCON1 at C0h, SBUF1 at C1h) to the original. One difference is that for timer based baud rates, the original serial port can use Timer 1 or Timer 2 to generate baud rates. This is selected via SFR bits. The new serial port can only use Timer 1.

TIMER RATE CONTROL

One important difference exists between the DS80C320 and 80C32 regarding timers. The original 80C32 used a 12 clock per cycle scheme for timers and consequently for some serial baud rates (depending on the mode). The DS80C320 architecture normally runs using 4 clocks per cycle. However, in the area of timers, the DS80C320 will default to a 12 clock per cycle

scheme on a reset. This allows existing code with real–time dependencies such as baud rates to operate properly. If an application needs higher speed timers or serial baud rates, the timers can be set to run at the 4 clock rate.

The Clock Control register (CKCON – 8Eh) determines these timer speeds. When the relevant CKCON bit is a logic 1, the DS80C320 uses 4 clocks per cycle to generate timer speeds. When the control bit is set to a 0, the DS80C320 uses 12 clocks for timer speeds. The reset condition is a 0. CKCON.5 selects the speed of Timer 2. CKCON.4 selects Timer 1 and CKCON.3 selects Timer 0. Note that unless a user desires very fast timing, it is unnecessary to alter these bits. Note that the timer controls are independent.

POWER FAIL RESET

The DS80C320 incorporates a precision band–gap voltage reference to determine when V_{CC} is out–of–tolerance. While powering up, internal circuits will hold the DS80C320 in a reset state until V_{CC} rises above the V_{RST} reset threshold. Once V_{CC} is above this level, the oscillator will begin running. An internal reset circuit will then count 65536 clocks to allow time for power and the oscillator to stabilize. The microcontroller will then exit the reset condition. No external components are needed to generate a power on reset. During power down or during a severe power glitch, as V_{CC} falls below V_{RST}, the microcontroller will also generate its own reset. It will hold the reset condition as long as power remains below the threshold. This reset will occur automatically, needing no action from the user or from the software. Refer to the Electrical Specifications for the exact value of V_{RST}.

POWER FAIL INTERRUPT

The same reference that generates a precision reset threshold can also generate an optional early warning Power–fail Interrupt (PFI). When enabled by the application software, this interrupt always has the highest priority. On detecting that the V_{CC} has dropped below V_{PFW} and that the PFI is enabled, the processor will vector to ROM address 0033h. The PFI enable is located in the Watchdog Control SFR (WDCON – D8h). Setting WDCON.5 to a logic one will enable the PFI. The application software can also read a flag at WDCON.4. This bit is set when a PFI condition has occurred. The flag is independent of the interrupt enable and software must manually clear it.

WATCHDOG TIMER

For applications that can not afford to run out–of–control, the DS80C320 incorporates a programmable Watchdog Timer circuit. It resets the uC if software fails to reset the Watchdog before the selected time interval has elapsed. The user selects one of four time–out values. After enabling the Watchdog, software must reset the timer prior to expiration of the interval, or the CPU will be reset. Both the Watchdog Enable and the Watchdog Reset bits are protected by a “Timed Access” circuit. This prevents accidentally clearing the Watchdog. Time–out values are precise since they are related to the crystal frequency as shown below in Table 4. For reference, the time periods at 25 MHz are also shown.

The DS80C320 Watchdog also provides a useful option for systems that may not require a reset. If enabled, then 512 clocks before giving a reset, the Watchdog will give an interrupt. The interrupt can also serve as a convenient time–base generator, or be used to wake–up the processor from Idle mode. The Watchdog function is controlled in the Clock Control (CKCON – 8Eh), Watchdog Control (WDCON – D8h), and Extended Interrupt Enable (EIE – E8h) SFRs. CKCON.7 and CKCON.6 are called WD1 and WD0 respectively and are used to select the Watchdog time–out period as shown in Table 4.

WATCHDOG TIME–OUT VALUES Table 4

WD1	WD0	INTERRUPT TIME–OUT	TIME (@25 MHz)	RESET TIME–OUT	TIME (@25 MHz)
0	0	2^{17} clocks	5.243 ms	2^{17} + 512 clocks	5.263 ms
0	1	2^{20} clocks	41.94 ms	2^{20} + 512 clocks	41.96 ms
1	0	2^{23} clocks	335.54 ms	2^{23} + 512 clocks	335.56 ms
1	1	2^{26} clocks	2684.35 ms	2^{26} + 512 clocks	2684.38 ms

As shown above, the Watchdog Timer uses the crystal frequency as a time base. A user selects one of four counter values to determine the time–out. These clock counter lengths are 2^{17}= 131,072 clocks; 2^{20} = 1,048,576; 2^{23} = 8,388,608 clocks; or 2^{26} = 67,108,864 clocks. The times shown in Table 4 above are with a 25 MHz crystal frequency. Note that once the counter chain has reached a conclusion, the optional interrupt is generated. Regardless of whether the user enables this interrupt, there are then 512 clocks left until a reset occurs. There are five control bits in special function registers that affect the Watchdog Timer and two status flags that report to the user.

WDIF (WDCON.3) is the interrupt flag that is set when there are 512 clocks remaining until a reset occurs. WTRF (WDCON.2) is the flag that is set when a Watchdog reset has occurred. This allows the application software to determine the source of a reset.

EWT (WDCON.1) is the enable for the Watchdog Timer. Software sets this bit to enable the timer. The bit is protected by Timed Access discussed below. RWT (WDCON.0) is the bit that software uses to restart the Watchdog Timer. Setting this bit restarts the timer for another full interval. Application software must set this bit prior to the time–out. As mentioned previously, WD1 and 0 (CKCON .7 and 6) select the time–out. Finally, the Watchdog Interrupt is enabled using EWDI (EIE.4). The Special Function Register map is shown below.

INTERRUPTS

The DS80C320 provides 13 sources of interrupt with three priority levels. The Power–fail Interrupt (PFI), if enabled, always has the highest priority. There are two remaining user selectable priorities: high and low. If two interrupts that have the same priority occur simultaneously, the natural precedence given below determines which is a acted upon. Except for the PFI, all interrupts that are new to the 8051 family have a lower natural priority than the originals.

INTERRUPT PRIORITY Table 5

NAME	DESCRIPTION	VECTOR	NATURAL PRIORITY	OLD/NEW
PFI	Power Fail Interrupt	33h	1	NEW
$\overline{\text{INT0}}$	External Interrupt 0	03h	2	OLD
TF0	Timer 0	0Bh	3	OLD
$\overline{\text{INT1}}$	External Interrupt 1	13h	4	OLD
TF1	Timer 1	1Bh	5	OLD
SCON0	TI0 or RI0 from serial port 0	23h	6	OLD
TF2	Timer 2	2Bh	7	OLD
SCON1	TI1 or RI1 from serial port 1	3Bh	8	NEW
INT2	External Interrupt 2	43h	9	NEW
$\overline{\text{INT3}}$	External Interrupt 3	4Bh	10	NEW
INT4	External Interrupt 4	53h	11	NEW
$\overline{\text{INT5}}$	External Interrupt 5	5Bh	12	NEW
WDTI	Watchdog Time–out Interrupt	63h	13	NEW

POWER MANAGEMENT

The DS80C320 provides the standard Idle and power down (Stop) that are available on the standard 80C32. However the DS80C320 has enhancements that make these modes more useful, and allow more power saving.

The Idle mode is invoked by setting the LSB of the Power Control register (PCON – 87h). Idle will leave internal clocks, serial port and timer running. No memory access will be performed so power is dramatically reduced. Since clocks are running, the Idle power consumption is related to crystal frequency. It should be approximately 1/2 of the operational power. The CPU can exit the Idle state with any interrupt or a reset.

The power–down or Stop mode is invoked by setting the PCON.1 bit. Stop mode is a lower power state than Idle since it turns off all internal clocking. The I_{CC} of a standard Stop mode is approximately 1 µA but is specified in the Electrical Specifications. The CPU will exit Stop mode from an external interrupt or a reset condition.

Note that internally generated interrupts (timer, serial port, watchdog) are not useful since they require clocking activity.

IDLE MODE ENHANCEMENTS

A simple enhancement to Idle mode makes it substantially more useful. The innovation involves not the Idle mode itself, but the watchdog timer. As mentioned above, the Watchdog Timer provides an optional interrupt capability. This interrupt can provide a periodic interval timer to bring the DS80C320 out of Idle mode. This can be useful even if the Watchdog is not normally used. By enabling the Watchdog Timer and its interrupt prior to invoking Idle, a user can periodically come out of Idle perform an operation, then return to Idle until the next operation. This will lower the overall power consumption. When using the Watchdog Interrupt to cancel the Idle state, make sure to restart the Watchdog Timer or it will cause a reset.

STOP MODE ENHANCEMENTS

The DS80C320 provides two enhancements to the Stop mode. As documented above, the DS80C320 provides a band–gap reference to determine Power–fail Interrupt and Reset thresholds. The default state is that the band–gap reference is off when Stop mode is invoked. This allows the extremely low power state mentioned above. A user can optionally choose to have the band–gap enabled during Stop mode. This means that PFI and power–fail reset will be activated and are valid means for leaving Stop mode.

In Stop mode with the band–gap on, I_{CC} will be approximately 50 µA compared with 1 µA with the band–gap off. If a user does not require a Power–fail Reset or Interrupt while in Stop mode, the band–gap can remain turned off. Note that only the most power sensitive applications should turn off the band–gap, as this results in an uncontrolled power down condition.

The control of the band–gap reference is located in the Extended Interrupt Flag register (EXIF – 91h). Setting BGS (EXIF.0) to a one will leave the band–gap reference enabled during Stop mode. The default or reset condition is with the bit at a logic 0. This results in the band–gap being turned off during Stop mode. Note that this bit has no control of the reference during full power or Idle modes.

The second feature allows an additional power saving option. This is the ability to start instantly when exiting Stop mode. It is accomplished using an internal ring oscillator that can be used when exiting Stop mode in response to an interrupt. The benefit of the ring oscillator is as follows.

Using Stop mode turns off the crystal oscillator and all internal clocks to save power. This requires that the oscillator be restarted when exiting Stop mode. Actual start–up time is crystal dependent, but is normally at least 4 ms. A common recommendation is 10 ms. In an application that will wake–up, perform a short operation, then return to sleep, the crystal start–up can be longer than the real transaction. However, the ring oscillator will start instantly. The user can perform a simple operation and return to sleep before the crystal has even stabilized. If the ring is used to start and the processor remains running, hardware will automatically switch to the crystal once a power–on reset interval (65536 clocks) has expired. This value is used to guarantee stability even though power is not being cycled.

If the user returns to Stop mode prior to switching of crystal, then all clocks will be turned off again. The ring oscillator runs at approximately 4 MHz but will not be a precision value. No real–time precision operations (including serial communication) should be conducted during this ring period. Figure 7 shows how the operation would compare when using the ring, and when starting up normally. The default state is to come out of Stop mode without using the ring oscillator.

This function is controlled using the RGSL – Ring Select bit at EXIF.1 (EXIF – 91h). When EXIF.1 is set, the ring oscillator will be used to come out of Stop mode quickly. As mentioned above, the processor will automatically switch from the ring (if enabled) to the crystal after a delay of 65536 crystal clocks. For a 3.57 MHz crystal, this is approximately 18 ms. The processor sets a flag called RGMD – Ring Mode to tell software that the ring is being used. This bit at EXIF.2 will be a logic 1 when the ring is in use. No serial communication or precision timing should be attempted while this bit is set, since the operating frequency is not precise.

RING OSCILLATOR START–UP Figure 4

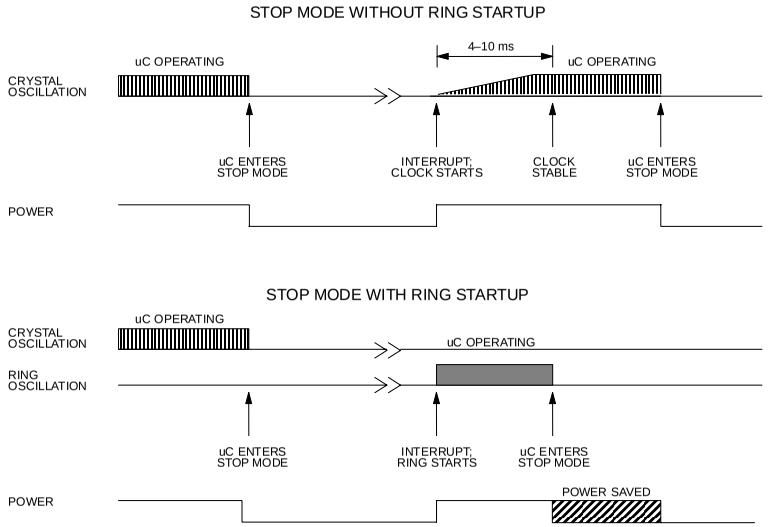


Diagram assumes that the operation following Stop requires less than 18 ms complete.

TIMED ACCESS PROTECTION

Selected SFR bits are critical to operation, making it desirable to protect against an accidental write operation. The Timed Access procedure prevents an errant cpu from accidentally altering a bit that would cause difficulty. The Timed Access procedure requires that the write of a protected bit be preceded by the following instructions :

```
MOV     0C7h, #0AAh
MOV     0C7h, #55h
```

By writing an AAh followed by a 55h to the Timed Access register (location C7h), the hardware opens a two cycle window that allows software to modify one of the protected bits. If the instruction that seeks to modify the protected bit is not immediately proceeded by these instructions, the write will not take effect. The protected bits are:

EXIF.0	BGS Band–gap Select
WDCON.6	POR Power–on Reset flag
WDCON.1	EWT Enable Watchdog
WDCON.0	RWT Reset Watchdog
WDCON.3	WDIF Watchdog Interrupt Flag

SPECIAL FUNCTION REGISTERS

Most special features of the DS80C320 or 80C32 are controlled by bits in special function registers (SFRs). This allows the DS80C320 to add many features but use the same instruction set. When writing software to use a new feature, the SFR must be defined to an assembler or compiler using an equate statement. This is the only change needed to access the new function. The DS80C320 duplicates the SFRs that are contained in the standard 80C32. Table 6 shows the register addresses and bit locations. Many are standard 80C32 registers. The High–Speed Microcontroller User's Guide describes all SFRs.

SPECIAL FUNCTION REGISTER LOCATIONS Table 6

REGISTER	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	ADDRESS
SP									81h
DPL									82h
DPH									83h
DPL1									84h
DPH1									85h
DPS	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**SEL**	86h
PCON	SMOD_0	**SMOD0**	–	–	GF1	GF0	STOP	IDLE	87h
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88h
TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	89h
TL0									8Ah
TL1									8Bh
TH0									8Ch
TH1									8Dh
CKCON	**WD1**	**WD0**	**T2M**	**T1M**	**T0M**	**MD2**	**MD1**	**MD0**	8Eh
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	90h
EXIF	**IE5**	**IE4**	**IE3**	**IE2**	**–**	**RGMD**	**RGSL**	**BGS**	91h
SCON0	SM0/FE_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	98h
SBUF0									99h
P2	P2.0	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	A0h
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	A8h
SADDR0									A9h
SADDR1									AAh
P3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	B0h
IP	–	PS1	PT2	PS0	PT1	PX1	PT0	PX0	B8h
SADEN0									B9h
SADEN1									BAh
SCON1	**SM0/FE_0**	**SM1_0**	**SM2_0**	**REN_0**	**TB8_0**	**RB8_0**	**TI_0**	**RI_0**	C0h
SBUF1									C1h
STATUS	PIP	HIP	LIP	1	1	1	1	1	C5h
TA									C7h
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/RL2	C8h
T2MOD	–	–	–	–	–	–	T2OE	DCEN	C9h
RCAP2L									CAh
RCAP2H									CBh
TL2									CCh
TH2									CDh
PSW	CY	AC	F0	RS1	RS0	OV	FL	P	D0h
WDCON	**SMOD_1**	**POR**	**EPFI**	**PFI**	**WDIF**	**WTRF**	**EWT**	**RWT**	D8h
ACC									E0h
EIE	**–**	**–**	**–**	**EWDI**	**EX5**	**EX4**	**EX3**	**EX2**	E8h
B									F0h
EIP	**–**	**–**	**–**	**PWDI**	**PX5**	**PX4**	**PX3**	**PX2**	F8h

PROGRAMMER'S GUIDE AND INSTRUCTION SET

Memory Organization

Program Memory

The 80C51 has separate address spaces for program and data memory. The Program memory can be up to 64k bytes long. The lower 4k can reside on-chip. Figure 1 shows a map of the 80C51 program memory.

The 80C51 can address up to 64k bytes of data memory to the chip. The MOVX instruction is used to access the external data memory.

The 80C51 has 128 bytes of on-chip RAM, plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri). Figure 2 shows the Data Memory organization.

Direct and Indirect Address Area

The 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into three segments as listed below and shown in Figure 3.

1. Register Banks 0-3: Locations 0 through 1FH (32 bytes). The device after reset defaults to register bank 0. To use the other register banks, the user must select them in software. Each register bank contains eight 1-byte registers 0 through 7. Reset initializes the stack pointer to location 07H, and it is incremented once to start from location 08H, which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location of the RAM where it is not used for data storage (i.e., the higher part of the RAM).
2. Bit Addressable Area: 16 bytes have been assigned for this segment, 20H-2FH. Each one of the 128 bits of this segment can be directly addressed (0-7FH). The bits can be referred to in two ways, both of which are acceptable by most assemblers. One way is to refer to their address (i.e., 0-7FH). The other way is with reference to bytes 20H to 2FH. Thus, bits 0-7 can also be referred to as bits 20.0-20.7, and bits 8-FH are the same as 21.0-21.7, and so on. Each of the 16 bytes in this segment can also be addressed as a byte.
3. Scratch Pad Area: 30H through 7FH are available to the user as data RAM. However, if the stack pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

Figure 2 shows the different segments of the on-chip RAM.

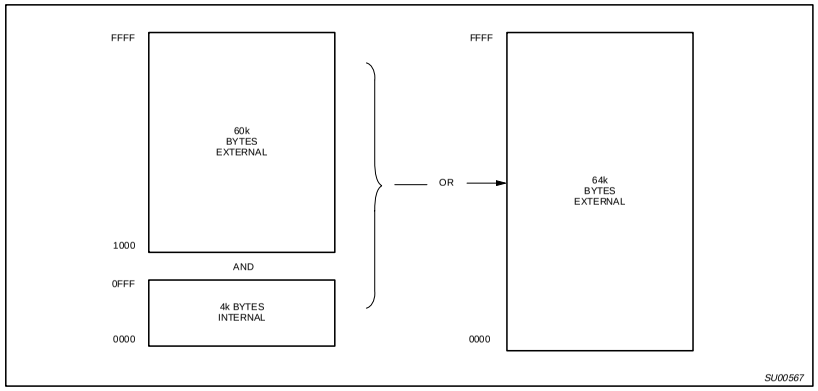


Figure 1. 80C51 Program Memory

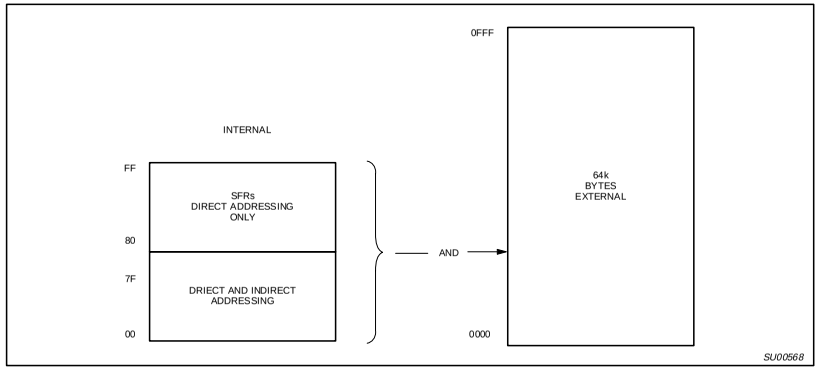


Figure 2. 80C51 Data Memory

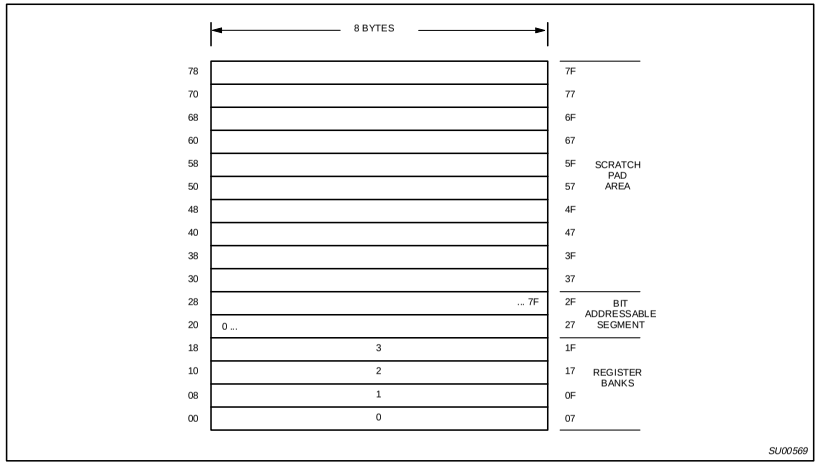


Figure 3. 128 Bytes of RAM Direct and Indirect Addressable

Table 1. 80C51 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt enable	A8H	EA	–	–	ES	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt priority	B8H	–	–	–	PS	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	–	–	–	–	–	–	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	Rxd	FFH
PCON[1]	Power control	87H	SMOD	–	–	–	GF1	GF0	PD	IDL	0xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	–	P	00H
SBUF	Serial data buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TMOD	Timer mode	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	00H

NOTES:
* Bit addressable
1. Bits GF1, GF0, PD, and IDL of the PCON register are not implemented on the NMOS 8051/8031.

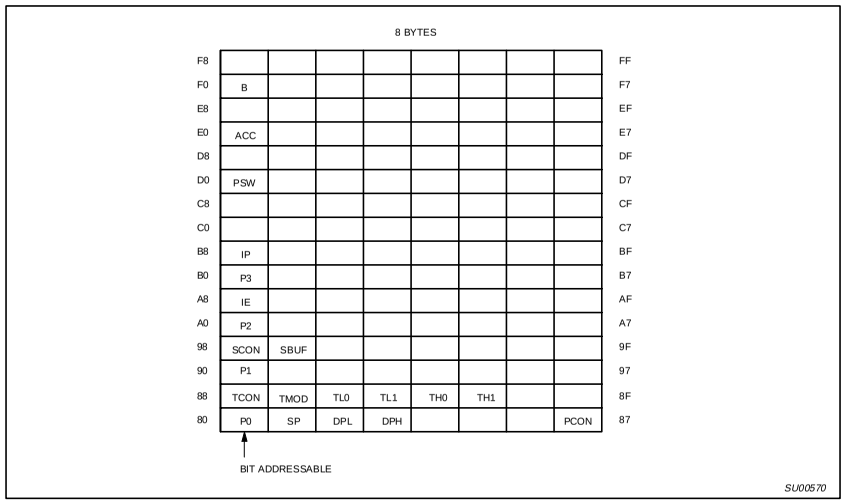


Figure 4. SFR Memory Map

Those SFRs that have their bits assigned for various functions are listed in this section. A brief description of each bit is provided for quick reference. For more detailed information refer to the Architecture Chapter of this book.

PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	–	P

CY	PSW.7	Carry Flag.
AC	PSW.6	Auxiliary Carry Flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1 (SEE NOTE 1).
RS0	PSW.3	Register Bank selector bit 0 (SEE NOTE 1).
OV	PSW.2	Overflow Flag.
–	PSW.1	Usable as a general purpose flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bus in the accumulator.

NOTE:
1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	–	–	–	GF1	GF0	PD	IDL

SMOD Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.

– Not implemented, reserved for future use.*

– Not implemented reserved for future use.*

– Not implemented reserved for future use.*

GF1 General purpose flag bit.

GF0 General purpose flag bit.

PD Power Down Bit. Setting this bit activates Power Down operation in the 80C51. (Available only in CMOS.)

IDL Idle mode bit. Setting this bit activates Idle Mode operation in the 80C51. (Available only in CMOS.)

If 1s are written to PD and IDL at the same time, PD takes precedence.

* User software should not write 1s to reserved bits. These bits may be used in future 8051 products to invoke new features.

Philips Semiconductors

80C51 Family

80C51 family programmer's guide and instruction set

INTERRUPTS:

To use any of the interrupts in the 80C51 Family, the following three steps must be taken.

1. **Set the EA (enable all) bit in the IE register to 1.**
2. **Set the corresponding individual interrupt enable bit in the IE register to 1.**
3. **Begin the interrupt service routine at the corresponding Vector Address of that interrupt. See Table below.**

INTERRUPT SOURCE	VECTOR ADDRESS
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H

In addition, for external interrupts, pins INT0 and INT1 (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.

ITx = 0 level activated

ITx = 1 transition activated

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	–	–	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Not implemented, reserved for future use.*
—	IE.5	Not implemented, reserved for future use.*
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

ASSIGNING HIGHER PRIORITY TO ONE OR MORE INTERRUPTS:

In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1.

Remember that while an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.

PRIORITY WITHIN LEVEL:

Priority within level is only to resolve simultaneous requests of the same priority level.

From high to low, interrupt sources are listed below:

IE0
TF0
IE1
TF1
RI or TI

IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

–	–	–	PS	PT1	PX1	PT0	PX0

–	IP.7	Not implemented, reserved for future use.*
–	IP.6	Not implemented, reserved for future use.*
–	IP.5	Not implemented, reserved for future use.*
PS	IP.4	Defines the Serial Port interrupt priority level.
PT1	IP.3	Defines the Timer 1 interrupt priority level.
PX1	IP.2	Defines External Interrupt 1 priority level.
PT0	IP.1	Defines the Timer 0 interrupt priority level.
PX0	IP.0	Defines the External Interrupt 0 priority level.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

Philips Semiconductors

80C51 Family — 80C51 family programmer's guide and instruction set

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
		Timer 1				**Timer 0**	

GATE	When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).
C/$\overline{T}$	Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
M1	Mode selector bit. (NOTE 1)
M0	Mode selector bit. (NOTE 1)

NOTE 1:

M1	M0		Operating Mode
0	0	0	13-bit Timer (8048 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standart Timer 0 control bits. TH0 is an8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

TIMER SET-UP

Tables 2 through 5 give some values for TMOD which can be used to set up Timer 0 in different modes.

It is assumed that only one timer is being used at a time. If it is desired to run Timers 0 and 1 simultaneously, in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if it is desired to run Timer 0 in mode 1 GATE (external control), and Timer 1 in mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 2 ORed with 60H from Table 5).

Moreover, it is assumed that the user, at this point, is not ready to turn the timers on and will do that at a different point in the program by setting bit TRx (in TCON) to 1.

TIMER/COUNTER 0

Table 2. As a Timer:

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	Two 8-bit Timers	03H	0BH

Table 3. As a Counter:

MODE	COUNTER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	One 8-bit Counter	07H	0FH

NOTES:
1. The timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on $\overline{INT0}$ (P3.2) when TR0 = 1 (hardware control).

Philips Semiconductors

80C51 Family

80C51 family programmer's guide and instruction set

TIMER/COUNTER 1

Table 4. As a Timer:

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	Does not run	30H	B0H

Table 5. As a Counter:

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	Not available	–	–

NOTES:
1. The timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on $\overline{INT1}$ (P3.2) when TR1 = 1 (hardware control).

SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier. (NOTE 1)
SM1	SCON.6	Serial Port mode specifier. (NOTE 1)
SM2	SCON.5	Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 6.)
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	$F_{OSC.}/12$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$F_{OSC.}/64$ or $F_{OSC.}/32$
1	1	3	9-bit UART	Variable

SERIAL PORT SET-UP:

Table 6.

MODE	SCON	SM2 VARIATION
0 1 2 3	10H 50H 90H D0H	Single Processor Environment (SM2 = 0)
0 1 2 3	NA 70H B0H F0H	Multiprocessor Environment (SM2 = 1)

GENERATING BAUD RATES

Serial Port in Mode 0:

Mode 0 has a fixed baud rate which is 1/12 of the oscillator frequency. To run the serial port in this mode none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

$$\text{Baud Rate} \quad \frac{\text{Osc Freq}}{12}$$

Serial Port in Mode 1:

Mode 1 has a variable baud rate. The baud rate is generated by Timer 1.

USING TIMER/COUNTER 1 TO GENERATE BAUD RATES:

For this purpose, Timer 1 is used in mode 2 (Auto-Reload). Refer to Timer Setup section of this chapter.

$$\text{Baud Rate} \quad \frac{K \quad \text{Osc Freq}}{32 \quad 12 \quad [256 \quad (TH1)]}$$

If SMOD = 0, then K = 1.
If SMOD = 1, then K = 2 (SMOD is in the PCON register).

Most of the time the user knows the baud rate and needs to know the reload value for TH1.

$$TH1 \quad 256 \quad \frac{K \quad \text{Osc Freq}}{384 \quad \text{baud rate}}$$

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency.

Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register (i.e., ORL PCON,#80H). The address of PCON is 87H.

SERIAL PORT IN MODE 2:

The baud rate is fixed in this mode and is 1/32 or 1/64 of the oscillator frequency, depending on the value of the SMOD bit in the PCON register.

In this mode none of the Timers are used and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = 1/32 Osc Freq.

SMOD = 0, Baud Rate = 1/64 Osc Freq.

To set the SMOD bit: ORL PCON,#80H. The address of PCON is 87H.

SERIAL PORT IN MODE 3:

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.

80C51 FAMILY INSTRUCTION SET

Table 7. 80C51 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings[1]

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:

Rn	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@Ri	8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
#data	8-bit constant included in the instruction.
#data 16	16-bit constant included in the instruction
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
bit	Direct Addressed bit in Internal Data RAM or Special Function Register.

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with carry	1	12
ADDC	A,#data	Add immediate data to A_{CC} with carry	2	12
SUBB	A,Rn	Subtract Register from A_{CC} with borrow	1	12
SUBB	A,direct	Subtract direct byte from A_{CC} with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from A_{CC} with borrow	1	12
SUBB	A,#data	Subtract immediate data from A_{CC} with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS (Continued)				
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment indirect RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A and B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator left	1	12
RLC	A	Rotate Accumulator left through the carry	1	12
RR	A	Rotate Accumulator right	1	12
RRC	A	Rotate Accumulator right through the carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
DATA TRANSFER (Continued)				
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	RN,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A_{CC}	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to A_{CC}	1	24
MOVX	A,@Ri	Move external RAM (8-bit addr) to A_{CC}	1	24
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A_{CC}	1	24
MOVX	A,@Ri,A	Move A_{CC} to external RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move A_{CC} to external RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order digit indirect RAM with A_{CC}	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to carry	2	24
ANL	C,/bit	AND complement of direct bit to carry	2	24
ORL	C,bit	OR direct bit to carry	2	24
ORL	C,/bit	OR complement of direct bit to carry	2	24
MOV	C,bit	Move direct bit to carry	2	12
MOV	bit,C	Move carry to direct bit	2	24
JC	rel	Jump if carry is set	2	24
JNC	rel	Jump if carry not set	2	24

Philips Semiconductors

80C51 Family

80C51 family programmer's guide and instruction set

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
BOOLEAN VARIABLE MANIPULATION (Continued)				
JB	rel	Jump if direct bit is set	3	24
JNB	rel	Jump if direct bit is not set	3	24
JBC	bit,rel	Jump if direct bit is set and clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute subroutine call	2	24
LCALL	addr16	Long subroutine call	3	24
RET		Return from subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute jump	2	24
LJMP	addr16	Long jump	3	24
SJMP	rel	Short jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is zero	2	24
JNZ	rel	Jump if Accumulator is not zero	2	24
CJNE	A,direct,rel	Compare direct byte to A_{CC} and jump if not equal	3	24
CJNE	A,#data,rel	Compare immediate to A_{CC} and jump if not equal	3	24
CJNE	RN,#data,rel	Compare immediate to register and jump if not equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	24
DJNZ	Rn,rel	Decrement register and jump if not zero	2	24
DJNZ	direct,rel	Decrement direct byte and jump if not zero	3	24
NOP		No operation	1	12

APPENDIX C: ELECTRIC DIAGRAMS

In this appendix are available some electric diagrams of the most frequently used **GPC® 324** interfaces. All these interface can be yourself produced and some of them are standard **grifo®** cards and, if required, they can be directly ordered.

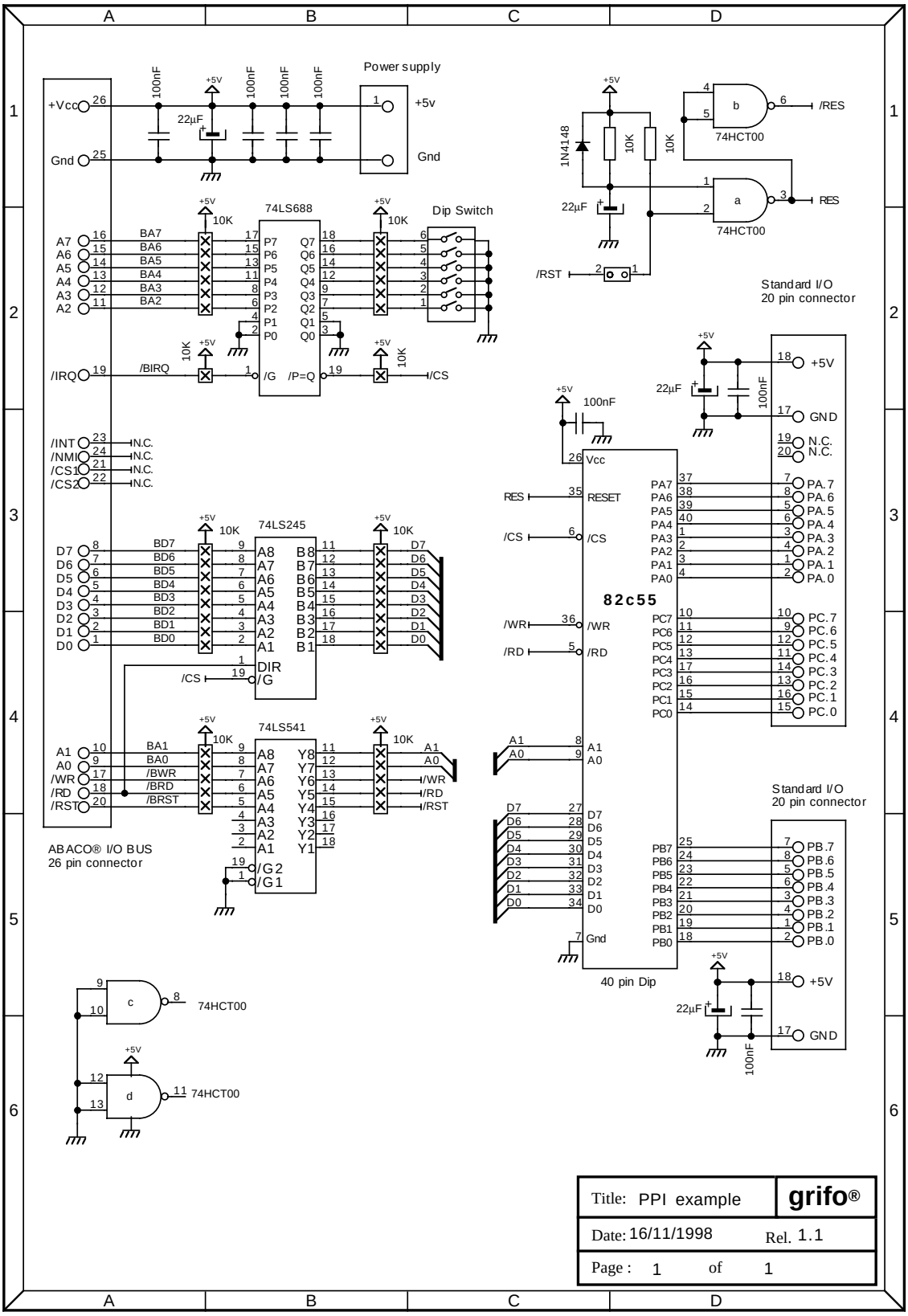


FIGURE C1: PPI EXPANSION ELECTRIC DIAGRAM

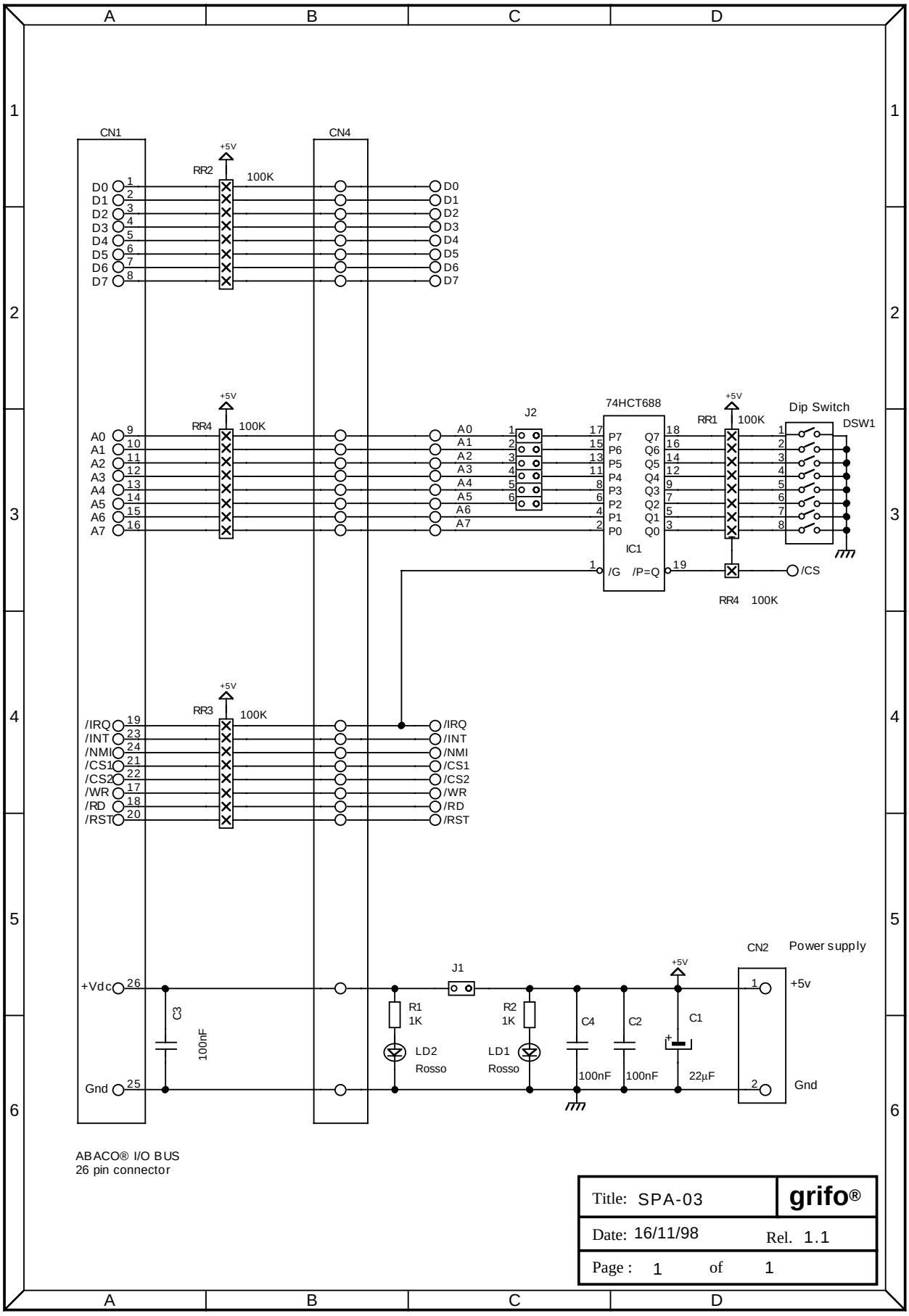


FIGURE C2: SPA 03 ELECTRIC DIAGRAM

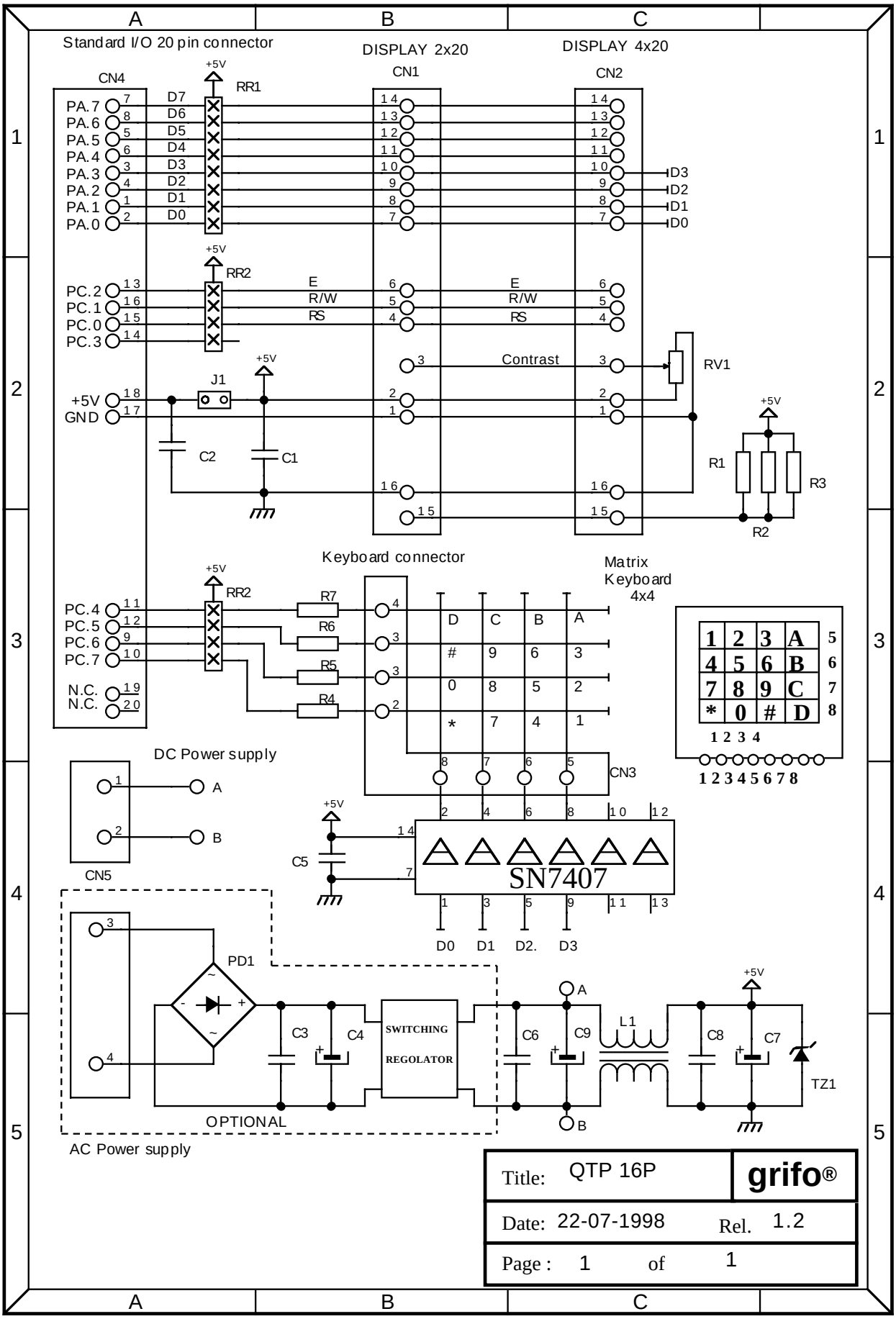


FIGURE C3: QTP 16P ELECTRIC DIAGRAM

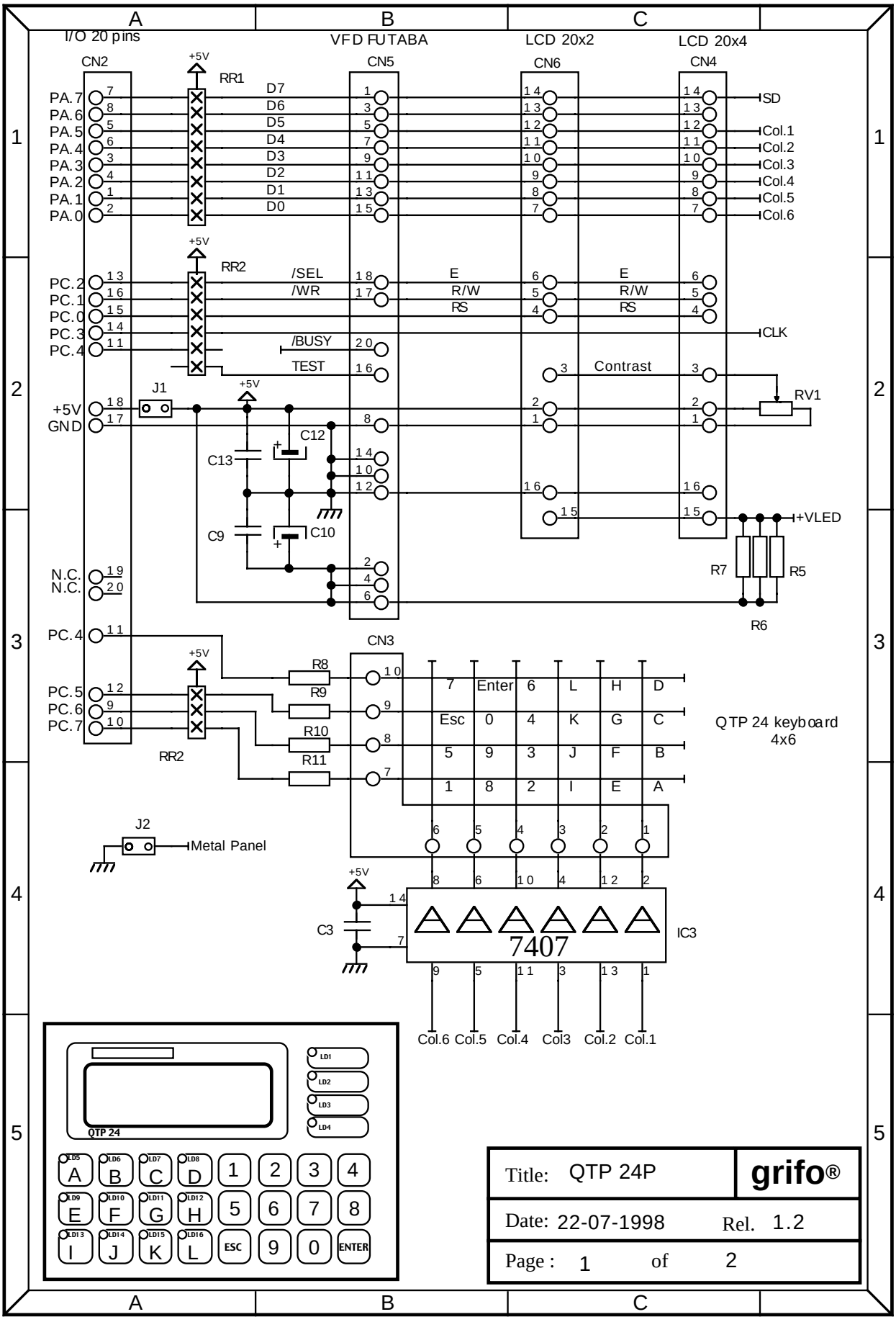


FIGURE C4: QTP 24P ELECTRIC DIAGRAM (1 OF 2)

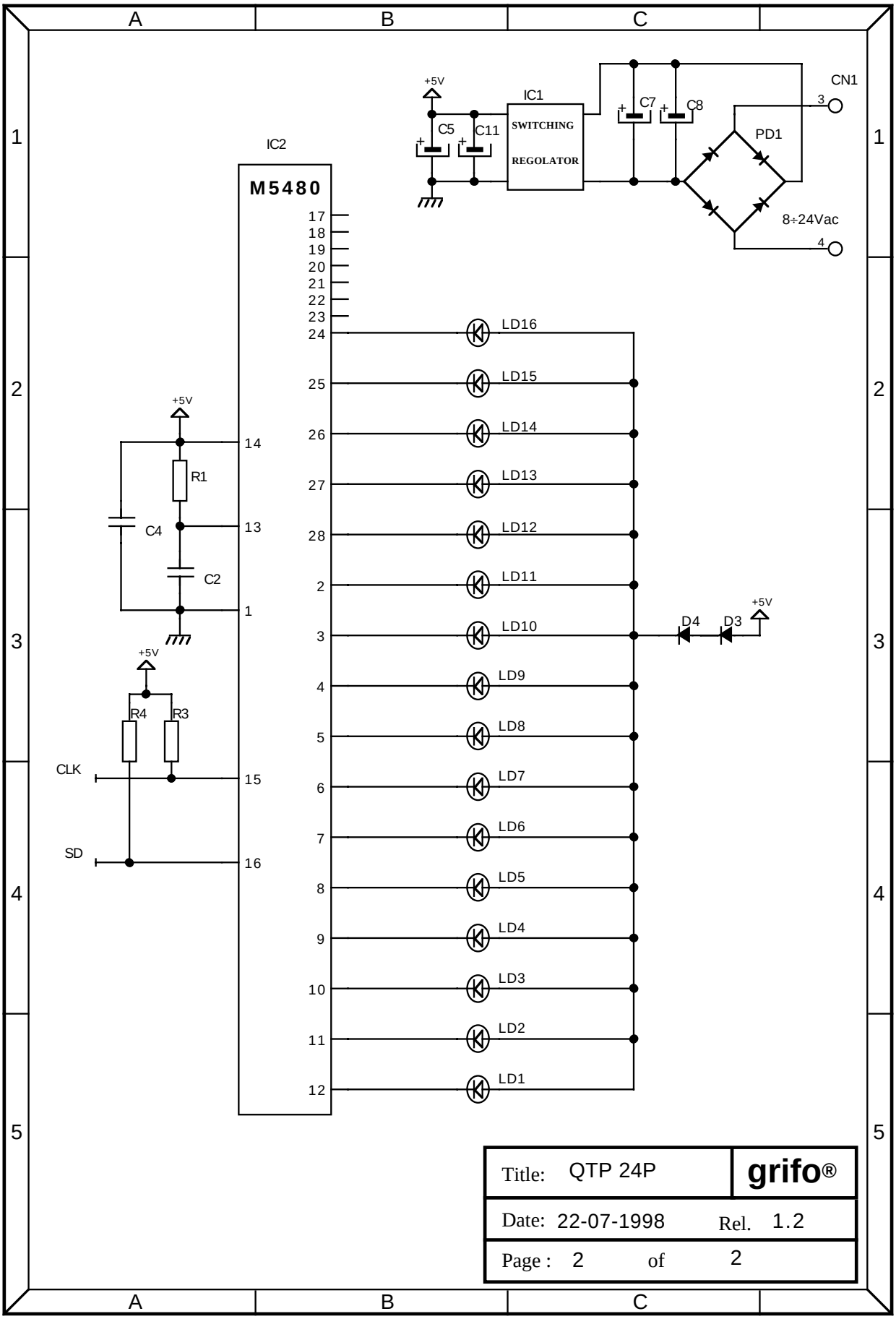


FIGURE C5: QTP 24P ELECTRIC DIAGRAM (2 OF 2)

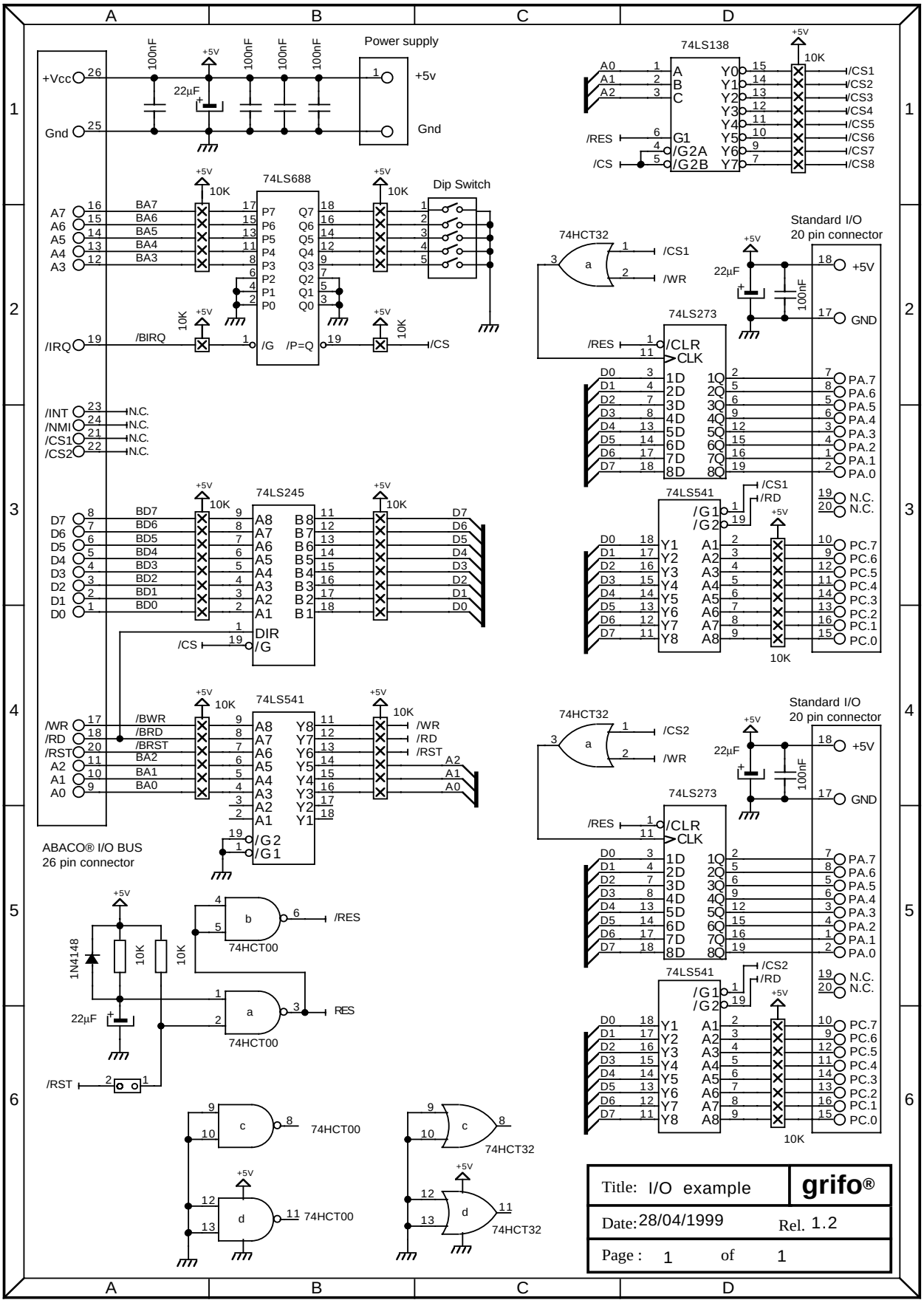


FIGURE C6: ABACO® I/O BUS INPUT OUTPUT ELECTRIC DIAGRAM

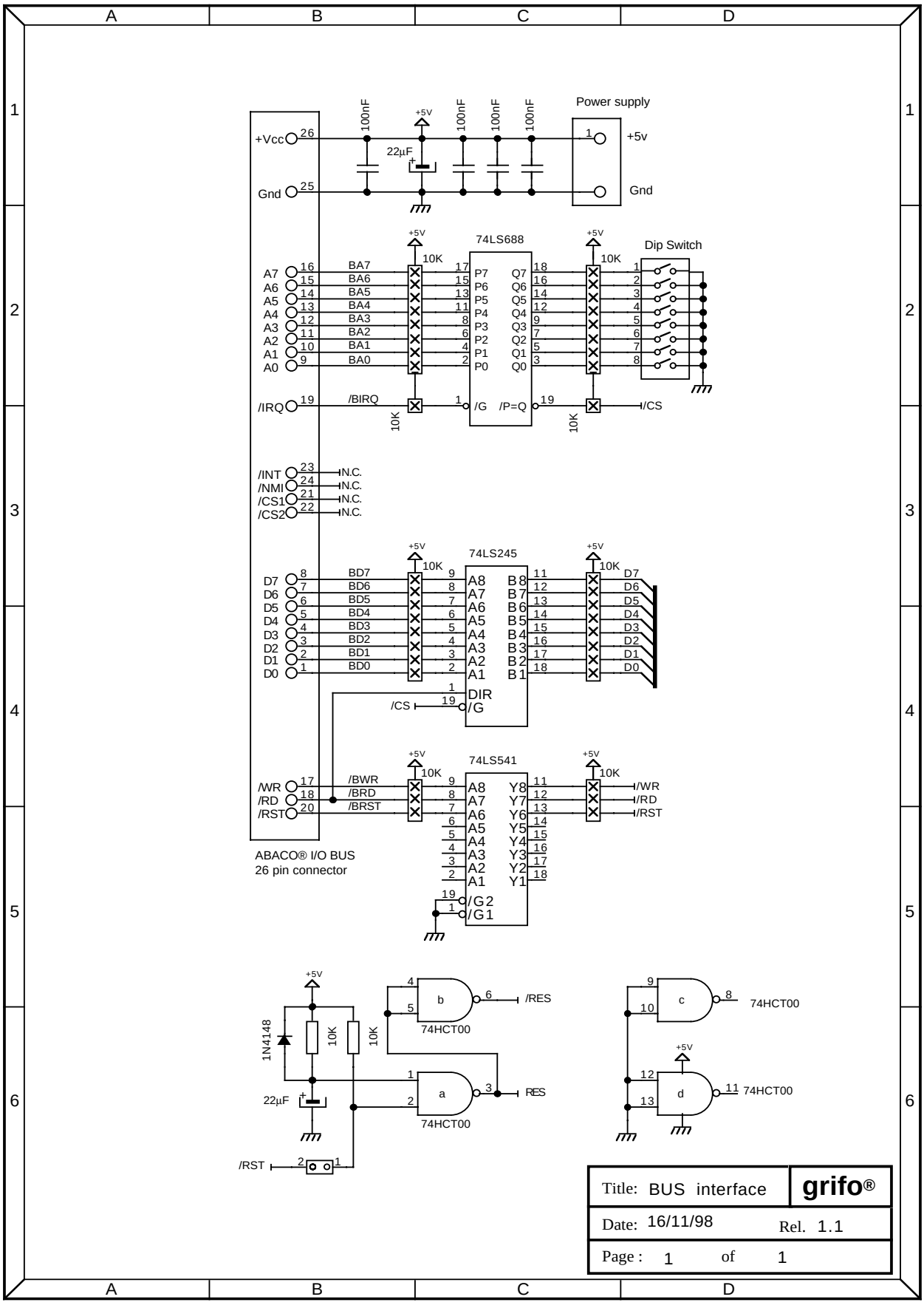


FIGURE C7: BUS INTERFACE ELECTRIC DIAGRAM

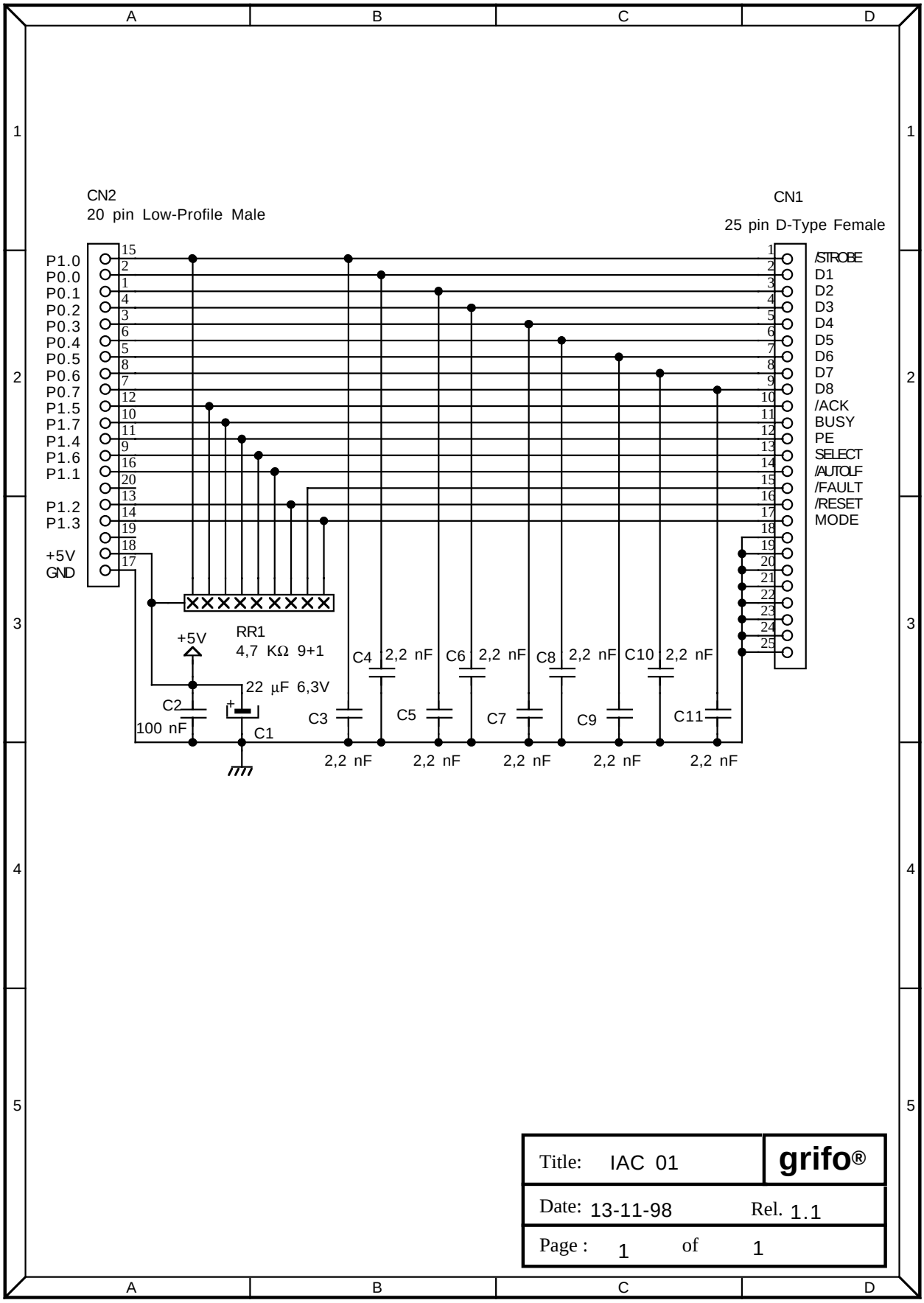


FIGURE C8: IAC 01 ELECTRIC DIAGRAM

APPENDIX D: ALPHABETICAL INDEX

M

N

O

P

R

S