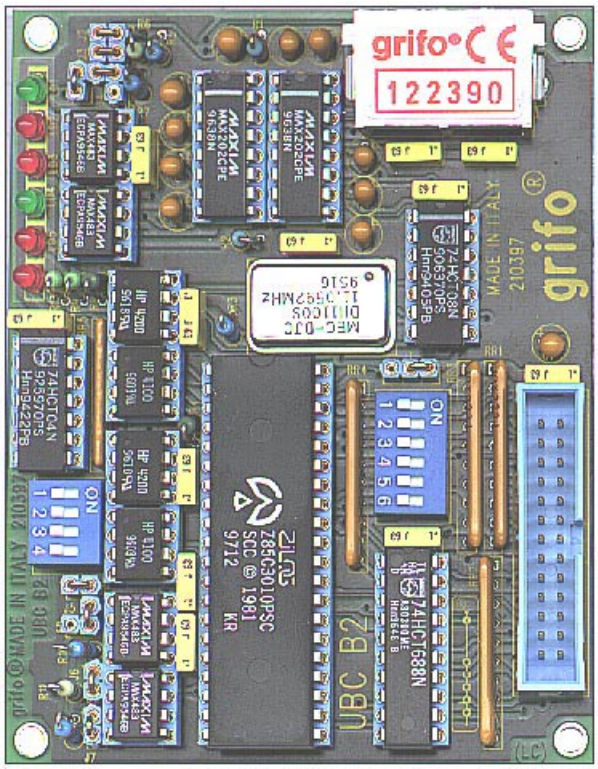
UBC B2

UART Block Communication Card
Abaco® I/O BUS 2 serial lines

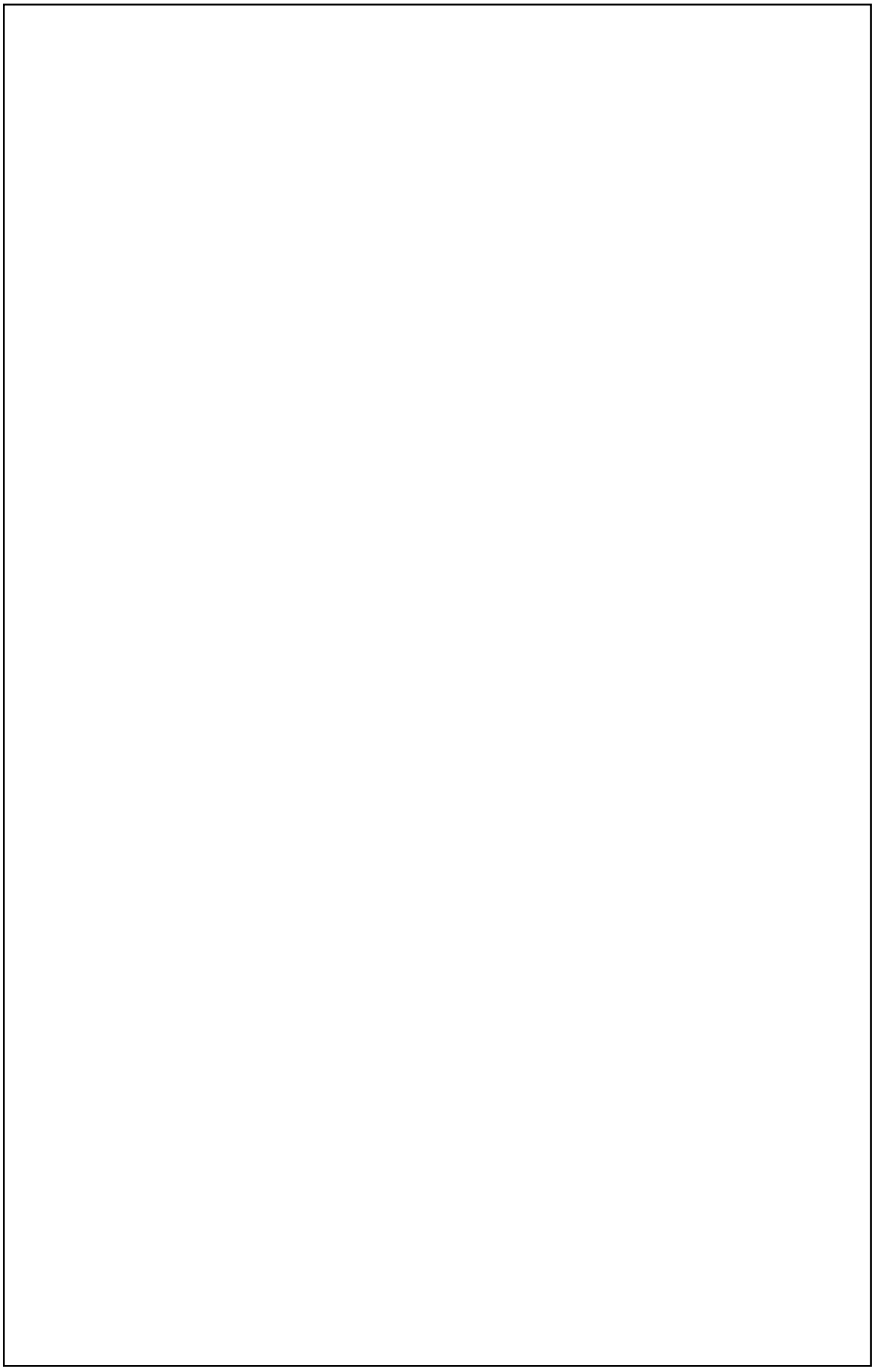
TECHNICAL MANUAL



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

UBC B2 Edition 5.10 *Rel. 14 October 2002*



UBC B2

UART Block Communication Card
Abaco® I/O BUS 2 serial lines

TECHNICAL MANUAL

Peripheral module for **industrial ABACO® I/O BUS**; half EUROPE format 100x80mm; **SCC Z85c30** with **11.0592 MHz** oscillator, which is capable to manage **HDLC**, **SDLC**, etc. protocols; protocol and **Baud rate**, up to **115 KBaud**, are software settable; 4 pins **Dip Switch** software readable; **2** serial lines bufferable as **RS 232**, **RS 422**, **RS 485** or **Current Loop**; **6** LEDs, installed on the front, to visualizethe status of TX, RX and RTS signals; possibility to generate an **interrupt** signal to connetct to /INT or /NMI BUS signals; **ABACO® I/O BUS** addressing space as low as 4 bytes only; 6 pins **Dip Switch** for I/O address setting; unique **5Vdc**power supply, **45mA** current consumption (basic configuration).

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

UBC B2 Edition 5.10 *Rel. 14 October 2002*

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

FIGURES 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 **UBC B2** card release **111002** and later. The validity of the bring informations is subordinate to the number of the card release. The user must always verify the correct correspondence among the two denotations. On the card the release number is present in more points both board printed diagram (serigraph) and printed circuit (for example between UART and drivers on the component side).

BOARD FEATURES

UBC B2 (**U**art **B**lock Communication **C**ard Bus **ABACO® I/O BUS 2** lines) board is a powerful module designed for industrial **ABACO®** I/O BUS serial communications featuring remarkable possibilities of interfacement to several external devices.
The board is based on the flexible Zilog SCC 85c30, which can support asynchronous communication and even, without external devices, well developed protocols like HDLC and SDLC.
The **UBC B2** board comes in half Europe format 100x80mm, it manages separetely 2 serial lines whose protocols and communication speeds are software settable; each of the 2 lines manage in autonomy the 4 canonical signals: TxD, RxD, CTS, RTS. In addition, each communication line can be buffered as RS 232, RS 422, RS 485 or CURRENT LOOP. At last, 3 of these signals are monitored separately for the 2 serial lines through LEDs.
The asynchronous communication can be made employing one of the following communication protocols:

- Baud rate selectable amongst 150 and 115.2 KBaud
- Stop bit selectable amongst 1; 1.5 and 2 bits
- Word length selectable from 5 to 8 bits
- Parity can be even, odd or none

A software readable 4-pins Dip Switch allows the User to set on the board particular conditions to inform the firmware about a particular situation (for example; a certain Baud rate and communication protocol have been chosen).
UBC B2 board takes 4 bytes of addressing space where all the registers that allow software management and programming of the board are allocated.
These addresses can be allocated in the **industrial ABACO® I/O BUS** addressing space through a 6 pins Dip Switch.
The board is capable to generate an interrupt signal when specific software settable events occour.

- Peripheral module for **industrial ABACO® I/O BUS**.
- Half EUROPE format 100x80mm.
- **SCC Z85c30** with **11.0592 MHz** oscillator, which is capable to manage **HDLC**, **SDLC**, etc. protocols; protocol and **Baud rate**, up to **115 KBaud**, are software settable.
- 4 pins **Dip Switch** software readable.
- **2** serial lines bufferable as **RS 232**, **RS 422**, **RS 485** or **Current Loop**.
- **6** LEDs, installed on the front, to visualize the status of TX, RX and RTS signals.
- Possibility to generate an **interrupt** signal to connetct to /INT or /NMI BUS signals.
- **ABACO® industrial BUS** addressing space as low as 4 bytes only.
- 6 pins **Dip Switch** for I/O address setting.
- Unique **5Vdc**power supply, **45mA** current consumption (basic configuration).

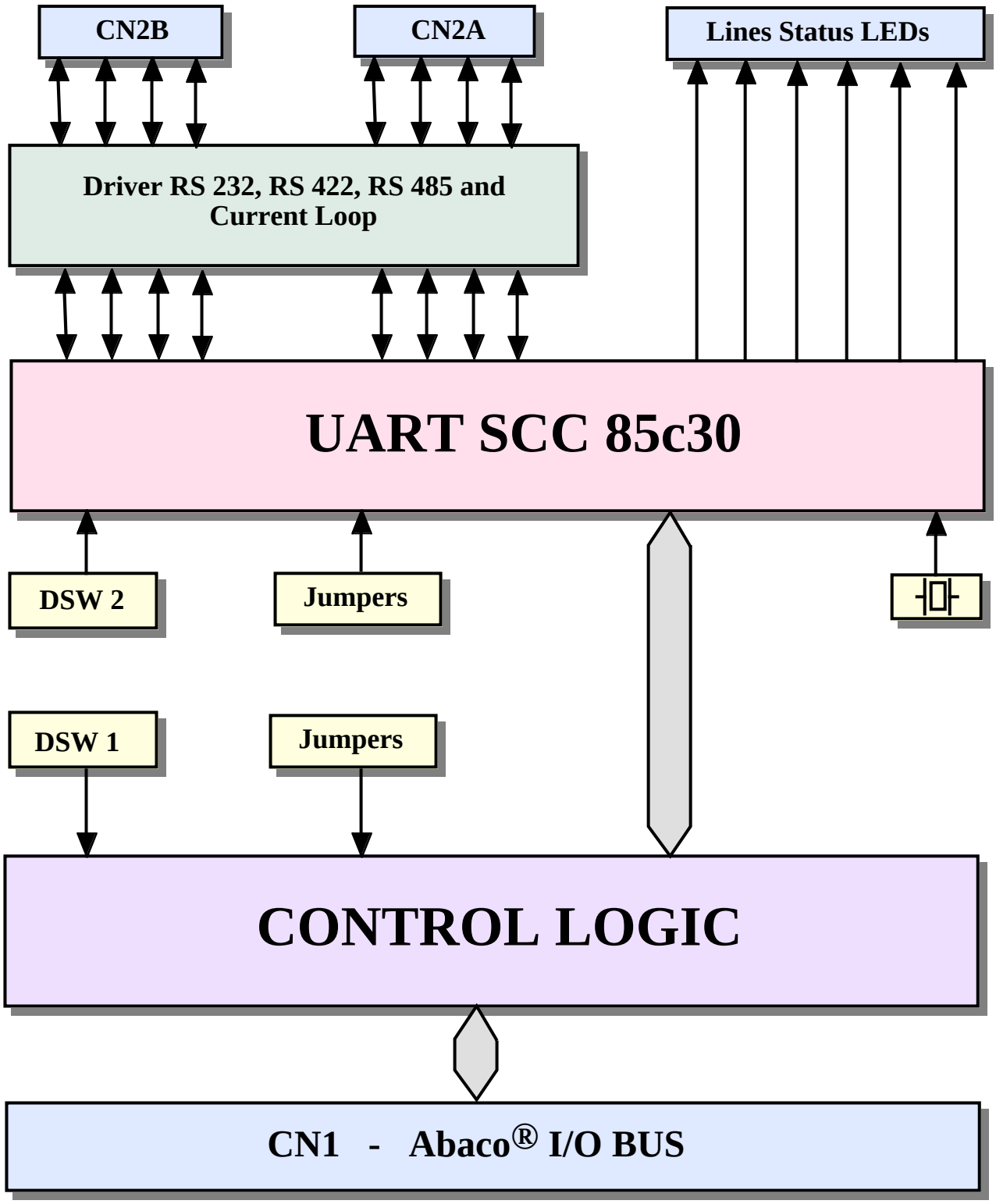


FIGURE 1: BLOCK DIAGRAM

TECHNICAL FEATURES

GENERAL FEATURES

Board Resources:
- Interface for **indutrial ABACO® I/O BUS.**
- **4 pins Dip switch** software readable.
- **6 pins Dip switch** to set I/O address.
- 2 Full Duplex serial lines in RS 232, RS 422, RS 485 or Current-Loop.

UART: Zilog SCC 85c30

Crystal (clock) frequency: 11.0592 MHz

BUS Interface: 8 bits wide BUS for data and address.
256 bytes total addressing space.
4 bytes of I/O space occupied.

PHYSICAL FEATURES

Size (W x H x D): Half Eurocard standard format: 100 x 80 x 17 mm

Weight: 85 g (basic version)

Connectors:
CN1: low profile 26 pins male
CN2A: 6 pins, Plug, female, 90 degreeses
CN2B: 6 pins, Plug, female, 90 degreeses

Temperature range: from 0 to 70 Centigrad degreeses

Relative humidity: 20% up to 90% (without condens)

ELECTRIC FEATURES

Power Supply:	+5 Vdc	
Consumption on 5 Vdc:	45 mA (default RS 232 configuration) 90 mA (maximum configuration))	
RS 422, RS 485 line termination:	Line termination resistance= Positive pull up resistance= Negative pull up resistance=	120 Ω 3.3 KΩ 3.3 KΩ

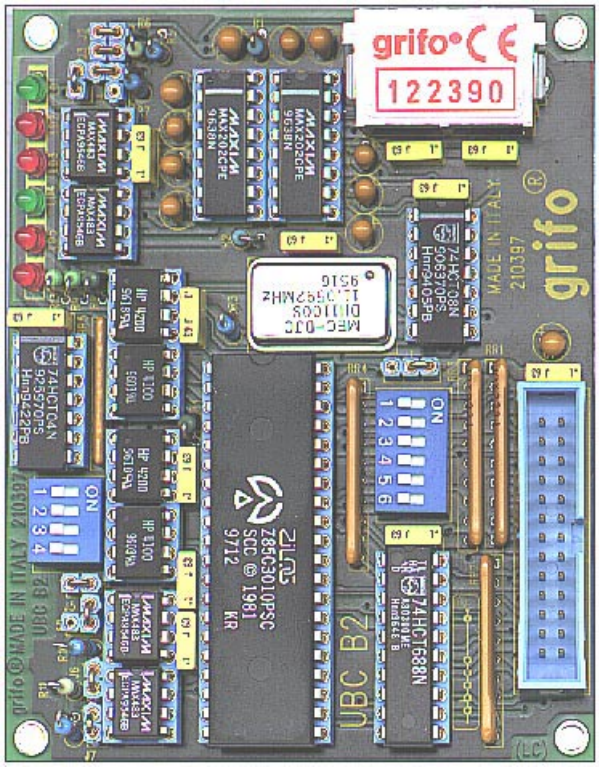


FIGURE 2: CARD PHOTO

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.

VISUAL SIGNALATIONS

UBC B2 board is provided with six LEDs in order to signal to the User some internal status conditions, as described in the following table:

LED	COLOUR	PURPOSE
LD1	Red	Indicates data traffic on signal TxB of serial line B.
LD2	Red	Indicates data traffic on signal RxB of serial line B.
LD3	Green	Indicates data traffic on signal TxA of serial line A.
LD4	Green	Indicates data traffic on signal RxA of serial line A.
LD5	Red	Indicates the status of signal RTSB of serial line B; when the LED is lit the signal is at logic level 0 and viceversa.
LD6	Green	Indicates the status of signal RTSA of serial line A; when the LED is lit the signal is at logic level 0 and viceversa.

FIGURE 3: VISUAL SIGNALATIONS TABLE

The main purpose of this LED is to provide the user a visual indication of the board status, making easier the operations to verify the correct workig of the system. To easily locate the LEDs on the board please see figure 4.

CONNECTIONS

The **UBC B2** module has 3 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) and connectors location (see figure 5). For further information about serial connections, please refer to the successive figures, which show the kind of board connection to perform.

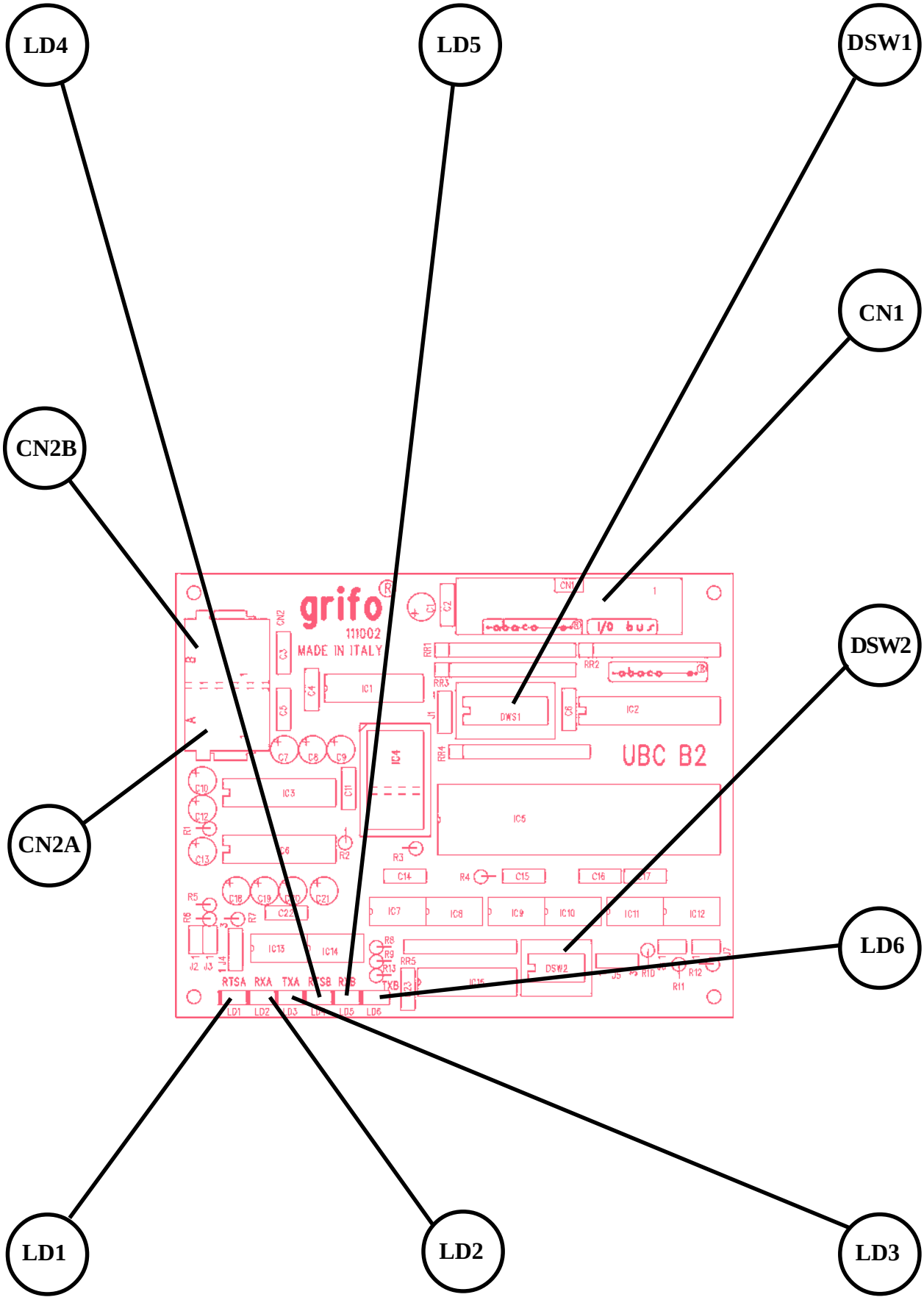


FIGURE 4: CONNETTORS, LEDS, DIP SWICTH, ETC. LOCATION

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 via **ABACO®** I/O BUS 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, etc. All this connector signals are at TTL level.

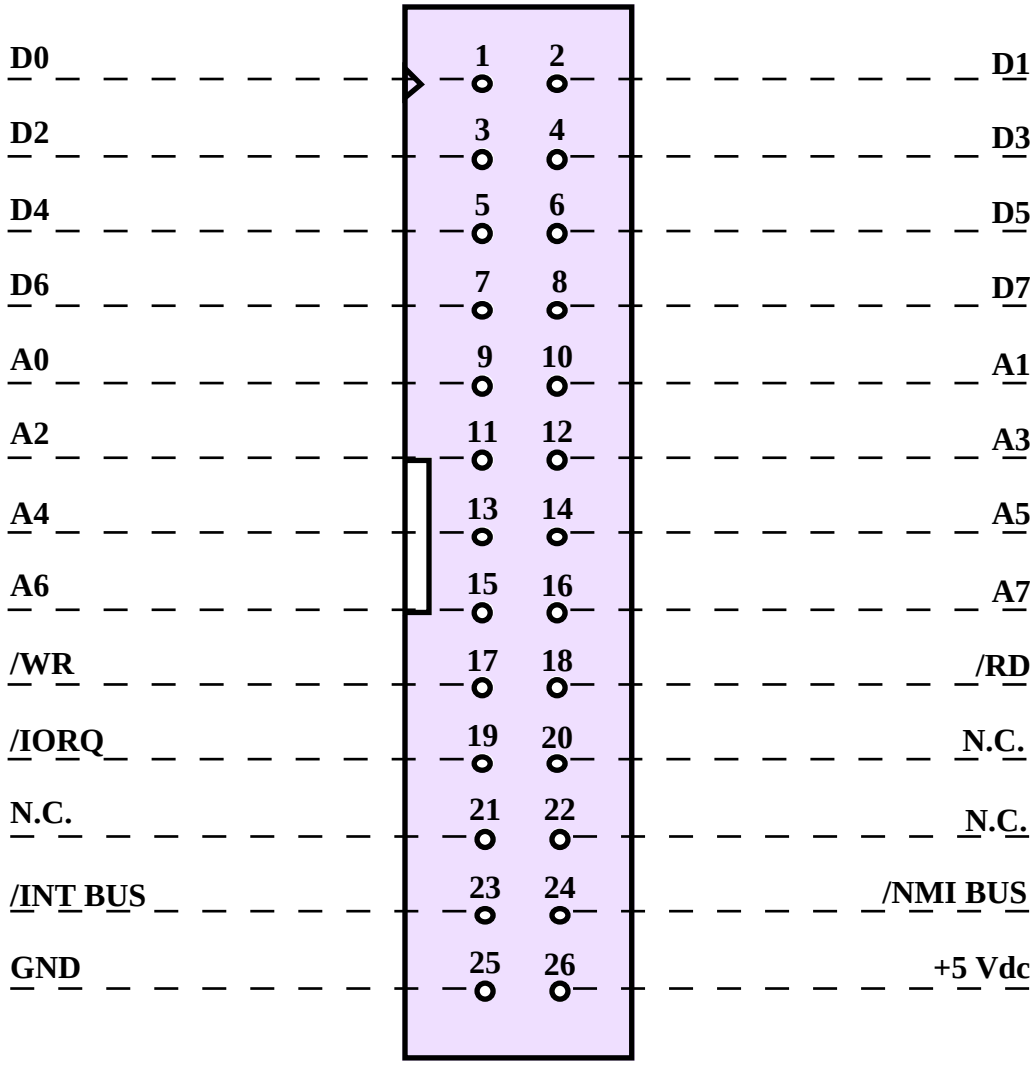


FIGURE 5: CN1 - ABACO® I/O BUS CONNECTOR

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.

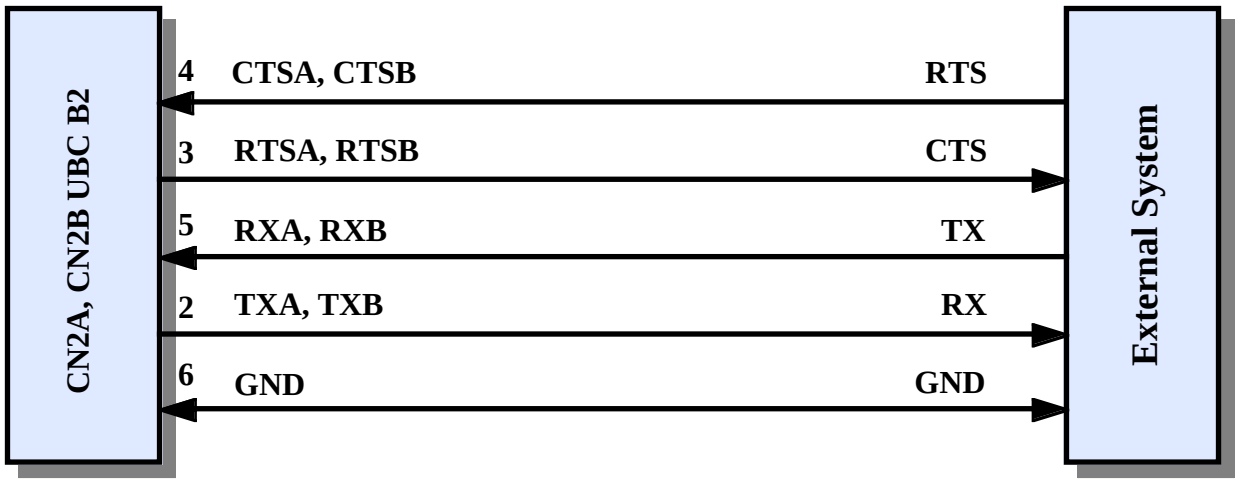


FIGURE 6: RS 232 CONNECTION EXAMPLE

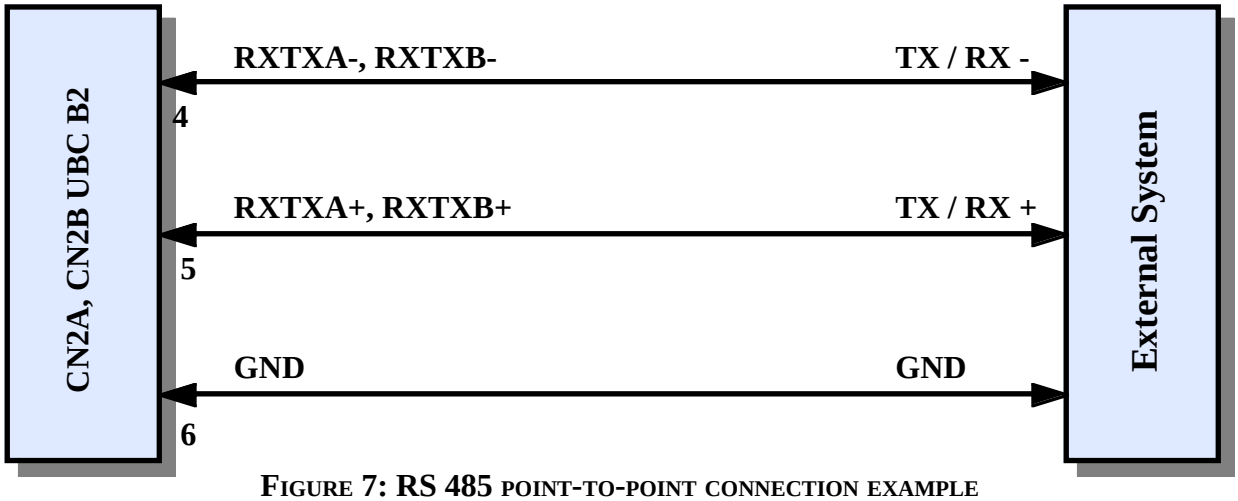


FIGURE 7: RS 485 POINT-TO-POINT CONNECTION EXAMPLE

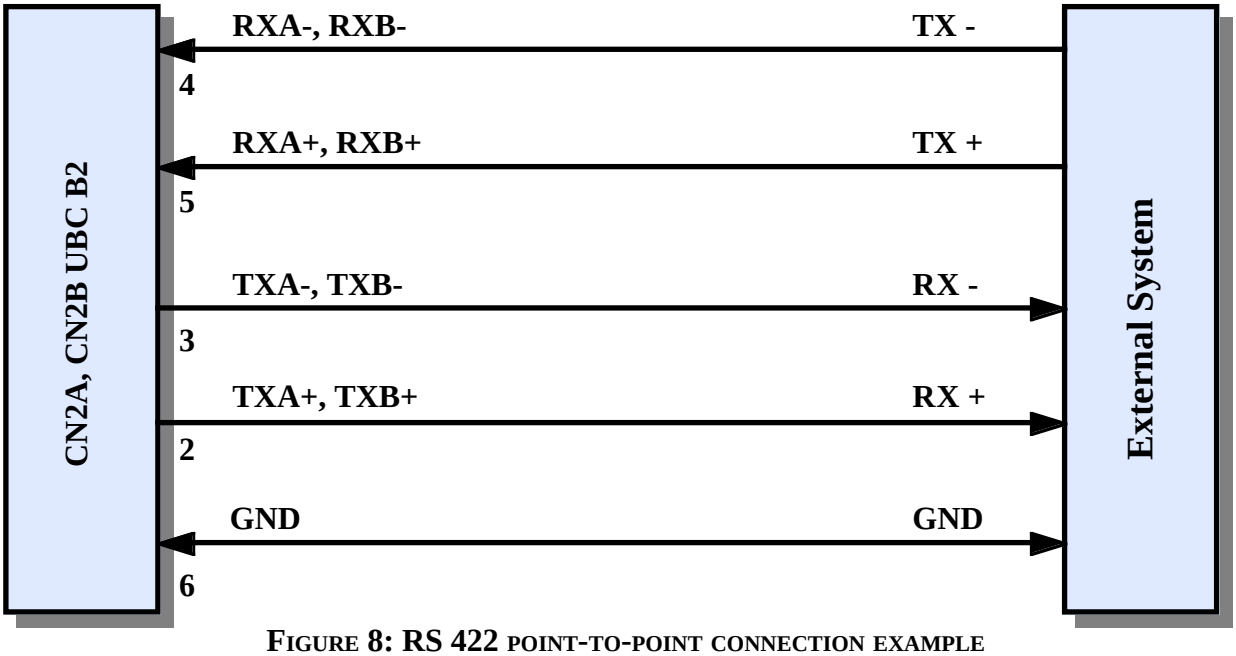


FIGURE 8: RS 422 POINT-TO-POINT CONNECTION EXAMPLE

CN2A - SERIAL LINE A CONNECTOR

CN2A is a 6 pins, female PLUG connector for serial line A, that can be buffered as RS 232, RS 422, RS 485 or Current Loop. Phisically, serial line A of **UBC B2** board is connected to serial line A. Placing of the signal has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the electric protocol follows the CCITT normative.

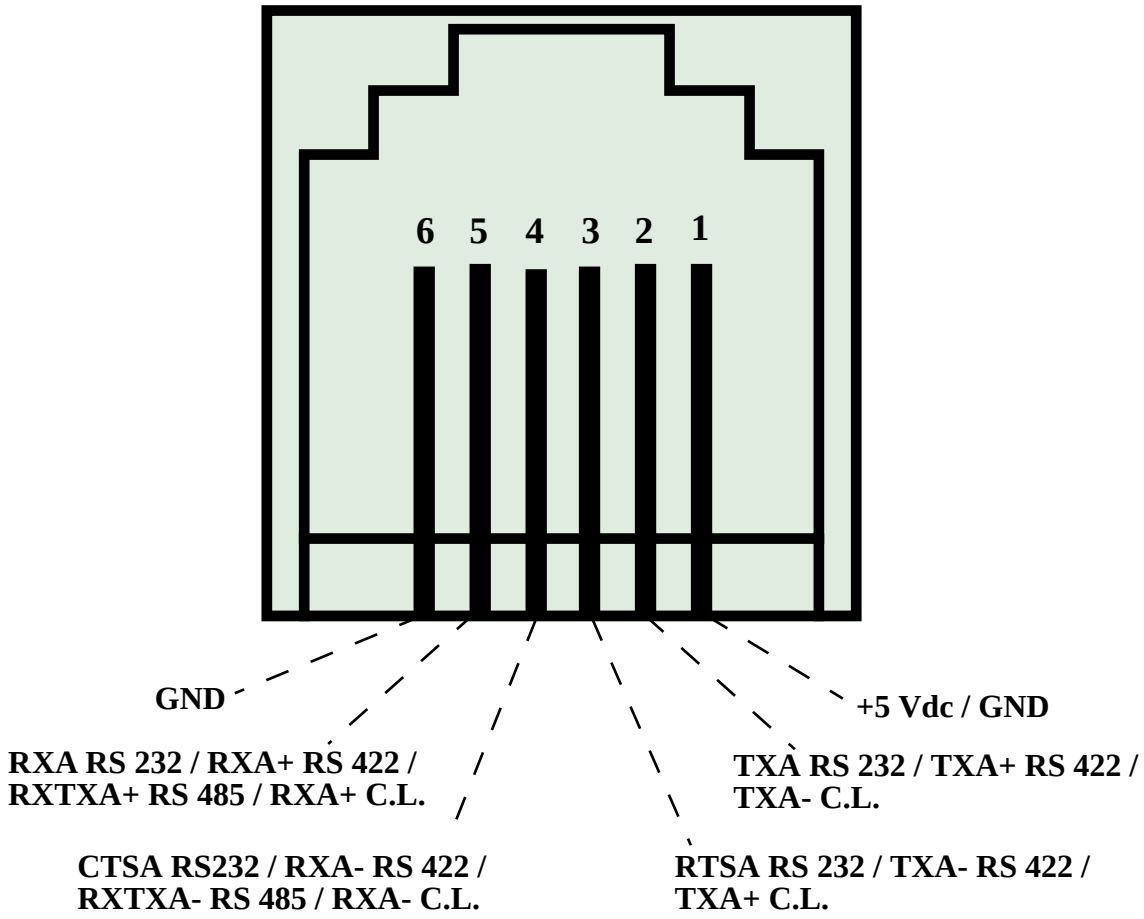


FIGURE 9: CN2A - SERIAL LINE A CONNECTOR

Signals description:

RXA RS 232	= I -	Serial line A RS 232 Receive Data.
TXA RS 232	= O -	Serial line A RS 232 Transmit Data.
CTSA RS 232	= I -	Serial line A RS 232 Clear To Send.
RTSA RS 232	= O -	Serial line A RS 232 Request To Send.
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.

RXTXA+ RS 485	=I/O-	Receive Transmit Data Positive: Serial line A positive signal for RS 485 serial differential receive and transmit.
RXA- C.L.	= I -	Receive Data Negative: Serial line A negative signal for Current Loop serial bipolar receive.
RXA+ C.L.	= I -	Receive Data Positive: Serial line A positive signal for Current Loop serial bipolar receive.
TXA- C.L.	= O -	Transmit Data Negative: Serial line A negative signal for Current Loop serial bipolar transmit.
TXA+ C.L.	= O -	Transmit Data Positive: Serial line A positive signal for Current Loop serial bipolar transmit.
+5 Vdc/GND	= I -	+5 Vdc or ground signal.
GND	= -	Ground signal.

NOTE
Handshake signal CTSA must be connected to another serial system, if it is going to be software managed. This is essential to acquire its status without ambiguity.

CN2B - SERIAL LINE B CONNECTOR

CN2B is a 6 pins, female PLUG connector for serial line B, that can be buffered as RS 232, RS 422, RS 485 or Current Loop. Phisically, serial line B of **UBC B2** board is connected to serial line B. Placing of the signal has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the electric protocol follows the CCITT normative.

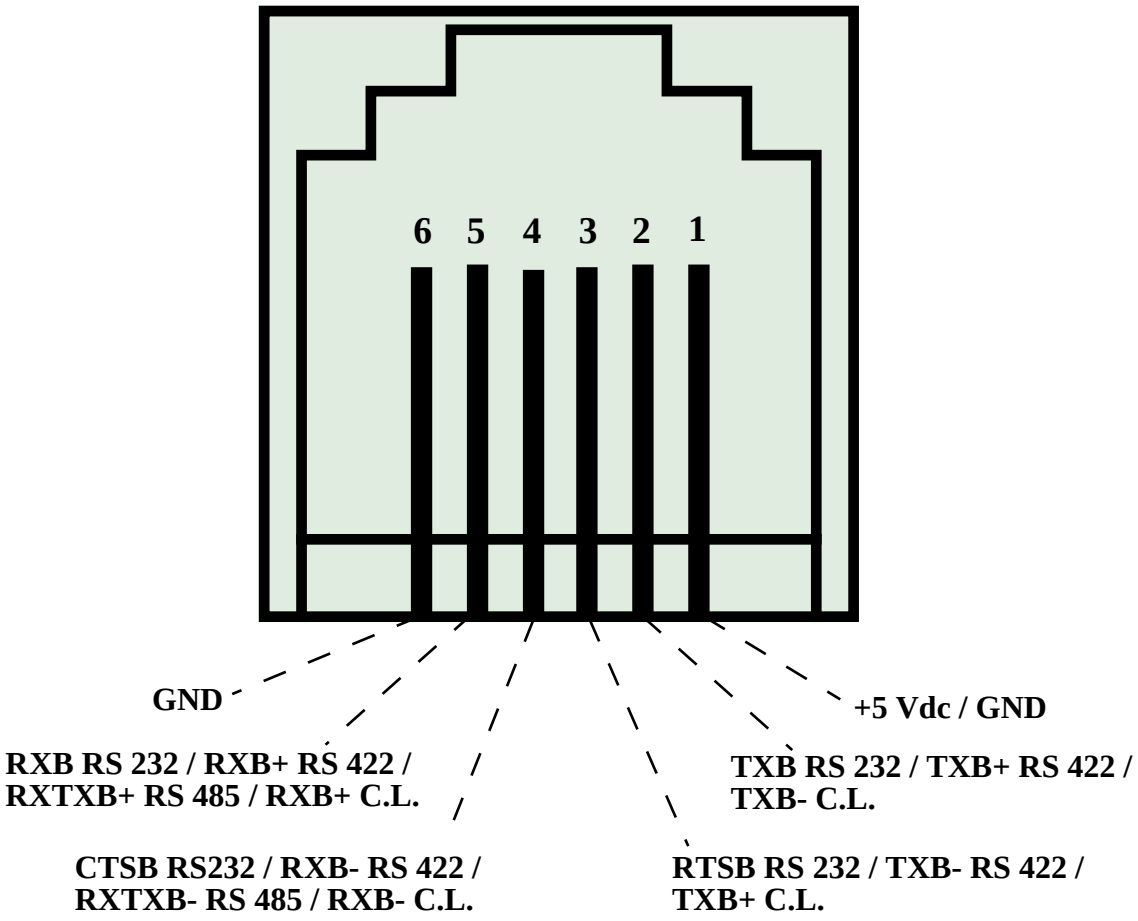


FIGURE 10: CN2B - SERIAL LINE B CONNECTOR

Signals description:

RXB RS 232	= I -	Serial line B RS 232 Receive Data.
TXB RS 232	= O -	Serial line B RS 232 Transmit Data.
CTSB RS 232	= I -	Serial line B RS 232 Clear To Send.
RTSB RS 232	= O -	Serial line B RS 232 Request To Send.
RXB- RS 422	= I -	Receive Data Negative: Serial line B negative signal for RS 422 serial differential receive.
RXB+ RS 422	= I -	Receive Data Positive: Serial line B positive signal for RS 422 serial differential receive.
TXB- RS 422	= O -	Transmit Data Negative: Serial line B negative signal for RS 422 serial differential transmit.
TXB+ RS 422	= O -	Transmit Data Positive: Serial line B positive signal for RS 422 serial differential transmit.
RXTXB- RS 485	=I/O-	Receive Transmit Data Negative: Serial line B negative signal for RS 485 serial differential receive and transmit.

RXTXB+ RS 485	=I/O-	Receive Transmit Data Positive: Serial line B positive signal for RS 485 serial differential receive and transmit.
RXB- C.L.	= I -	Receive Data Negative: Serial line B negative signal for Current Loop serial bipolar receive.
RXB+ C.L.	= I -	Receive Data Positive: Serial line B positive signal for Current Loop serial bipolar receive.
TXB- C.L.	= O -	Transmit Data Negative: Serial line B negative signal for Current Loop serial bipolar transmit.
TXB+ C.L.	= O -	Transmit Data Positive: Serial line B positive signal for Current Loop serial bipolar transmit.
+5 Vdc/GND	= I -	+5 Vdc or ground signal.
GND	= -	Ground signal.

NOTE
Handshake signal CTSB must be connected to another serial system, if it is going to be software managed. This is essential to acquire its status without ambiguity.

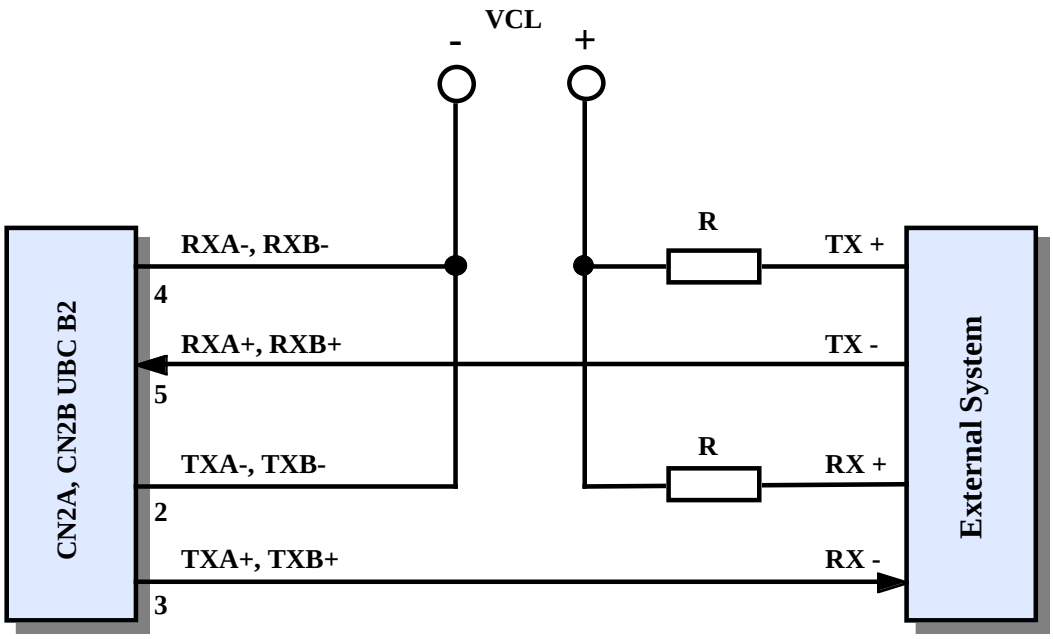


FIGURE 11: 4 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

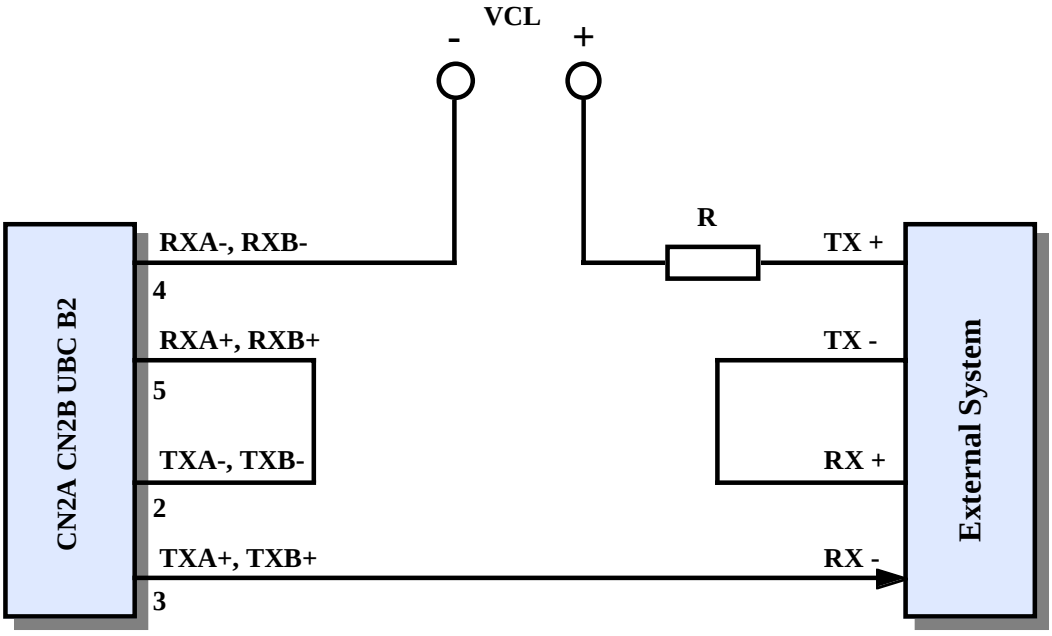


FIGURE 12: 2 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

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;
- 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).

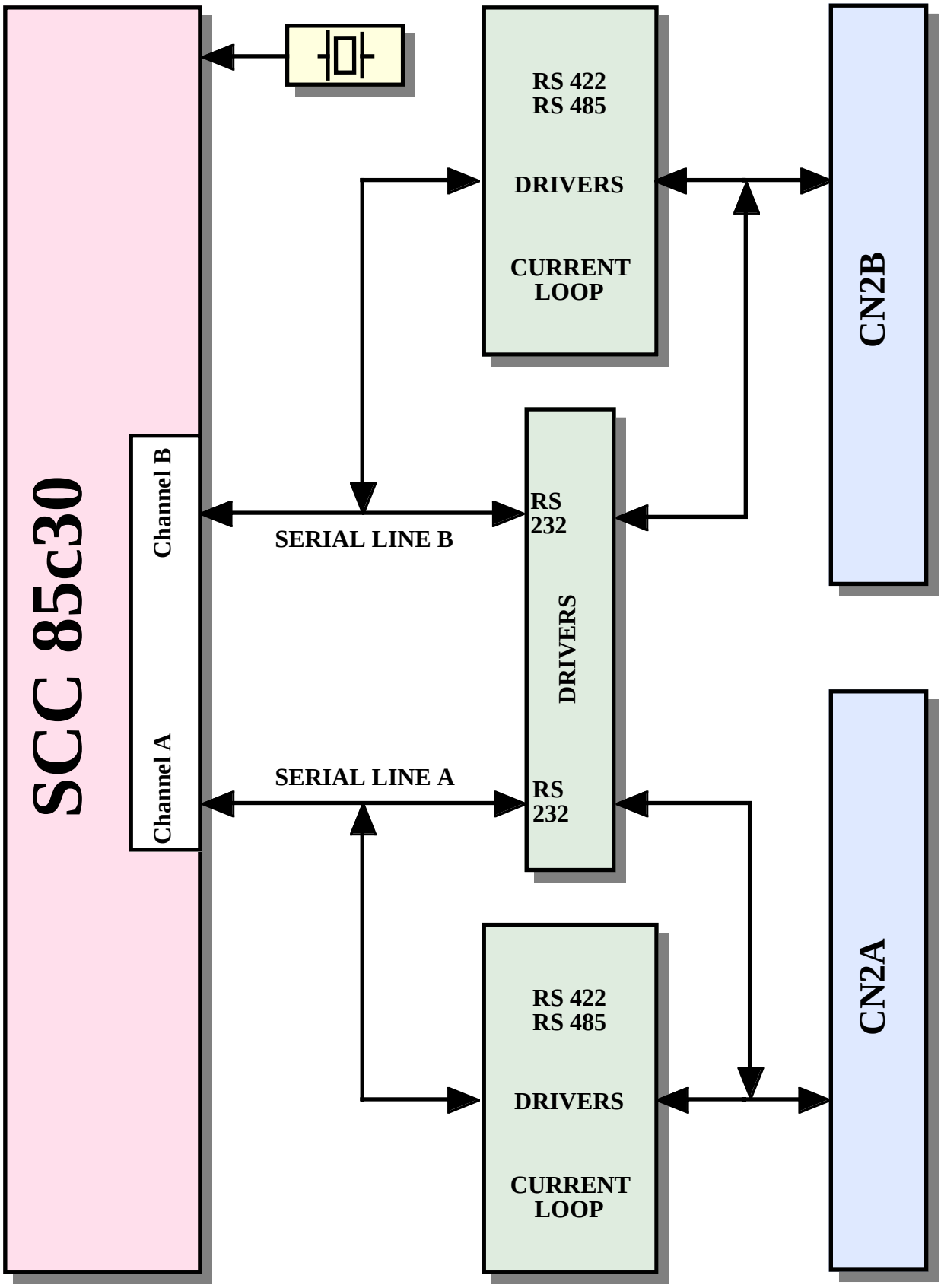


FIGURE 13: SERIAL COMMUNICATION DIAGRAM

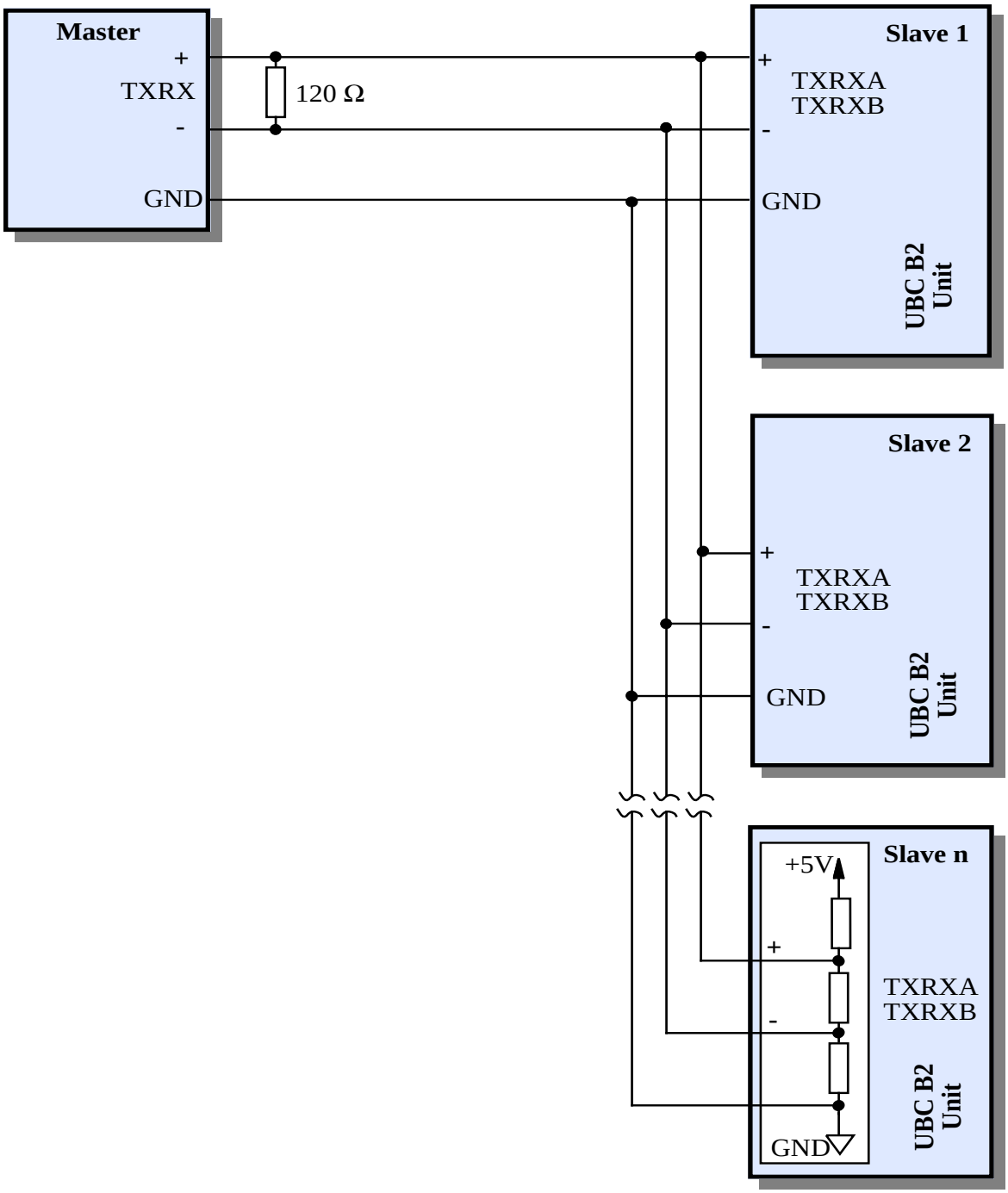


FIGURE 14: RS 485 NETWORK CONNECTION EXAMPLE

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 **UBC B2** 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.

JUMPERS

On **UBC B2** there are 9 jumpers (2 of them are solder 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	N. PINS	PURPOSE
J1	3	Connects to BUS **ABACO®** I/O BUS the signal /INT coming from SCC 85c30.
J2, J3	2	Connect RS 422 and RS 485 forcing and terminating circuitery of serial line B.
J4	3	Selects directionality and activation modality of serial line B in RS 422 and RS 485.
J5	3	Selects directionality and activation modality of serial line A in RS 422 and RS 485.
J6, J7	2	Connect RS 422 and RS 485 forcing and terminating circuitery of serial line A.
JS1	3	Selects connection for pin 1 of CN2A.
JS2	3	Selects connection for pin 1 of CN2B.

FIGURE 15: JUMPERS SUMMARIZING TABLE

The following tables describe all the right connections of **UBC B2** jumpers with their relative functions. To recognize these valid connections, please refer to the board printed diagram (serigraph) or to figures 1 or 2 of this manual, where the pins numeration is listed, while for recognizing jumpers location, please refer to figures 18 and 19.
The "*****" used in the following tables, denotes the default connection, or on the other hand the connection set up at the end of testing phase, that is the configuration the user receives.

2 PINS JUMPERS

JUMPER	CONNECTION	PURPOSE	DEF.
J2, J3	not connected	Do not connect forcing and terminating circuitery to RS 485 serial line or to RS 422 reception line of serial line B.	*
	connected	Connect forcing and terminating circuitery to RS 485 serial line or to RS 422 reception line of serial line B.	
J6, J7	not connected	Do not connect forcing and terminating circuitery to RS 485 serial line or to RS 422 reception line of serial line A.	*
	connected	Connect forcing and terminating circuitery to RS 485 serial line or to RS 422 reception line of serial line A.	

FIGURE 16: 2 PINS JUMPERS TABLE

3 PINS JUMPERS

JUMPERS	CONNECTION	PURPOSE	DEF.
J1	position 1-2	Connects signal /INT of SCC 85c30 to signal /INT of BUS **ABACO®** I/O BUS (pin 23 on CN1).	
	position 2-3	Connects signal /INT of SCC 85c30 to signal /NMI of BUS **ABACO®** I/O BUS (pin 24 on CN1).	
	not connected	Does not connect signal /INT of SCC 85c30 to BUS **ABACO®**.	*
J4	position 1-2	Selects communication for serial line B in RS 485 (half duplex with 2 fili).	
	position 2-3	Selects communication for serial line B in RS 422 (full duplex or half duplex with 4 fili).	*
J5	position 1-2	Selects communication for serial line A in RS 485 (half duplex with 2 fili).	
	position 2-3	Selects communication for serial line A in RS 422 (full duplex or half duplex with 4 fili).	*
JS1	position 1-2	Connects pin 1 of CN2A to GND.	*
	position 2-3	Connects pin 1 of CN2A to +5 Vdc.	
JS2	position 1-2	Connects pin 1 of CN2B to GND.	*
	position 2-3	Connects pin 1 of CN2B to +5 Vdc.	

FIGURE 17: 3 PINS JUMPERS TABLE

SOLDER JUMPERS

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

I/O CONNECTION

To prevent possible connecting problems between **UBC B2** 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 (GND). For TTL signals, the 0 Vdc level corresponds to logic state "0", while 5Vdc level corrisponds to logic state "1".

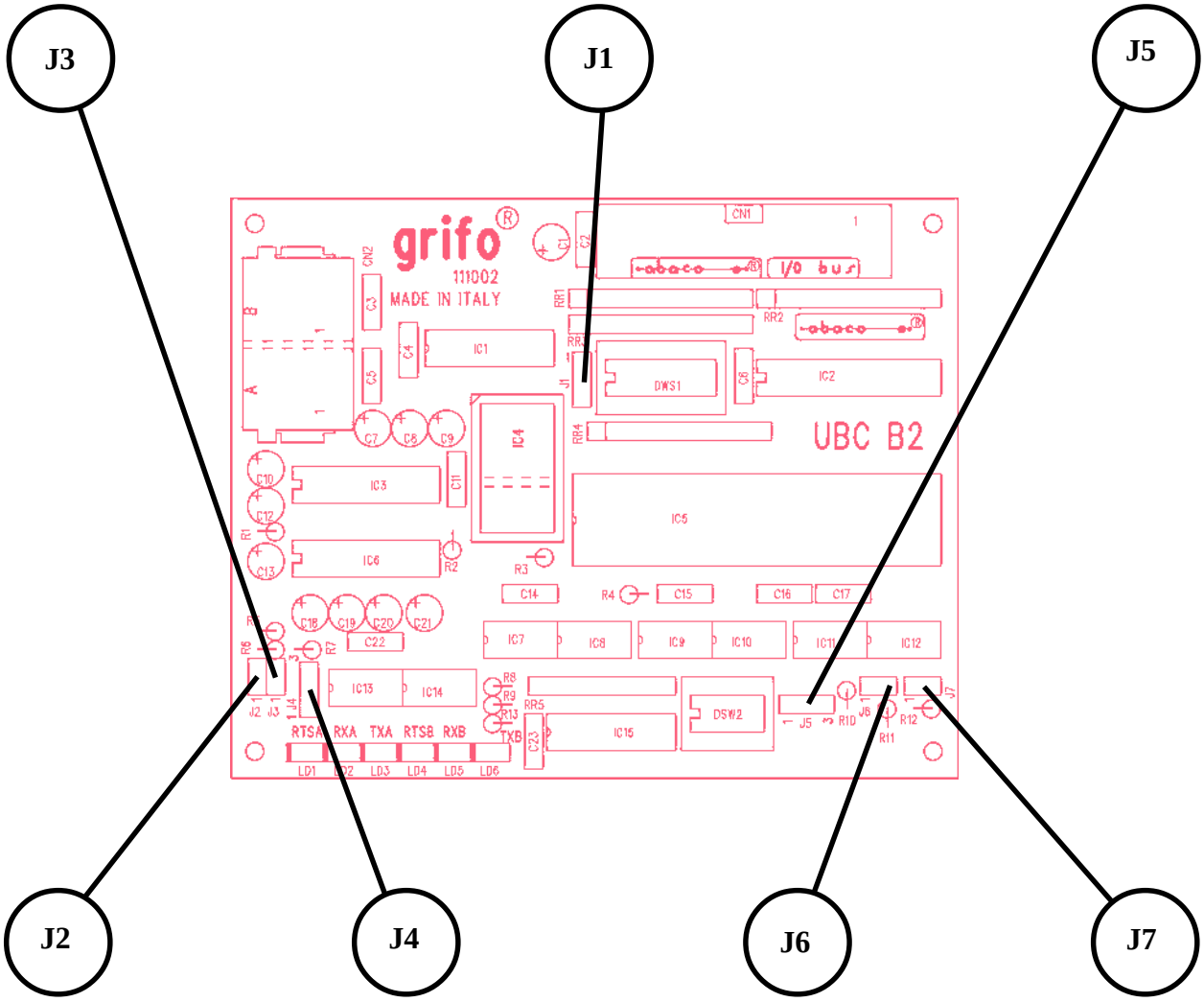


FIGURE 18: JUMPERS LOCATION (COMPONENT SIDE)

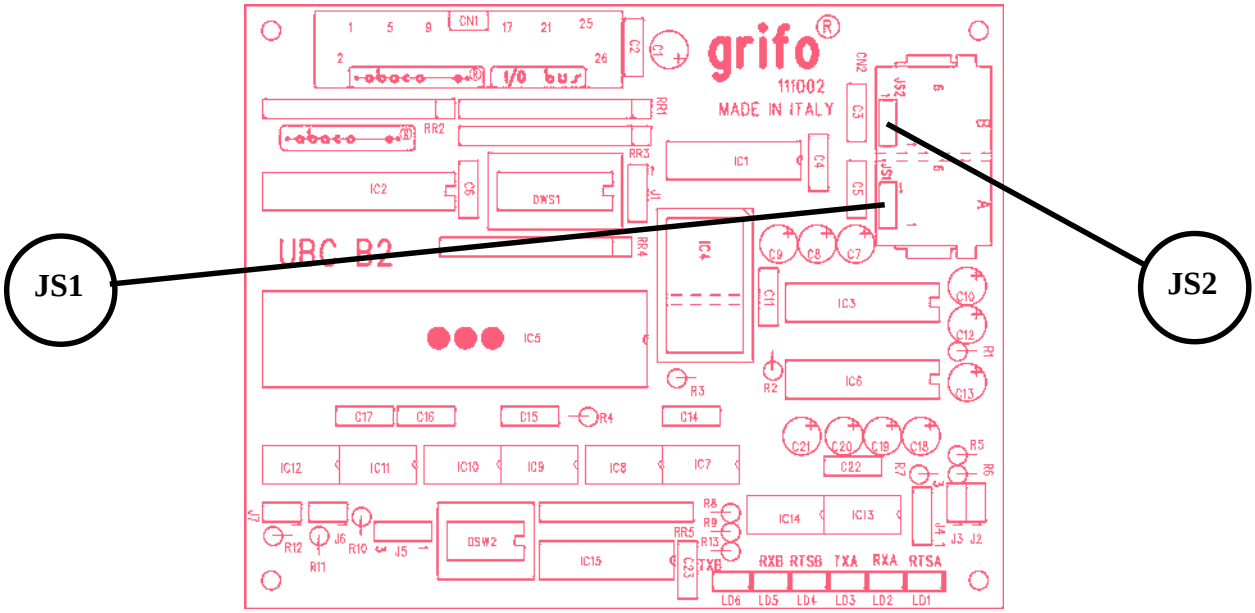


FIGURE 19: JUMPERS LOCATION (SOLDER SIDE)

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 CN2A 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 (jumpers not mentioned in the below description have no influence on communication):

- SERIAL LINE A CONFIGURED IN RS 232 (default configuration)

			IC6	= driver MAX 202
J6, J7	=	not connected	IC9	= no component
J5	=	don't care	IC10	= no component
			IC11	= no component
			IC12	= no component

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

			IC6	= no component
J6, J7	=	not connected	IC9	= driver HP 4100
J5	=	don't care	IC10	= driver HP 4200
			IC11	= no component
			IC12	= no component

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 11÷14.

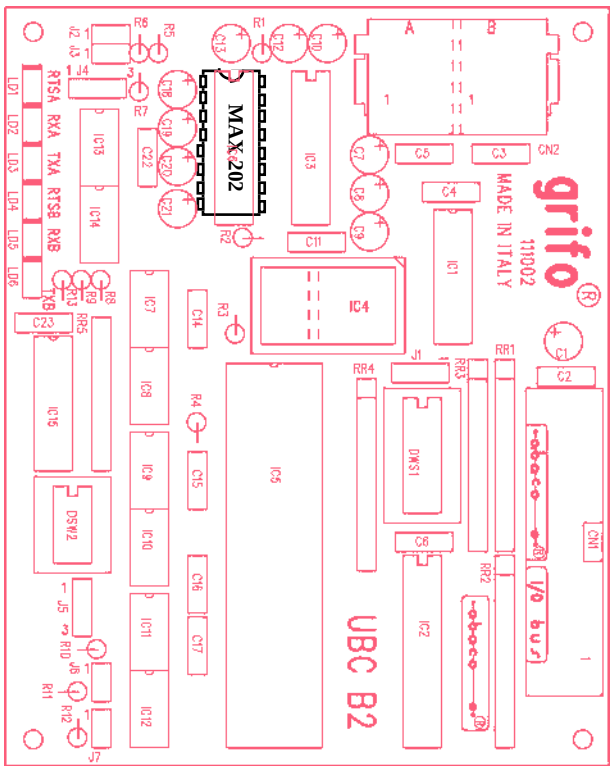
- SERIAL LINE A CONFIGURED IN RS 422 (**.RS422** option)

			IC6	= no component
J6, J7	=	(*1)	IC9	= no component
J5	=	position 2-3	IC10	= no component
			IC11	= driver MAX 483 or SN 75176
			IC12	= driver MAX 483 or SN 75176

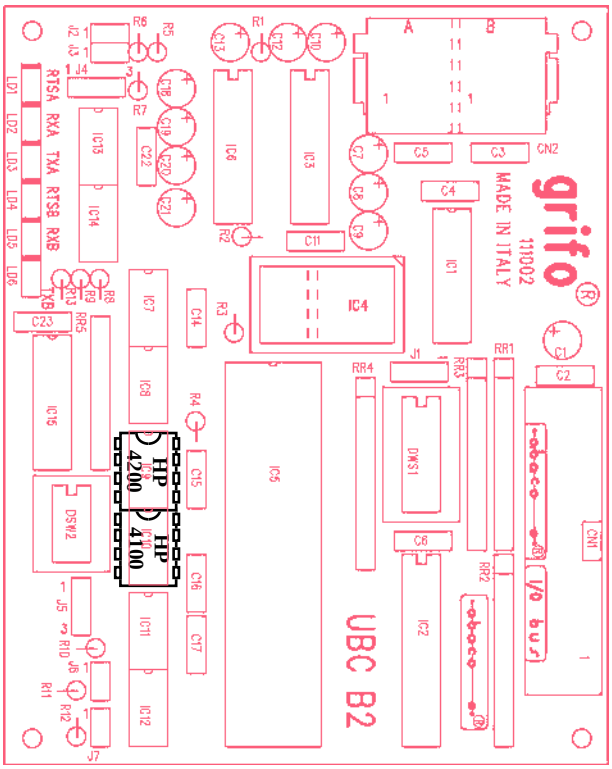
Status of signal /RTSA, which is software managed, allows to enable or disable the transmitter as follows:

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

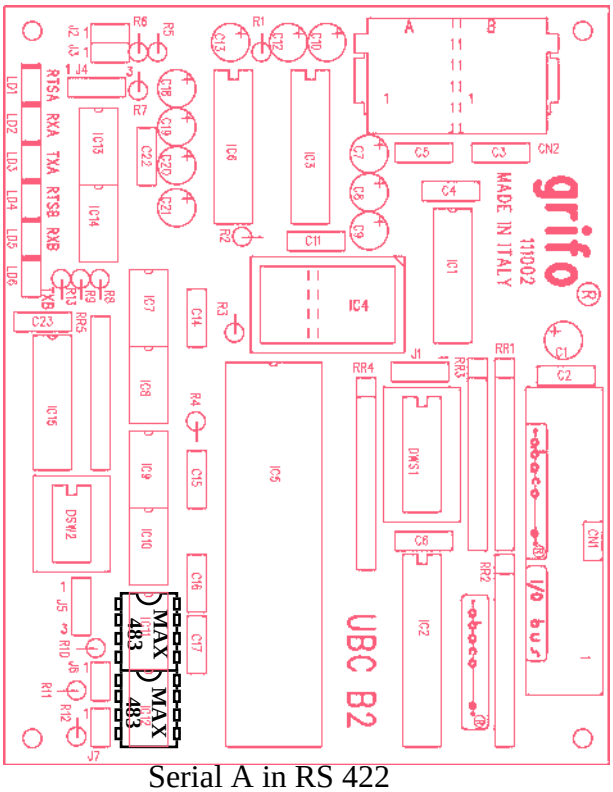
In point to point connections, signal /RTSA can be always kept low (trasnmitter always enabled), while in multi point connections transmitter must be enabled only when a transmission is requested.



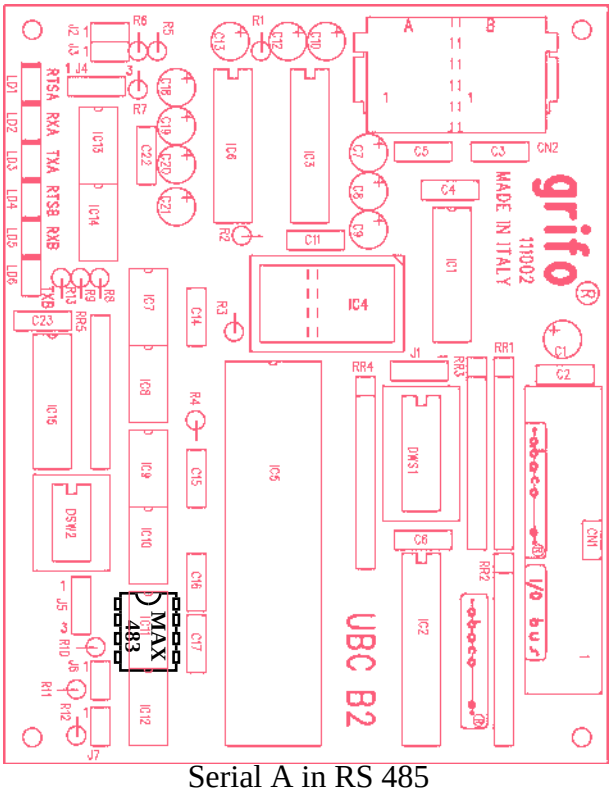
Serial A in RS 232



Serial A in Current Loop



Serial A in RS 422



Serial A in RS 485

FIGURE 20: SERIAL LINE A COMMUNICATION DRIVERS LOCATION

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

			IC6	= no component
J6, J7	=	(*1)	IC9	= no component
J5	=	position 1-2	IC10	= no component
			IC11	= driver MAX 483 or SN 75176
			IC12	= no component

In this modality the signals to use are pins 4 and 5 of connector CN2A, that become transmission or reception lines according to the status of signal /RTSA, managed by software, as follows:

/RTSA = low level	= logic state 0	->	transmitter enabled
/RTSA = 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.

(*1) 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 J6 and J7. 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.

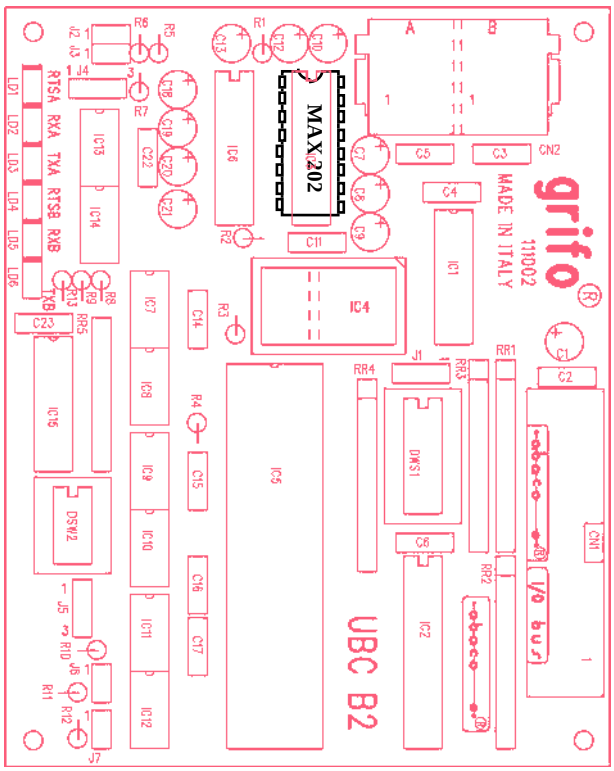
- SERIAL LINE B CONFIGURED IN RS 232 (default configuration)

			IC3	= driver MAX 202
J2, J3	=	not connected	IC7	= no component
J4	=	don't care	IC8	= no component
			IC13	= no component
			IC14	= no component

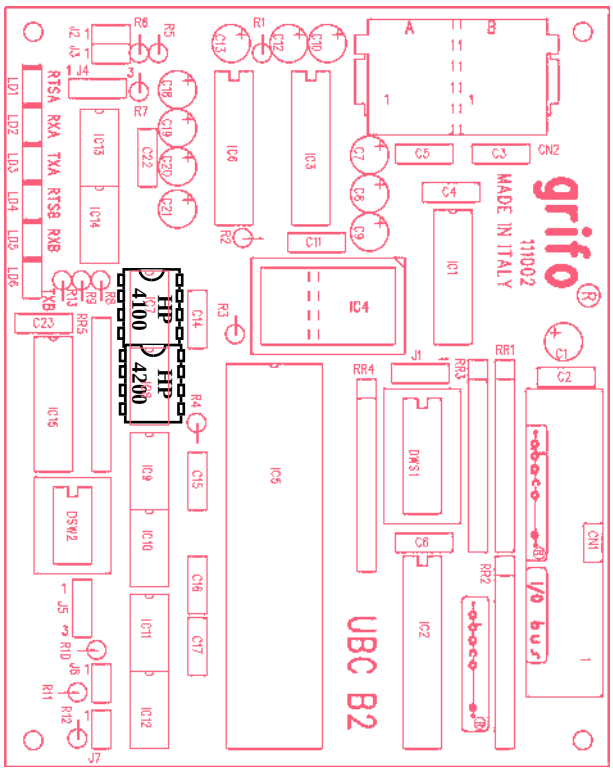
- SERIAL LINE B CONFIGURED IN CURRENT LOOP (**.CLOOP** option)

			IC3	= no component
J2, J3	=	not connected	IC7	= driver HP 4100
J4	=	don't care	IC8	= driver HP 4200
			IC13	= no component
			IC14	= no component

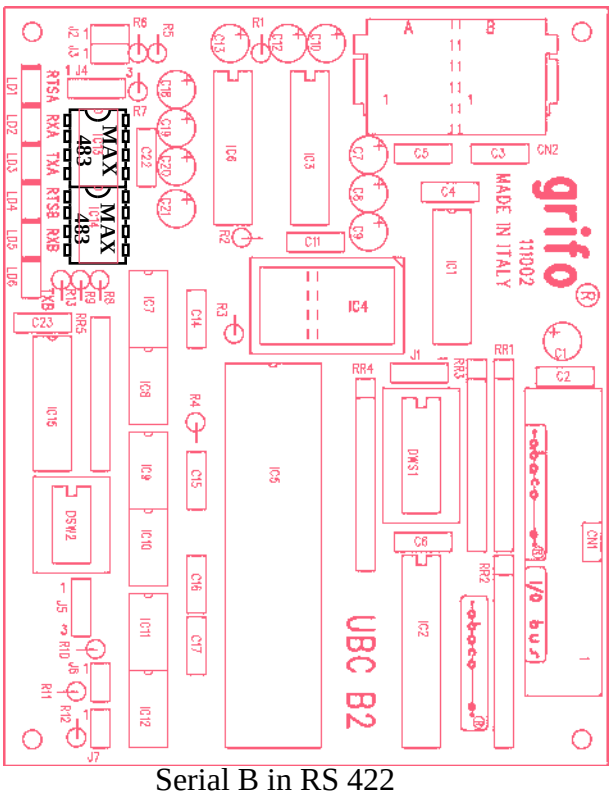
Please remark that current loop serial interface is passive, so it must be connected an active Current Loop serial line, that is a line provided with its own power supply. Current loop interface can be employed to make both point to point and multi point connections through a 2 wires or a 4 wires connection as described in figures 11÷14.



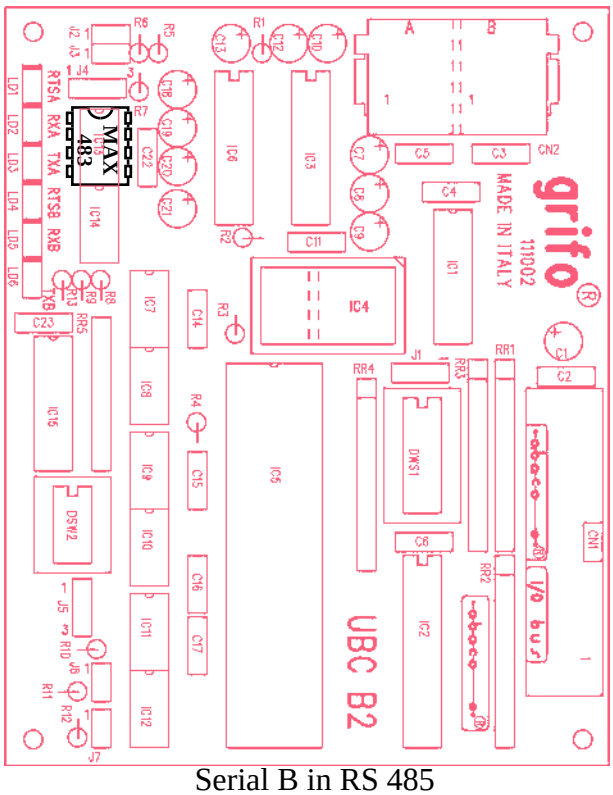
Serial B in RS 232



Serial B in Current Loop



Serial B in RS 422



Serial B in RS 485

FIGURE 21: SERIAL LINE B COMMUNICATION DRIVERS LOCATION

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

			IC3	= no component
J2, J3	=	(*2)	IC7	= no component
J4	=	position 2-3	IC8	= no component
			IC13	= driver MAX 483 or SN 75176
			IC14	= driver MAX 483 or SN 75176

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

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

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

SERIAL LINE B CONFIGURED IN RS 485 (**.RS485** option)

			IC3	= no component
J2, J3	=	(*2)	IC7	= no component
J4	=	position 1-2	IC8	= no component
			IC13	= driver MAX 483 or SN 75176
			IC14	= no component

In this modality the signals to use are pins 4 and 5 of connector CN2B, that become transmission or reception lines according to the status of signal /RTSB, managed by software, as follows:

/RTSB = low level = logic state 0 -> transmitter enabled
/RTSB = 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.

(*2) 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 J2 and J3. 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, signals /RTSA and /RTSB are kept to a logic level high, so in any of these two cases driver RS 485 is receiving or RS 422 transmission is disabled, avoiding eventual conflicts in communication.
For further information please refer to figures 6÷14 or to SCC 85C30 paragraph.

INTERRUPTS

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 manual of **GPC®** card used.

- SCC 85c30 serial lines -> They generate an /NMI or an /INT interrupt on BUS **ABACO®**, I/O BUS according to the connection of jumper J1.

Setting the conditions that generate the interrupts is completely software manageable by programming SCC 85c30 registers.
Please remark that **UBC B2** structure allows to use more that one card with interrupt actived at the same time but vectored interrupts cannot be used.

CONFIGURATION INPUT

UBC B2 board is provided with one 4 pins dip switch (DSW2), typically used for application configuration, that can be read by software. The most frequent applications are: working condition selection or on board firmware parameters setting, like, for example: language setting, id code inside a serial communication network, communication protocol selection, test modalities selection, configuration mode selection, etc.
The four switches of DSW2 are connected to as many handshake signals of SCC 85c30, according to the following table:

DSW2.1	->	**/DCDA**
DSW2.2	->	**/DCDB**
DSW2.3	->	**/SYNCA**
DSW2.4	->	**/SYNCB**

Please refer to chapter “PERIPHERAL DEVICES SOFTWARE DESCRIPTIO” for information about how to read them, while to locate them on the board please refer to figures 1 and 5.

ADDRESSES AND MAPS

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.

BOARD MAPPING

UBC B2 board is mapped into a **4** bytes I/O addressing space, that can be allocated starting from different base addresses according to how the board is configured. This feature allows to use several **UBC B2** cards on the same **ABACO®** I/O BUS, or to install them on a BUS where other peripheral modules are installed obtaining a structure that can be expanded without any difficulty or modifications to the application software. These bytes allow the complete control of board settings and status and the complete flow of input and output data.
The base address can be defined through the specific BUS interface circuitry on the board itself; this circuitry uses the eight pins dip switch called **DSW1**, from which it reads the address set by the user. Here follows the corrispondance between dips configuration and address signals, to easily locate such component please refer to figure 5.

DSW1.1	->	Address A2
DSW1.2	->	Address A3
DSW1.3	->	Address A4
DSW1.4	->	Address A5
DSW1.5	->	Address A6
DSW1.6	->	Address A7

These dips are driven in complemented logic, this means that if a switch is **ON** generates a **logic zero**, viceversa if a switch is **OFF** generates a **logic one**.

NOTE
When allocating the mapping address of the boards, plase be careful not to allocate more than one device in the same addressing space (count also the number of bytes occupied by the card). If this condtion will not be respected, a BUS conflict will happen; such conflict will compromise the correct working of the whole system.

As an example, dip configuration to set address 040H is repored here:

DSW1.1	->	ON
DSW1.2	->	ON
DSW1.3	->	ON
DSW1.4	->	ON
DSW1.5	->	OFF
DSW1.6	->	ON

To locate them on the board please refer to figures 1 and 5.

I/O ADDRESSES

The on board control logic manages the allocation of all the peripheral devices registers in the microprocessor I/O space, that is 256 bytes long.
Indication <**baseaddr**> means the base address of the board decided with DSW1, as previously described.

DEVICE	REG.	ADDRESS	R/W	PURPOSE
SCC 85C30	RSB	<baseaddr>+00H	R/W	Serial line B status register.
	RDB	<baseaddr>+01H	R/W	Serial line B data register.
	RSA	<baseaddr>+02H	R/W	Serial line A status register.
	RDA	<baseaddr>+03H	R/W	Serial line A data register.

FIGURE 22: INTERNAL REGISTERS ADDRESSES TABLE

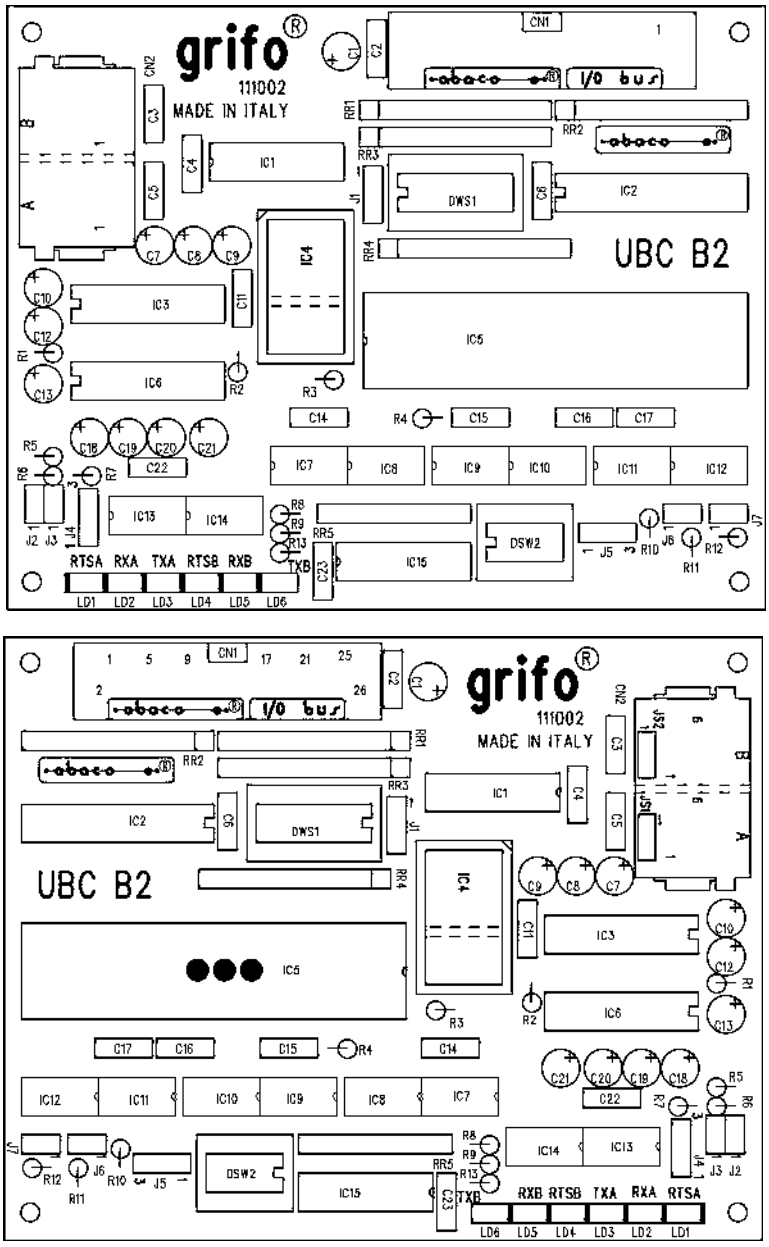


FIGURE 23: COMPONENTS MAP (COMPONENTS SIDE ABOVE, SOLDER SIDE BELOW)

PERIPHERAL DEVICES SOFTWARE DESCRIPTION

In the previous paragraphs are described the external registers addresses, while in this one there is a specific description of registers meaning and function (please refer to I/O addresses table, for the registers names and addresses values). For a more detailed description of the devices, please refer to manufacturing company documentation. In the following paragraphs the **D7÷D0** and **.0÷7** indications denote the eight bits of a register.

CONFIGURATION INPUT

UBC B2 board is provided with one 4 pins dip switch (DSW2) that can be read by software as described here.
The four switches of DSW2 are connected to as many hardware handshake signals of SCC 85c30, and can be acquired by software performing a read operation from status registers of the two sections with the following correspondace:

DSW2.1	->	**RSA.3** (/DCDA)
DSW2.2	->	**RSB.3** (/DCDB)
DSW2.3	->	**RSA.4** (/SYNCA)
DSW2.4	->	**RSB.4** (/SYNCB)

These dips are driven in complemented logic, this means that if a switch is **ON** generates a **logic zero**, viceversa if a switch is **OFF** generates a **logic one**.
Status registers of SCC 85c30 are described in the following paragraph.

SCC 85C30

This peripheral manages two independent serial lines, called A and B. Communication mode can be:
- Synchronous
- Asynchronous
- SDLC/HDLC

This peripheral is managed through 4 registers allocated in he I/O addressing space. Two of these registers, RS and RSB, are used to set and get the peripheral status (one for each line), while regisers RDA and RDB are used for data transfer. Both status and data registers can be accessed in input (to acquire the peripheral status o received data) and in output (to program the peripheral or to data to send). SCC 85C30 uses an indirect addressing mode to its internal registers; this consists in writing proper values to the status register to reach the whole set of internal registers. These latter are 16 writable and 9 readable, and are described in the following pages .

Write register 0 (Command Register)
This register is used for controlling the peripheral, resetting certain of its states and pointing to other internal registers:

	D7	D6	D5	D4	D3	D2	D1	D0
WR0 =	**CRC1**	**CRC0**	**CD2**	**CD1**	**CD0**	**P2**	**P1**	**P0**

where:

CRC1 CRC0 = CRC reset codes 0 and 1
0 0 -> Null code
0 1 -> Reset receive CRC checker
1 0 -> Reset transmit CRC generator
1 1 -> Reset transmit CRC Underrun/End of transmission latch

CD2 CD1 CD0 = Command code selection
0 0 0 -> Null code
0 0 1 -> Point to high registers WR8÷WR15, RD8÷RD15
0 1 0 -> Reset external/status interrupts
0 1 1 -> Send Abort in SDLC mode
1 0 0 -> Enable interrupt on next received char
1 0 1 -> Reset transmission interrupt pending
1 1 0 -> Error reset
1 1 1 -> Reset highest priority Interrupt Under Service (IUS)

P2 P1 P0 = Register selection code, used in conjunction with "Point to high registers" command code (see above)
0 0 0 -> WR0, RD0, WR8, RD8
0 0 1 -> WR1, RD1, WR9
0 1 0 -> WR2, RD2, WR10, RD10
0 1 1 -> WR3, RD3, WR11
1 0 0 -> WR4, WR12, RD12
1 0 1 -> WR5, WR13, RD13
1 1 0 -> WR6, WR14
1 1 1 -> WR7, WR15, RD15

Write register 1 (Interrupt and data transfer mode definition)
Transmit /Receive interrupt and data transfer mode definition; Wait/Ready mode definition:

	D7	D6	D5	D4	D3	D2	D1	D0
WR1 =	**AWR**	**W/R**	**R/T**	**IM1**	**IM0**	**P**	**AIT**	**AIE**

where:

AWR = Wait/Request pin enable: AWR=0 -> disabled
W/R = Wait/Request function enable: W/R=0 -> /Wait function
R/T = Wait/Rquest function on receive or transmit: R/T=0 -> transmit
IM1 IM0 = Receive interrupt modes selection
0 0 -> Receive interrupts disabled
0 1 -> Receive interrupt on first character or special condition
1 0 -> Receive interrupt on all characters or special condition
1 1 -> Receive interrupt on first character or special condition
P -> Enable parity not matching as special condition: P=1 -> enabled
AIT -> Enable interrupt when transmit buffer becomes empty: AIT=1 -> enabled
AIE -> External/status master interrupt enable: AIE=1 -> enabled

Write register 2 (Interrupt vector)
Only one interrupt vector exists but it can be accessed through either channels:

	D7	D6	D5	D4	D3	D2	D1	D0
WR2 =	**V7**	**V6**	**V5**	**V4**	**V3**	**V2**	**V1**	**V0**

where:

Vn	= Interrupt vector n-th bit

Write register 3 (Receive parameters and control)
This register contains control bits and parameter for receiver logic:

	D7	D6	D5	D4	D3	D2	D1	D0
WR3 =	**R1**	**R0**	**AA**	**IF**	**AR**	**RI**	**CS**	**A**

where:

R1 R0	= Number of bits per character
0 0	-> 5 bit
0 1	-> 6 bit
1 0	-> 7 bit
1 1	-> 8 bit
AA	= Handshake enables: AA=1 -> enabled
IF	= Enable hunt mode for synchronization: IF=1 -> enabled
AR	= Receiver CRC enable: AR=1 -> enabled
RI	= Enable address search mode for SDLC: RI=1 -> enabled
CS	= SYNC character load inhibit: CS=1 -> inhibited
A	= Receiver enable (set as after all other bits): A=1 -> enabled

Write register 4 (Miscellaneous parameters)
Receiver/transmitter miscellaneous parameters and modes:

	D7	D6	D5	D4	D3	D2	D1	D0
WR4 =	**VC1**	**VC0**	**MS1**	**MS0**	**BS1**	**BS0**	**P/D**	**P**

where:

VC1 VC0	= Clock rate 1 and 0
0 0	-> Data rate = clock rate (BITRATE=1)
0 1	-> Data rate = 1/16 fclock rate (BITRATE=16)
1 0	-> Data rate = 1/32 clock rate (BITRATE=32)
1 1	-> Data rate = 1/64 clock rate (BITRATE=64)
MS1 MS0	= SYNC modes 1 and 0
0 0	-> 6 or 8 bits character synchronization (see Sync bit in WR10)
0 1	-> 12 or 16 bits character synchronization (see Sync bit in WR10)
1 0	-> SDLC mode (flag to WR7 is 01111110)
1 1	-> External sync mode
BS1 BS0	= Stop bits 1 and 0 in asynchronous mode
0 0	-> Synchronous modes selection
0 1	-> 1 stop bit per character
1 0	-> 1+1/2 stop bits per character
1 1	-> 2 stop bits per character
P/D	= Even or odd parity: P/D=1 -> even parity
P	= Parity check enable: P=1 -> enabled

Write register 5 (Transmit parameters and control)
This register contains the bits that influence the transmitter behaviour (C/S influences also the receiver):

	D7	D6	D5	D4	D3	D2	D1	D0
WR5 =	**DTR**	**BC1**	**BC0**	**IB**	**AT**	**C/S**	**RTS**	**A**

where:

DTR = /DTR pin enable: DTR=1 -> /DTR is active when low

BC1	BC0	=Number of bits per character in transmission selection
0	0	-> 5 or less bits
0	1	-> 7 bits
1	0	-> 6 bits
1	1	-> 8 bits

IB = Send break: IB=1 -> send
AT = Transmit enable: AT=1 -> enabled
C/S = Polynomial CRC selection: C/S=1 -> CRC 16 polynomial
C/S=0 -> SDLC polynomial
RTS = /RTS pin enable: RTS=1 -> /RTS is active when low
A = Transmit CRC enable: A=1 -> enable

Write register 6 (Sync characters or SDLC address field)
This register contains the Sync character in monosync mode, the fist bye of 16-bits sync character in external sync mode or the secondary address field used in SDLC mode:

	D7	D6	D5	D4	D3	D2	D1	D0
WR6 =	**S7**	**S6**	**S5**	**S4**	**S3**	**S2**	**S1**	**S0**

where:

S7	S6	S5	S4	S3	S2	S1	S0	
SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	-> Monosync 8 bits
SYNC1	SYNC0	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	-> Monosync 6 bits
SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	-> Bisync 16 bits
SYNC3	SYNC2	SYNC1	SYNC0	1	1	1	1	-> Bisync 12 bits
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	-> SDLC
ADR7	ADR6	ADR5	ADR4	X	X	X	X	-> SDLC add. range

Write register 7 (SYNC character od SDLC flag)
This register contains the rest of the synchronization informations:

	D7	D6	D5	D4	D3	D2	D1	D0
WR7 =	**S15**	**S14**	**S13**	**S12**	**S11**	**S10**	**S9**	**S8**

where:

S15	S14	S13	S12	S11	S10	S9	S8	
SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	-> Monosync 8 bits
SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	X	X	-> Monosync 6 bits
SYNC15	SYNC14	SYNC13	SYNC12	SYNC11	SYNC10	SYNC9	SYNC8	-> Bisync 16 bits
SYNC11	SYNC10	SYNC9	SYNC8	SYNC7	SYNC6	SYNC5	SYNC4	-> Bisync 12 bits
0	1	1	1	1	1	1	0	-> SDLC

Write register 8 (Transmit buffer)
This is the transmit buffer register:

	D7	D6	D5	D4	D3	D2	D1	D0
WR8 =	**TX7**	**TX6**	**TX5**	**TX4**	**TX3**	**TX2**	**TX1**	**TX0**

where:

TXn = n-th bitof data to transmit

Write register 9 (Master interrupt control)
This register contains the interrput control bits and allows to reset the USRT channels:

	D7	D6	D5	D4	D3	D2	D1	D0
WR9=	**R1**	**R0**	**0**	**SH/L**	**MIE**	**DLC**	**NV**	**VIS**

where:

R1 R0 = UART reset command bits
0 0 -> No reset
0 1 -> Channel B reset
1 0 -> Channel A reset
1 1 -> Force hardware reset
0 = Not used (must be zero)
SH/L = Which vector bits modify to indicate status: SH/L=1 -> modify V6÷V4
SH/L=0 -> modify V3÷V1
MIE = Master interrupt enable: MIE=1 -> interrupts enabled
DLC = Disable lower daisy chain: DLC=1 -> disabled
NV = Disable interrupt vector output : NV=1 -> disabled
VIS = Interrupt vector includes status bits: VIS=1 -> variable vector

Write register 10 (Miscellaneous control bits)
Miscellaneous Transmitter/Receiver control bits:

	D7	D6	D5	D4	D3	D2	D1	D0
WR10 =	**CRC**	**FM1**	**FM0**	**GP**	**MFI**	**AFU**	**LM**	**S**

where:

CRC = CRC presets: CRC=1 -> CRC preset to 1
CRC=0 -> CRC preset to 0
FM1 FM0 = Data encoding 1 and 2
0 0 -> Modo NRZ
0 1 -> Modo NRZI
1 0 -> Modo FM1 (transition high)
1 1 -> Modo FM0 (transition low)
GP = Go active on POLL: GP=1 -> enabled
MFI = SDLC idle line condition: MFI=1 -> send "1s"
MF1=0 -> send flags
AFU = Ensble SDLC send abort on transmit underrun : AFU=1 enableed
LM = Enable loop mode: LM=1 -> enabled
S = Select SYNC char length: S=1 -> 6 bits
S=0 -> 8 bits

Write register 11 (Clock mode control)
This register allows to select the source of both the receive and transmit clock:

	D7	D6	D5	D4	D3	D2	D1	D0
WR11 =	**XT**	**RC1**	**RC0**	**TC1**	**TC0**	**TR**	**TR1**	**TR0**

where:

XT clock source on pin /RTxC: XT=1 -> source is quartz crystal
XT=0 -> source is TTL-compatible signal

RC1 RC0 = Receiver clock 1 and 0
0 0 -> Receive clock = pin /RTxC
0 1 -> Receive Clock = pin /TRxC
1 0 -> Receive Clock = baud rate generator output
1 1 -> Receive Clock = DPLL output

TC1 TC0 = Transmit clock 1 and 0
0 0 -> Transmit Clock = pin /RTxC
0 1 -> Transmit Clock = pin /TRxC
1 0 -> Transmit Clock= baud rate generator outputo
1 1 -> Transmit Clock = DPLL output

TR = Select direction of pin /TRxC: TR=0 -> input
TR=1 -> output

TR1 TR0 = /TRxC output source
0 0 -> XTAL oscillator output
0 1 -> transmit clock
1 0 -> baud rate generator output
1 1 -> DPLL output

Write registera 12 and 13 (Baud rate generator time constant)
These registers contain the baud rate generator time constant:

	D7	D6	D5	D4	D3	D2	D1	D0
WR12 =	**TC7**	**TC6**	**TC5**	**TC4**	**TC3**	**TC2**	**TC1**	**TC0**
WR13 =	**TC15**	**TC14**	**TC13**	**TC12**	**TC11**	**TC10**	**TC9**	**TC8**

The time constant to write in registers 12 and 13 can be calculated using the following formula:

$$TC = (11059200 / (2 * BITRATE * BAUDRATE)) - 2$$

where BAUDRATE is the desired rate in bits per second and BITRATE is the value written in write register 4. Suggested value is 16.

Write register 14 (Miscellaneous control bits)
This register contains miscellaneous control bits:

	D7	D6	D5	D4	D3	D2	D1	D0
WR14 =	**C2**	**C1**	**C0**	**LL**	**AE**	**DTR**	**BRS**	**BRE**

where:

C2	C1	C0	= Digital pahse-locked loop (DPLL) command bits
0	0	0	-> Null command
0	0	1	-> Enter search mode
0	1	0	-> Reset missing clock
0	1	1	-> Disable DPLL
1	0	0	-> Set baud rate generator as clock source
1	0	1	-> Set pin /RTxC as clock source
1	1	0	-> Set FM mode
1	1	1	-> Set NRZI mode

LL = Enable local loopback: LL=1 -> enabled
AE = Enable auto echo: AE=1 -> enabled
DTR = Enable /DTR: DTR=1 -> enabled
BRS = select baud rate generator source: BRS=1 -> pin PCLK
BRS=0 -> /RTxC or XTAL
BRE = Enable baud rate generator: BRE=1 -> enable

Write register 15 (External/status interrupt control)
This register selects special interrupt sources:

	D7	D6	D5	D4	D3	D2	D1	D0
WR15 =	**BA**	**TU**	**CTS**	**SU**	**DCD**	**0**	**ZC**	**0**

where:
BA = Generates interrupt on break/abort
TU = Generates interrupt on transmitter underrun/EOM
CTS = Generates interrupt on /CTS status variation
SH = Generates interrupt on SYNC pin or hunt bit variation
DCD = Generates interrupt on /DCD pin variation
0 = Not used (must be zero)
ZC = Generates interrupt when baud rate generator counter reaches 0
The above described states are referred to a logic status "1" of the corresponding bit.

Read register 0 (Buffer and external status)
This register contains bits that report receiver and transmitter buffer status and external status:

	D7	D6	D5	D4	D3	D2	D1	D0
RD0 =	**BA**	**TU**	**CTS**	**SU**	**DCD**	**TBE**	**ZC**	**RCA**

where:
BA = Break/abort occoured: BA=1 -> occoured
TU = Transmission underrun/EOM: TU=1 -> occoured
CTS = pin /CTS status
SH = pin /SYNC or hunt status
DCD = pin /DCD status
TBE = Transmission buffer status: TBE=1 -> empty
ZC = Baud rate generator zero count: ZC=1 -> zero reached
RCA = Receive character available: RCA=1 -> available

Read register 1
This register contains special receive ondition status bits and the residue code for the I-field in SDLC mode:

	D7	D6	D5	D4	D3	D2	D1	D0
RD1 =	**EOF**	**CRC**	**ROE**	**PE**	**RC0**	**RC1**	**RC2**	**AS**

where:

EOF	= End of frame (SDLC)
CRC	= CRC or framing error
ROE	= Receiver overrun error
PE	= Receive parity error
RC0	= SDLC residue code 0
RC1	= SDLC residue code 1
RC2	= SDLC residue code 2
AS	= Everithing sent

The above described states are referred to a logic status "1" of the corresponding bit.

Read register 2
This register contains the value of the interrupt vector:

	D7	D6	D5	D4	D3	D2	D1	D0
RD2 =	**V7**	**V6**	**V5**	**V4**	**V3**	**V2**	**V1**	**V0**

where:

Vn	= n-th bit of the interrupt vector

Read register 3
This is the interrupt pending register. It exists only on channel A but contains informations about both the channels. In channel B it always returns zero:

	D7	D6	D5	D4	D3	D2	D1	D0
RD3 =	**0**	**0**	**CAR**	**CAT**	**CAE**	**CBR**	**CBT**	**CBE**

where:

0	= Not used (always return a zero)
CAR	= Channel A reception interrupt pending
CAT	= Channel A transmission interrupt pending
CAE	= Channel A external/status variation interrupt pending
CBR	= Channel B reception interrupt pending
CBT	= Channel B transmission interrupt pending
CBE	= Channel B external/status variation interrupt pending

The above described states are referred to a logic status "1" of the corresponding bit.

Read register 8
This is the receive data register:

	D7	D6	D5	D4	D3	D2	D1	D0
RD8 =	**RX7**	**RX6**	**RX5**	**RX4**	**RX3**	**RX2**	**RX1**	**RX0**

where:

RXn	= n-thbit of received data

Read register 10
This register contains some miscellaneous status bits:

	D7	D6	D5	D4	D3	D2	D1	D0
RD10 =	**1CM**	**2CM**	**0**	**LS**	**0**	**0**	**OL**	**0**

where:
1CM = One clock missing
2CM = Two clocks missing
0 = Not used (always return a zero)
LS = Loop sending
OL = On loop
The above described states are referred to a logic status "1" of the corresponding bit.

Read register 12 and 13
These registers return the value stored in WR12 and WR13, that is the time constant for the baud rate generator.

	D7	D6	D5	D4	D3	D2	D1	D0
RD12 =	**TC7**	**TC6**	**TC5**	**TC4**	**TC3**	**TC2**	**TC1**	**TC0**
RD13 =	**TC15**	**TC14**	**TC13**	**TC12**	**TC11**	**TC10**	**TC9**	**TC8**

Read register 15
This register reflects the value stored in write register 15, the external/staus interrupt enable bits. For the meanings of such bits please refer to the description of write register 15. The two unused bits always return zero.

	D7	D6	D5	D4	D3	D2	D1	D0
RD15 =	**BA**	**TU**	**CTS**	**SU**	**DCD**	**0**	**ZC**	**0**

After a reset or a power on SCC 85C30 disables both the serial lines, just like their hardware sigals.

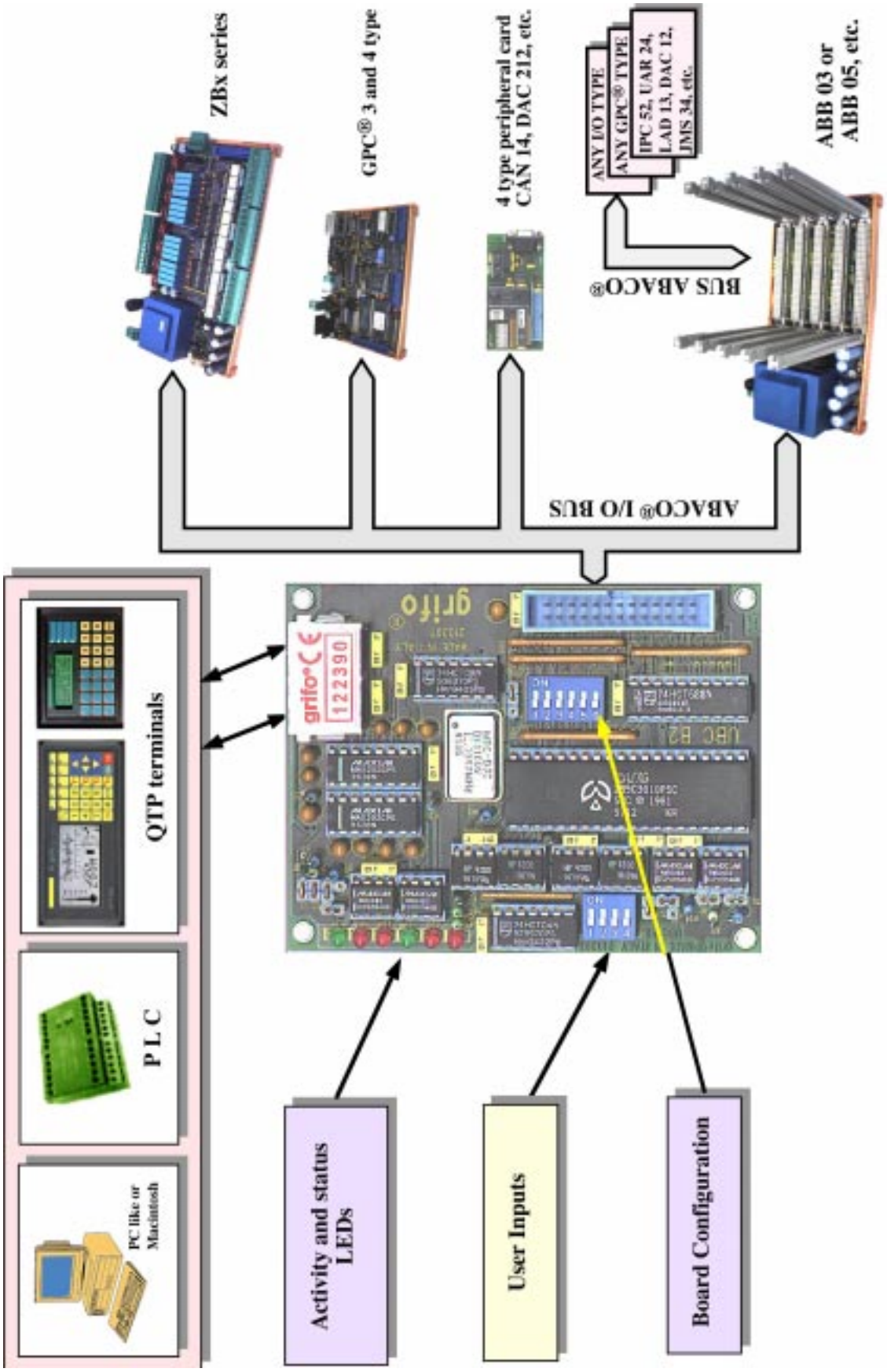


FIGURE 24: AVAILABLE CONNECTIONS DIAGRAM

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 **UBC B2**.

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

Manual HEWLETT PACKARD: *Optoelectronics Designer's Catalog*

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

Manual ZILOG: *Z8530 SCC Serial Communication Controller*

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

APPENDIX A: ALPHABETICAL INDEX