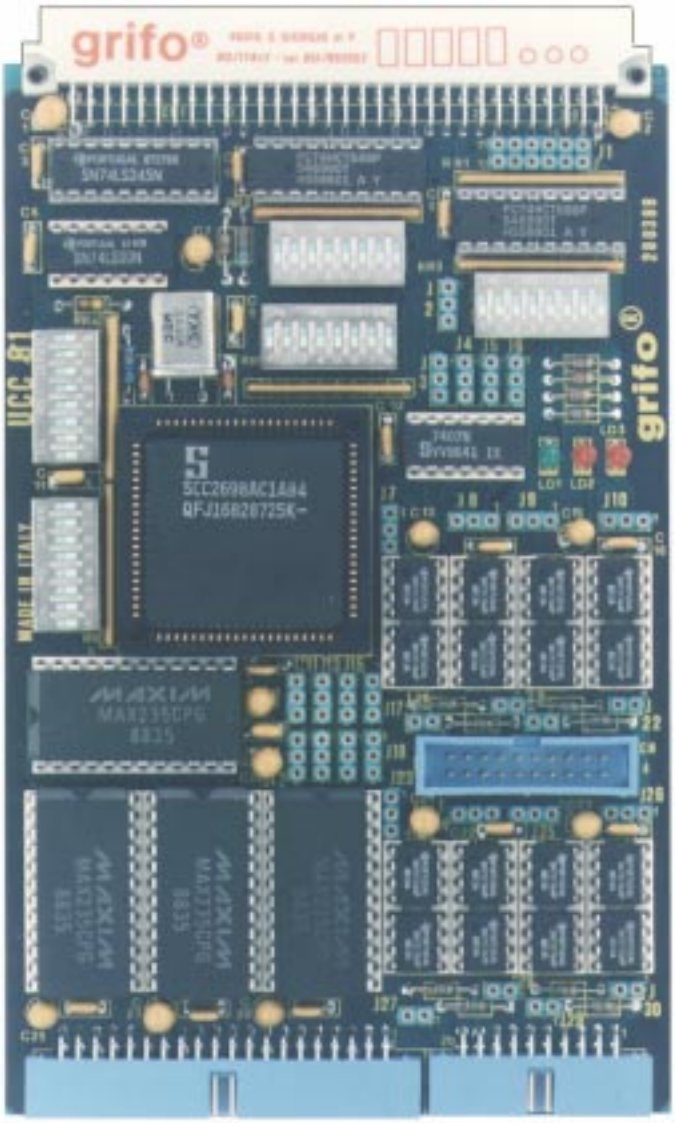
UCC 08

UART Communication Card
8 serial lines

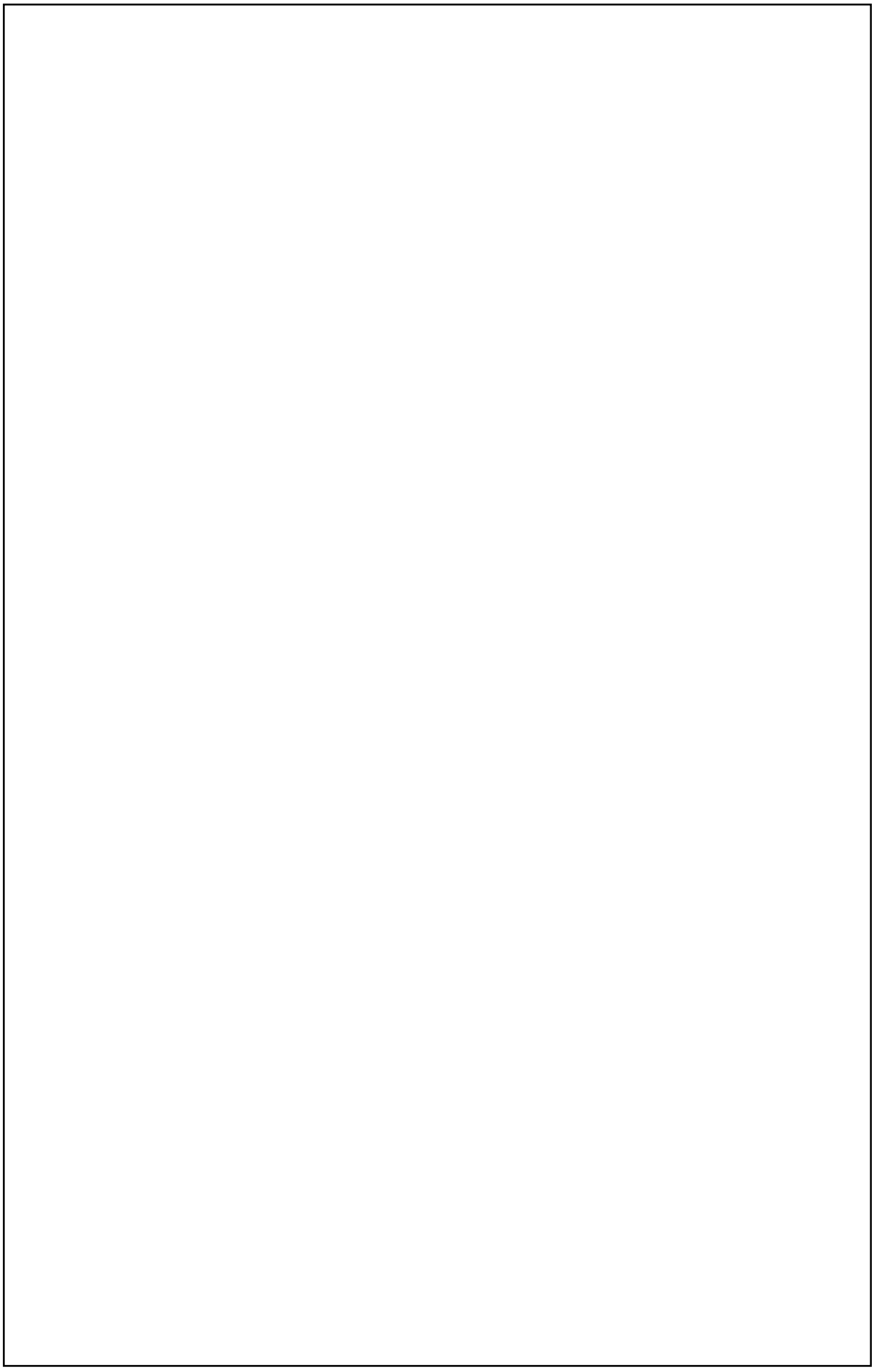
TECHNICAL MANUAL



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

UCC 08 Edition 5.20 *Rel. 18 November 2002*



UCC 08

UART Communication Card
8 serial lines

TECHNICAL MANUAL

Peripheral module for **ABACO® industrial BUS**; EUROCARD format 100x160mm; **SCC 2698**, **3.6864 MHz** oscillator, **baud rate** is software settable up to **38,4 KBaud**; **Three** software readable **8-pins Dip Switch** da 8 vie leggibile da software; **8** serial lines configurable as **RS 232**, **RS 422** or **RS 485**; **3** LEDs to visualize interrupt requests and board accesses; possibility to generate an **interrupt** connectable to /INT or to /NMI; addressing space on **BUS ABACO®** only 64 bytes; **two 8-pins Dip Switch** to set the I/O address; unque power supply **5Vdc**, **160 mA** (in basic configuration and no trasmission in progress).

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

UCC 08 Edition 5.20 *Rel. 18 November 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 **UCC 08** card release **111193** 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 under **grifo®** trademark on component and solder side).

BOARD FEATURES

UCC 08 (**U**art **C**ommunication **C**ard **8** serial lines) board is a powerful module designed for **ABACO®** industrial BUS serial communications featuring remarkable possibilities of interfacement to several external devices.
The board is based on the flexible SCC 2698, which can support asynchronous communication.
The **UCC 08** board comes in Europcard format 100x160mm, it manages separetely as much as 8 serial lines whose protocols and communication speeds are software settable; each of the 8 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 or RS 485. At last, interrupt activation and board acces are monitored through 3 LEDs.
The asynchronous communication can be made employing the following communication parameters:

- Baud rate selectable amongst 50 and 38.4 KBaud, fixed or user selectable
- Stop bit selectable amongst 1; 1.5 and 2 bits
- Word length selectable from 5 to 8 bits
- Reception buffer is 4 characters
- Parity can be even, odd or none

Five software readable 8-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).
Following configurations can be ordered:

- **UCC 08**	8 driver RS 232
- **UCC 08.8RS422**	8 driver RS 422
- **UCC 08.8RS485**	8 driver RS 485
- **UCC 08.4+4RS422**	4 drivers RS 232 e 4 driver RS 422
- **UCC 08.4+4RS485**	4 drivers RS 232 e 4 driver RS 485
- **UCC 08.8RS232**	8 driver RS 232
- **UCC 08.4RS422**	4 driver RS 422
- **UCC 08.4RS485**	4 driver RS 485

UCC 08 board takes 64 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 **ABACO® industrial BUS** addressing space through two of the above mentioned 8-pins Dip Switch.
The board is capable to generate an interrupt signal when specific software settable events occour.

- Peripheral module for **ABACO® industrial BUS**.
- EUROCARD format 100x160mm.
- **SCC 2698**, **3.6864 MHz** oscillator, **baud rate** is software settable up to **38,4 KBaud**.
- **Three** software readable **8-pins Dip Switch** da 8 vie leggibile da software.
- **8** serial lines configurable as **RS 232**, **RS 422** or **RS 485**.
- **3** LEDs to visualize interrupt requests and board accesses.
- Possibility to generate an **interrupt** connectable to /INT or to /NMI.
- Addressing space on **BUS ABACO®** only 64 bytes.
- **Two 8-pins Dip Switch** to set the I/O address.
- Unque power supply **5Vdc**, **160 mA** (in basic configuration and no trasmission in progress).

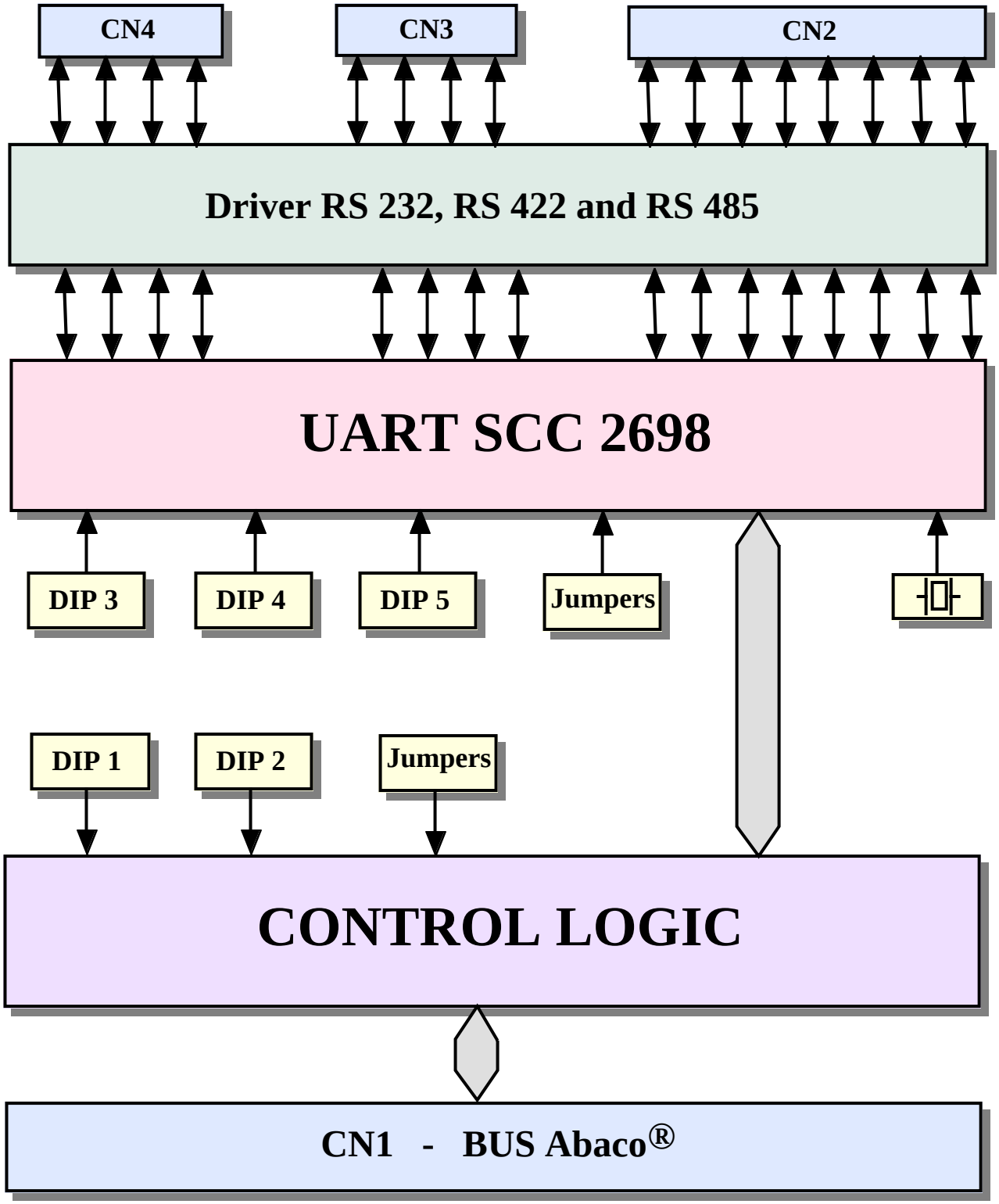


FIGURE 1: BLOC DIAGRAM

TECHNICAL FEATURES

GENERAL FEATURES

Board Resources:
- Interface for **ABACO® industrial BUS**.
- **Three 8-pins Dip switch** software readable.
- **Two 8-pins Dip switch** to set I/O address.
- 8 serial lines Full Duplex in RS 232, RS 422 or RS 485.

On board UART: SCC 2698

Clock frequency: 3.6864 MHz

BUS interface:
8 bits BUS for data.
8, 16 or 22 bits BUS for addresses.
256, 64K or 4M bytes of total addressing space.
64 bytes of I/O space occupied.

Serial communication:
Software configurable protocol:
- BAUD RATE: from 50 to 38.4 KBaud
- STOP BIT: 1, 1,5 and 2 bit
- Word lengtha: from 5 to 8 bit
- PARITY: None, Even, Odd

PHYSICAL FEATURES

Size (W x H x D): Eurocard standard format: 100 x 160 x 17 mm

Weight: 170 g (basic version)

Connectors:
CN1: 64 pin DIN 41612 A+C type C
CN2: Low profile, 40 pins, male, 90 degreeses
CN3: Low profile, 20 pins, male, 90 degreeses
CN4: Low profile, 20 pins, male

Temperature range: from 0 to 70 Centigrad degreeses

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

ELECTRIC FEATURES

Power Supply: +5 Vdc

Consumption on 5 Vdc: 160 mA (default RS 232 configuration)
250 mA (maximum configuration))

RS 422, RS 485 line termination: Line termination resistance= 120 Ω

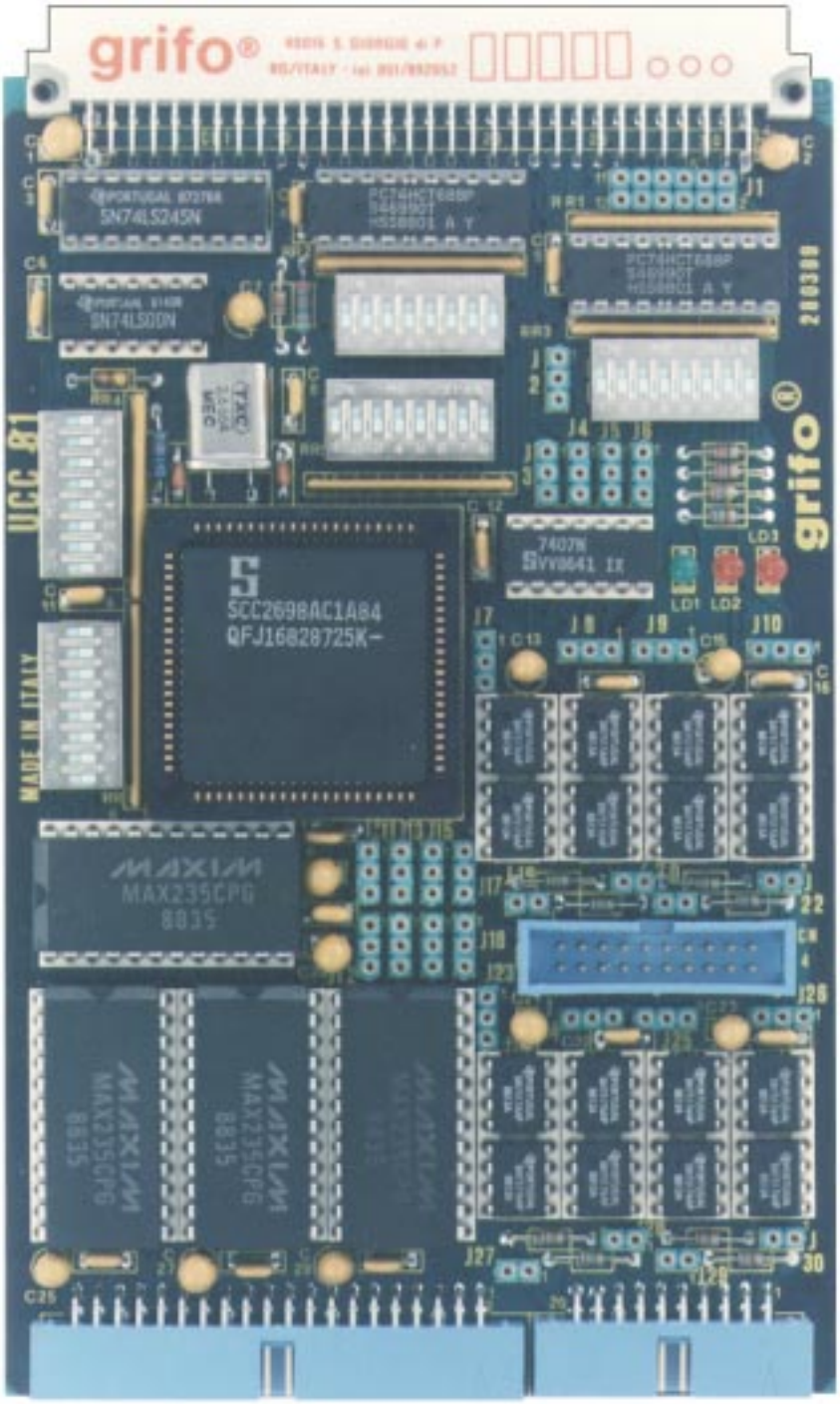


FIGURE 2: CARD PHOTO

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

UCC 08 board is provided with three LEDs in order to signal to the user some internal status conditions, as described in the following table:

LED	COLOUR	PURPOSE
LD1	Red	When on, indicates an access to in read or write to one of the serial lines in progress.
LD2	Red	When ON, indicates that **UCC 08** has sent a maskable interrupt on **BUS ABACO®** (line /INT).
LD3	Red	When ON, indicates that **UCC 08** has sent a non maskable interrupt on **BUS ABACO®** (line /NMI).

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 **UCC 08** module has 4 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 4). For further information about serial connections, please refer to the successive figures, which show the kind of board connection to perform.

DIP SWITCH

UCC 08 board is provided with five 8 pins dip switch, typically used for system configuration, three of these dip switches are software readable by the User (DIP3, DIP4 and DIP5), while the remaining two are used for mapping address selection (DIP1 and DIP2). The most frequent applications are: working condition selection or on board firmware parameters setting. For further information please refer to the paragraphs “I/O MAPPING” and “PERIPHERAL DEVICES SOFTWARE DESCRIPTION”, while to easily locate the dip switch please refer to figure 4.

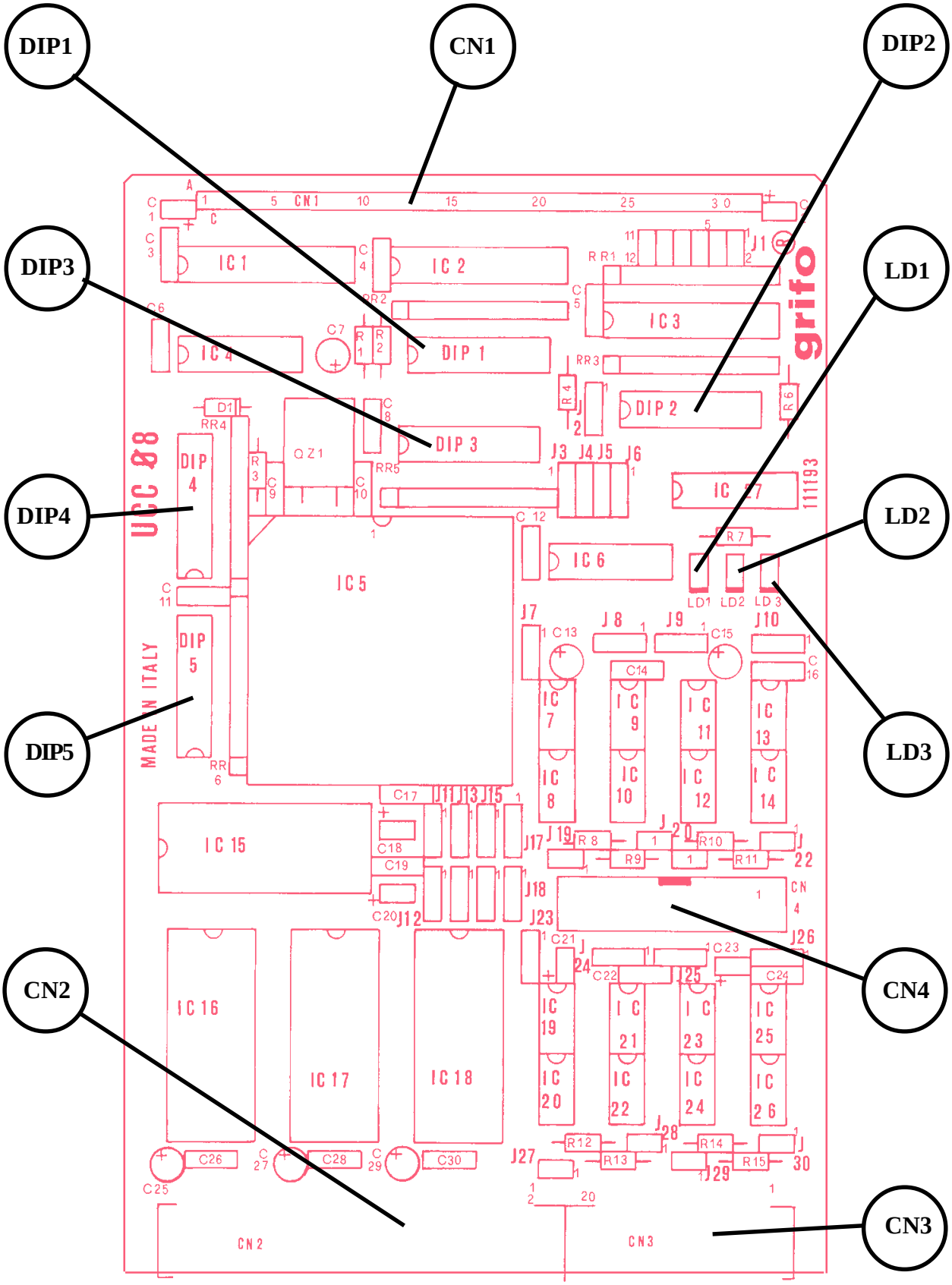


FIGURE 4: CONNECTORS, LEDS, DIP SWICTH, ETC LOCATION.

CN1 - CONNECTOR FOR BUS ABACO®

The connector for **ABACO® industrial BUS**, called CN1 on board,is a DIN 41612, male, a 90 °, type C, A+C.
Here follows the pin-out of the connector installed on **UCC 08**, in addition there is the standard 8 bits and 16 bits**ABACO®** BUS pin-out.
Please remark that all the signals here described are TTL, except for the power supplies.

A 16 bit BUS	A 8 bit BUS	A UCC 08	PIN	C UCC 08	C 8 bit BUS	C 16 bit BUS
GND	GND	GND	1	GND	GND	GND
+5 Vdc	+5 Vdc	+5 Vdc	2	+5 Vdc	+5 Vdc	+5 Vdc
D0	D0	D0	3	N.C.		D8
D1	D1	D1	4	N.C.		D9
D2	D2	D2	5	N.C.		D10
D3	D3	D3	6	/INT	/INT	/INT
D4	D4	D4	7	/NMI	/NMI	/NMI
D5	D5	D5	8	N.C.	/HALT	D11
D6	D6	D6	9	N.C.	/MREQ	/MREQ
D7	D7	D7	10	/IORQ	/IORQ	/IORQ
A0	A0	A0	11	/RD	/RD	/RDLDS
A1	A1	A1	12	/WR	/WR	/WRLDS
A2	A2	A2	13	N.C.	/BUSAK	D12
A3	A3	A3	14	N.C.	/WAIT	/WAIT
A4	A4	A4	15	N.C.	/BUSRQ	D13
A5	A5	A5	16	/RESET	/RESET	/RESET
A6	A6	A6	17	/M1	/M1	/IACK
A7	A7	A7	18	N.C.	/RFSH	D14
A8	A8	A8	19	N.C.	/MEMDIS	/MEMDIS
A9	A9	A9	20	N.C.	VDUSEL	A22
A10	A10	A10	21	N.C.	/IEI	D15
A11	A11	A11	22	N.C.		
A12	A12	A12	23	N.C.	CLK	CLK
A13	A13	A13	24	N.C.		/RDUDS
A14	A14	A14	25	N.C.		/WRUDS
A15	A15	A15	26	A21		A21
A16		A16	27	A20		A20
A17		A17	28	A19		A19
A18		A18	29	N.C.	/R.T.	/R.T.
+12 Vdc	+12 Vdc	N.C.	30	N.C.	-12 Vdc	-12 Vdc
+5 Vdc	+5 Vdc	+5 Vdc	31	+5 Vdc	+5 Vdc	+5 Vdc
GND	GND	GND	32	GND	GND	GND

FIGURE 5: CN1 - BUS ABACO® CONNECTOR

Signals description:

8 bits CPU

A0-A15	=	O	- Address BUS
D0-D7	=	I/O	- Data BUS
/INT	=	I	- Interrupt request
/NMI	=	I	- Non Maskable Interrupt
/HALT	=	O	- Halt state
/MREQ	=	O	- Memory Request
/IORQ	=	O	- Input Output Request
/RD	=	O	- Read cycle status
/WR	=	O	- Write cycle status
/BUSAK	=	O	- BUS Acknowledge
/WAIT	=	I	- Wait
/BUSRQ	=	I	- BUS Request
/RESET	=	O	- Reset
/M1	=	O	- Machine cycle one
/RFSH	=	O	- Refresh for dynamic RAM
/MEMDIS	=	I	- Memory Display
VDUSEL	=	O	- VDU Selection
/IEI	=	I	- Interrupt Enable Input
CLK	=	O	- System clock
R.B.	=	I	- Reset button
+5 Vdc	=	I	- Power supply at +5 Vdc
+12 Vdc	=	I	- Power supply at +12 Vdc
-12 Vdc	=	I	- Power supply at -12 Vdc
GND	=		- Ground signal

16 bits CPU

A16-A22	=	O	- Address BUS
D8-D15	=	I/O	- Data BUS
/RD UDS	=	O	- Read Upper Data Strobe
/WR UDS	=	O	- Write Upper Data Strobe
/IACK	=	O	- Interrupt Acknowledge
/RD LDS	=	O	- Read Lower Data Strobe
/WR LDS	=	O	- Write Lower Data Strobe

NOTE
Directionality indications as above stated are referred to a master (**GPC®**) board and have been kept untouched to avoid ambiguity in case of multi-boards systems.

CN2 - CONNECTOR FOR RS 232 COMMUNICATION

CN2 is a 40 pins, 90 degreeses, low profile connector for RS 232 communication. 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.

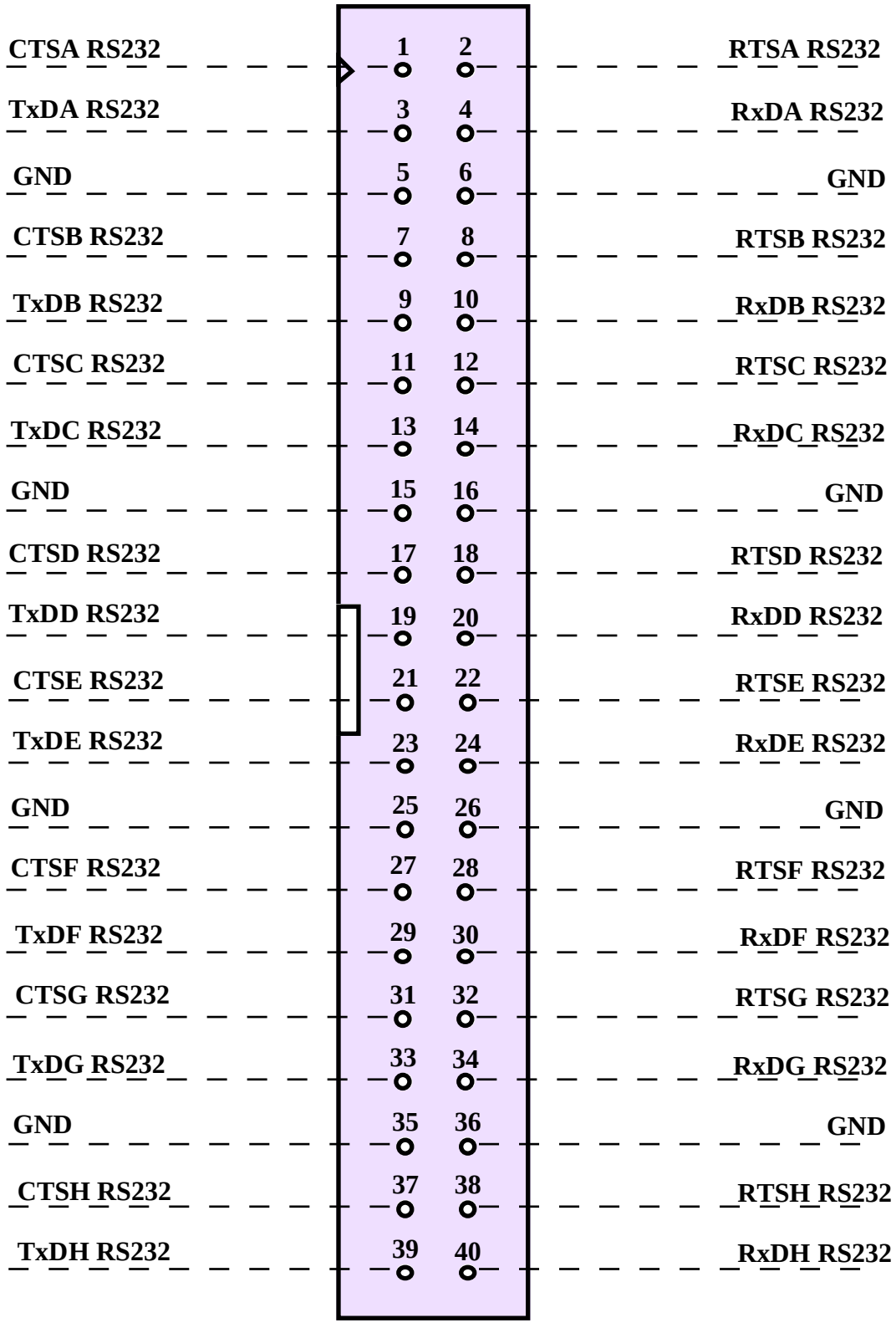


FIGURE 6: CN2 - CONNECTOR FOR RS 232 COMMUNICATION

Signals description:

RxDy RS 232 = I - RS 232 serial lines A÷H Receive Data.
TxDy RS 232 = O - RS 232 serial lines A÷H Transmit Data.
CTSy RS 232 = I - RS 232 serial lines A÷H Clear To Send.
RTSy RS 232 = O - RS 232 serial lines A÷H Request To Send.
GND = - Ground.

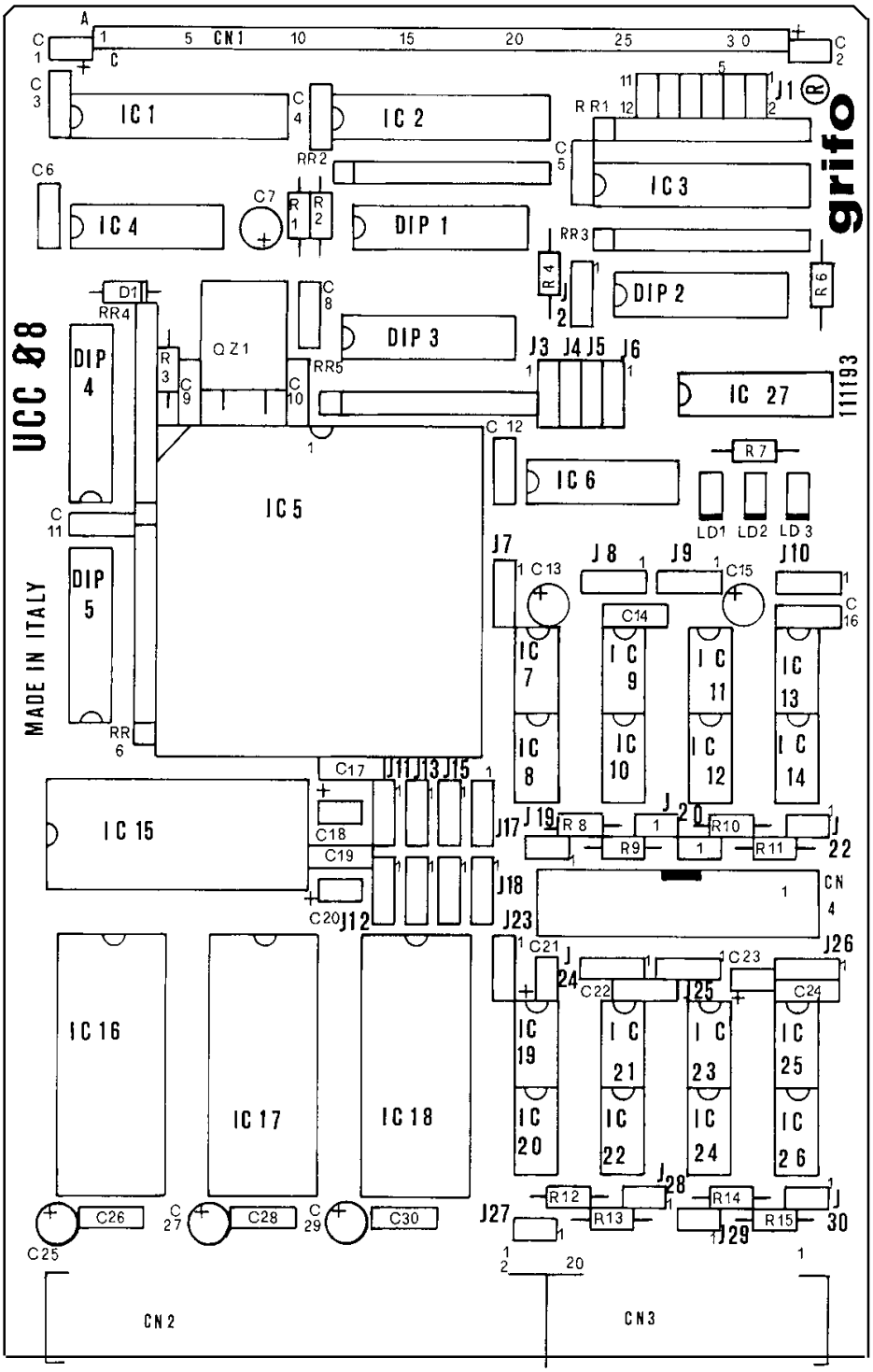


FIGURE 7: COMPONENTS MAP

CN3- CONNECTOR FOR RS 422/RS 485 OF SERIAL LINES A÷D

CN3 is a 20 pins, 90 degreeses, low profile connector for RS 422/RS 485 communication of serial lines from A to D. 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.

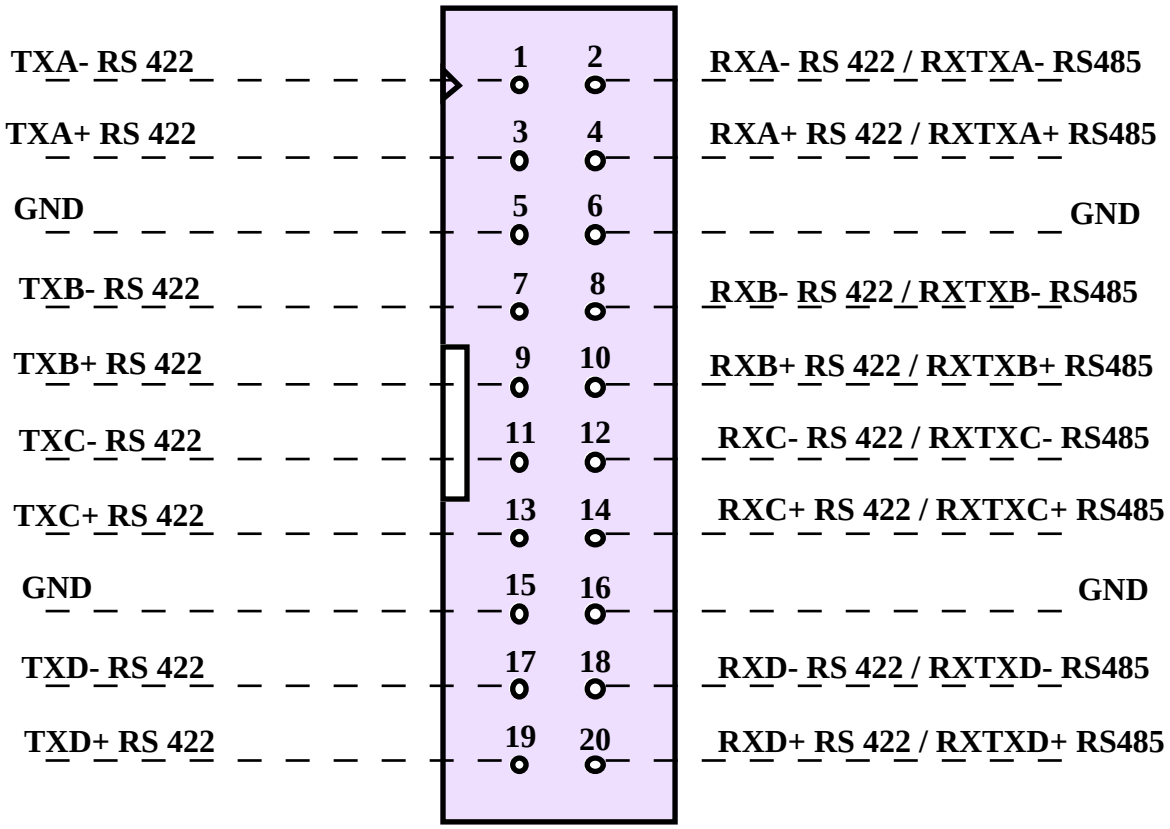


FIGURE 8: CN3 - CONNECTOR FOR RS 422/RS 485 OF SERIAL LINES A÷D

Signal description:

RXy- RS 422	= I - RS 422 of serial line A÷D Receive Data Negative.
RXy+ RS 422	= I - RS 422 of serial line A÷D Receive Data Positive.
TXy- RS 422	= O - RS 422 of serial line A÷D Transmit Data Negative.
TXy+ RS 422	= O - RS 422 of serial line A÷D Transmit Data Positive.
RXTXy- RS 485	=I/O- RS 422 of serial line A÷D Receive Transmit Data Negative.
RXTXy+ RS 485	=I/O- RS 422 of serial line A÷D Receive Transmit Data Positive.
GND	= - Ground.

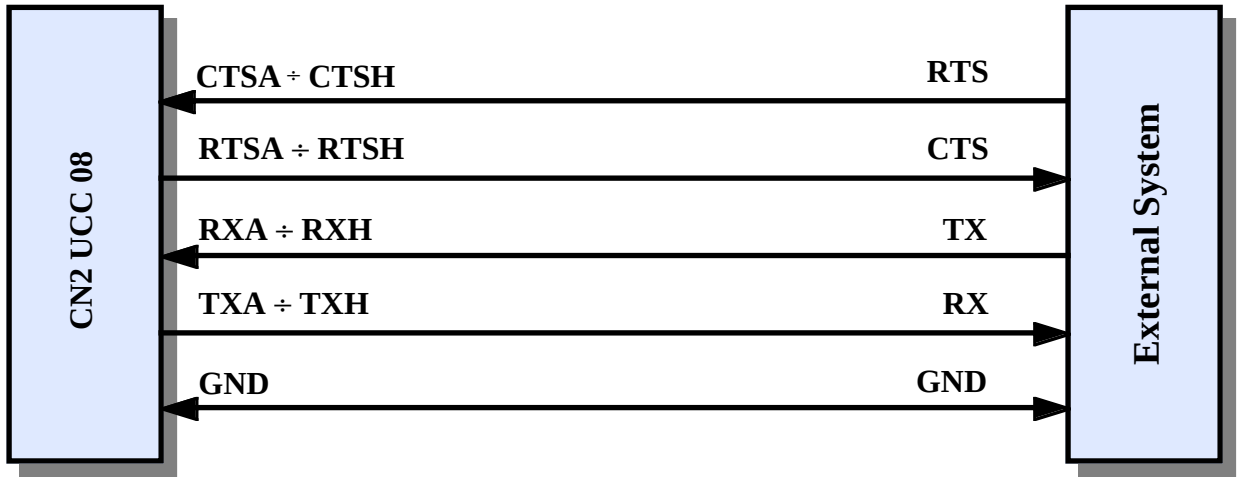


FIGURE 9: RS 232 POINT TO POINT CONNECTION EXAMPLE

FIGURE 10: RS 485 POINT TO POINT CONNECTION EXAMPLE

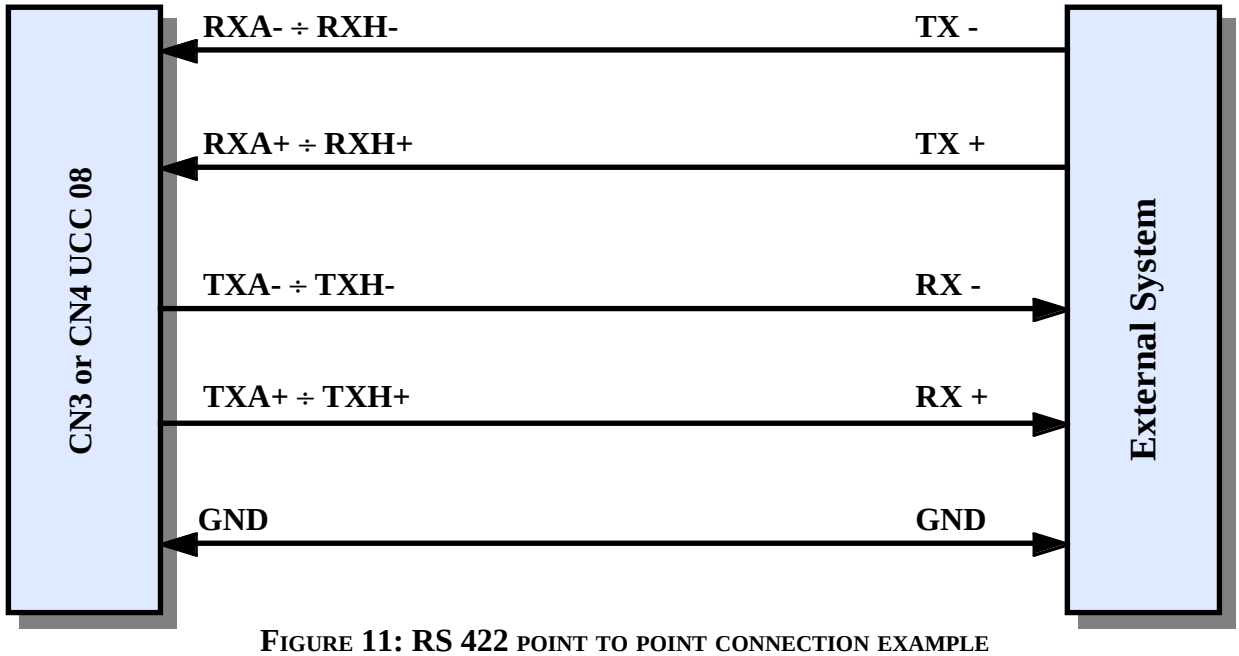


FIGURE 11: RS 422 POINT TO POINT CONNECTION EXAMPLE

CN4 - CONNECTOR FOR RS 422/RS 485 OF SERIAL LINES E÷H

CN4 is a 20 pins, 90 degreeses, low profile connector for RS 422/RS 485 communication of serial lines from A to D. 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.

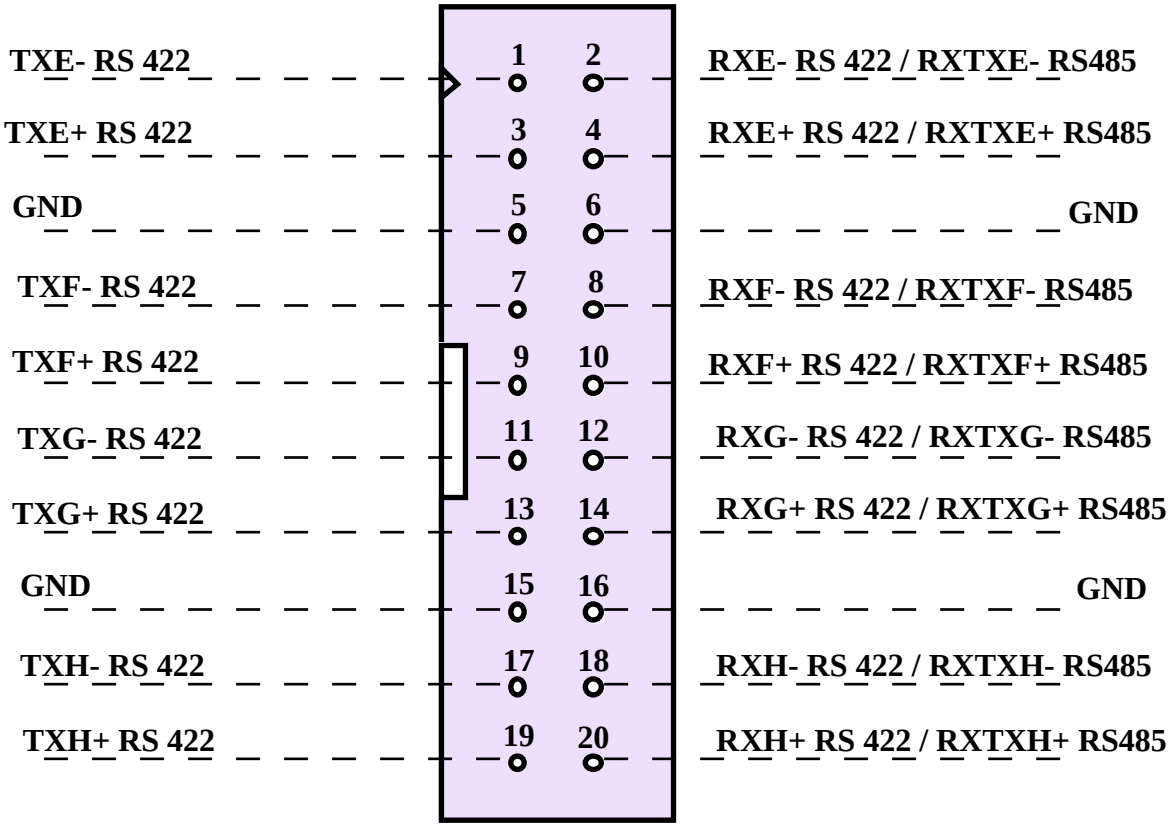


FIGURE 12: CN4 - CONNECTOR FOR RS 422/RS 485 OF SERIAL LINES E÷H

Signal description:

RXy- RS 422	= I - RS 422 of serial line E÷H Receive Data Negative.
RXy+ RS 422	= I - RS 422 of serial line E÷H Receive Data Positive.
TXy- RS 422	= O - RS 422 of serial line E÷H Transmit Data Negative.
TXy+ RS 422	= O - RS 422 of serial line E÷H Transmit Data Positive.
RXTXy- RS 485	=I/O- RS 422 of serial line E÷H Receive Transmit Data Negative.
RXTXy+ RS 485	=I/O- RS 422 of serial line E÷H Receive Transmit Data Positive.
GND	= - Ground.

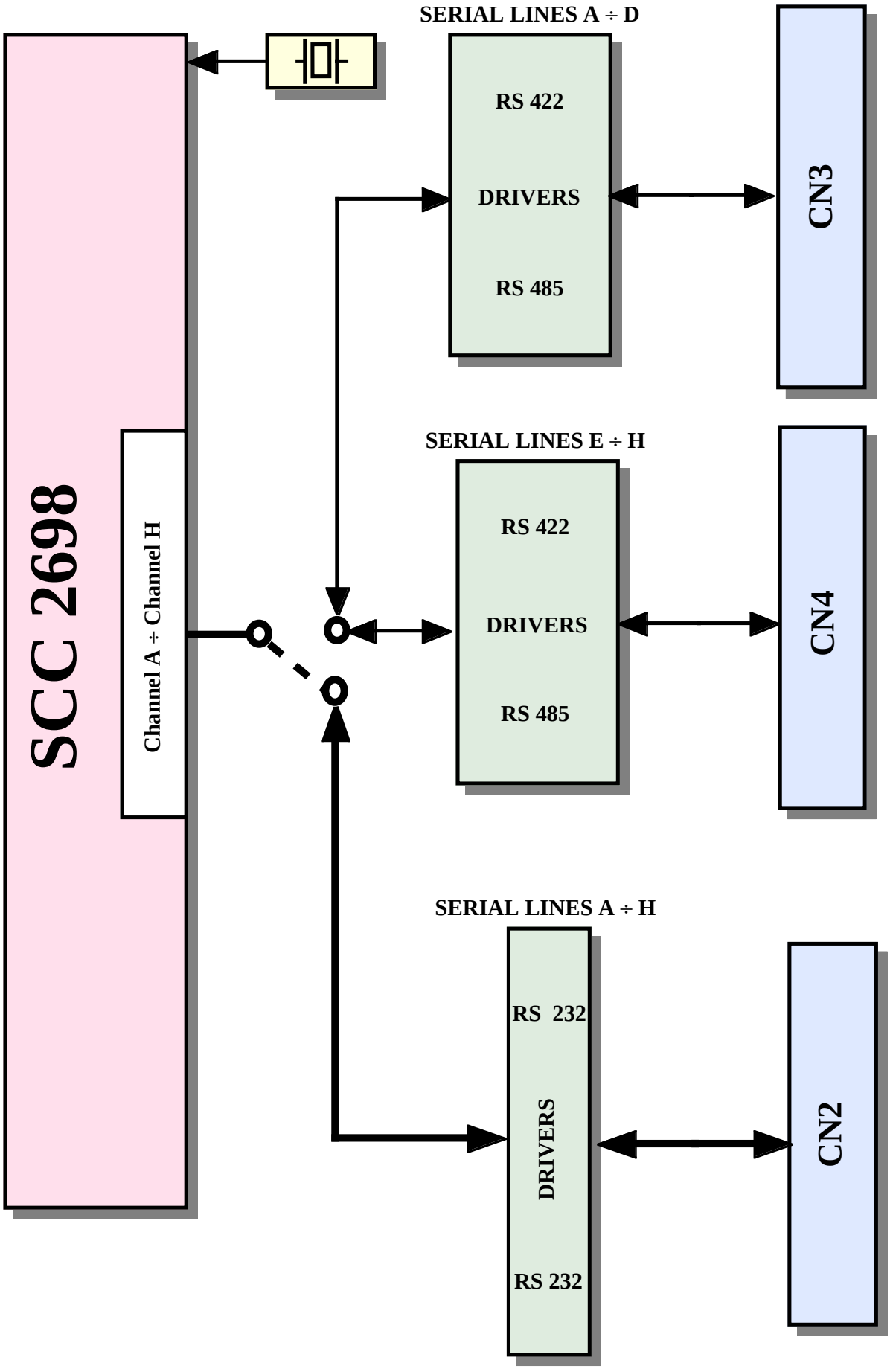


FIGURE 14: SERIAL COMMUNICATION DIAGRAM

JUMPERS

On **UCC 08** there are 30 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	12	Determines the address in case of enhanced extended addressing and connects signal /M1 coming from BUS **ABACO®** to the card.
J2	3	Selects the addressing mode of the board.
J3	3	Selects connection of signal /INTR for serial lines A and B to BUS **ABACO®**.
J4	3	Selects connection of signal /INTR for serial lines C and D to BUS **ABACO®**.
J5	3	Selects connection of signal /INTR for serial lines E and F to BUS **ABACO®**.
J6	3	Selects connection of signal /INTR for serial lines G and H to BUS **ABACO®**.
J7	3	Selects the RS 422 or RS 485 working mode for serial line H.
J8	3	Selects the RS 422 or RS 485 working mode for serial line G.
J9	3	Selects the RS 422 or RS 485 working mode for serial line F.
J10	3	Selects the RS 422 or RS 485 working mode for serial line E.
J11	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line H.
J12	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line G.
J13	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line F.
J14	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line E.
J15	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line D.

FIGURE 15: JUMPERS SUMMARIZING TABLE - PART 1

Tables in the following pages describe all the right connections of **UCC 08** jumpers with their relative functions. To recognize these valid connections, please refer to the board printed diagram (serigraph) or to figure 7 of this manual, where the pins numeration is listed, while for recognizing jumpers location, please refer to figures 20 and 22.
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.

JUMPERS	N. PINS	PURPOSE
J16	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line C.
J17	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line B.
J18	3	Determines the connection of reception signal in RS 232 or RS 422/RS 485 for serial line A.
J19	2	Connects RS 422/RS 485 termination for serial line H.
J20	2	Connects RS 422/RS 485 termination for serial line G.
J21	2	Connects RS 422/RS 485 termination for serial line F.
J22	2	Connects RS 422/RS 485 termination for serial line E.
J23	3	Selects the RS 422 or RS 485 working mode for serial line D.
J24	3	Selects the RS 422 or RS 485 working mode for serial line C.
J25	3	Selects the RS 422 or RS 485 working mode for serial line B.
J26	3	Selects the RS 422 or RS 485 working mode for serial line A.
J27	2	Connects RS 422/RS 485 termination for serial line D.
J28	2	Connects RS 422/RS 485 termination for serial line C.
J29	2	Connects RS 422/RS 485 termination for serial line B.
J30	2	Connects RS 422/RS 485 termination for serial line A.

FIGURE 16: JUMPERS SUMMARIZING TABLE - PART 2

2 PINS JUMPERS

JUMPER	CONNECTION	PURPOSE	DEF.
J19	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port H.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port H.	
J20	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port G.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port G.	
J21	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port F.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port F.	
J22	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port E.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port E.	
J27	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port D.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port D.	
J28	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port C.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port C.	
J29	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port B.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port B.	
J30	not connected	Does not connect the termination circuitery to RS 485 signals or RS 422 reception signal of serial port A.	*
	connected	Connects the termination circuitery to RS 485 signals or RS 422 reception signal of serial port A.	

FIGURE 17: 2 PINS JUMPERS TABLE

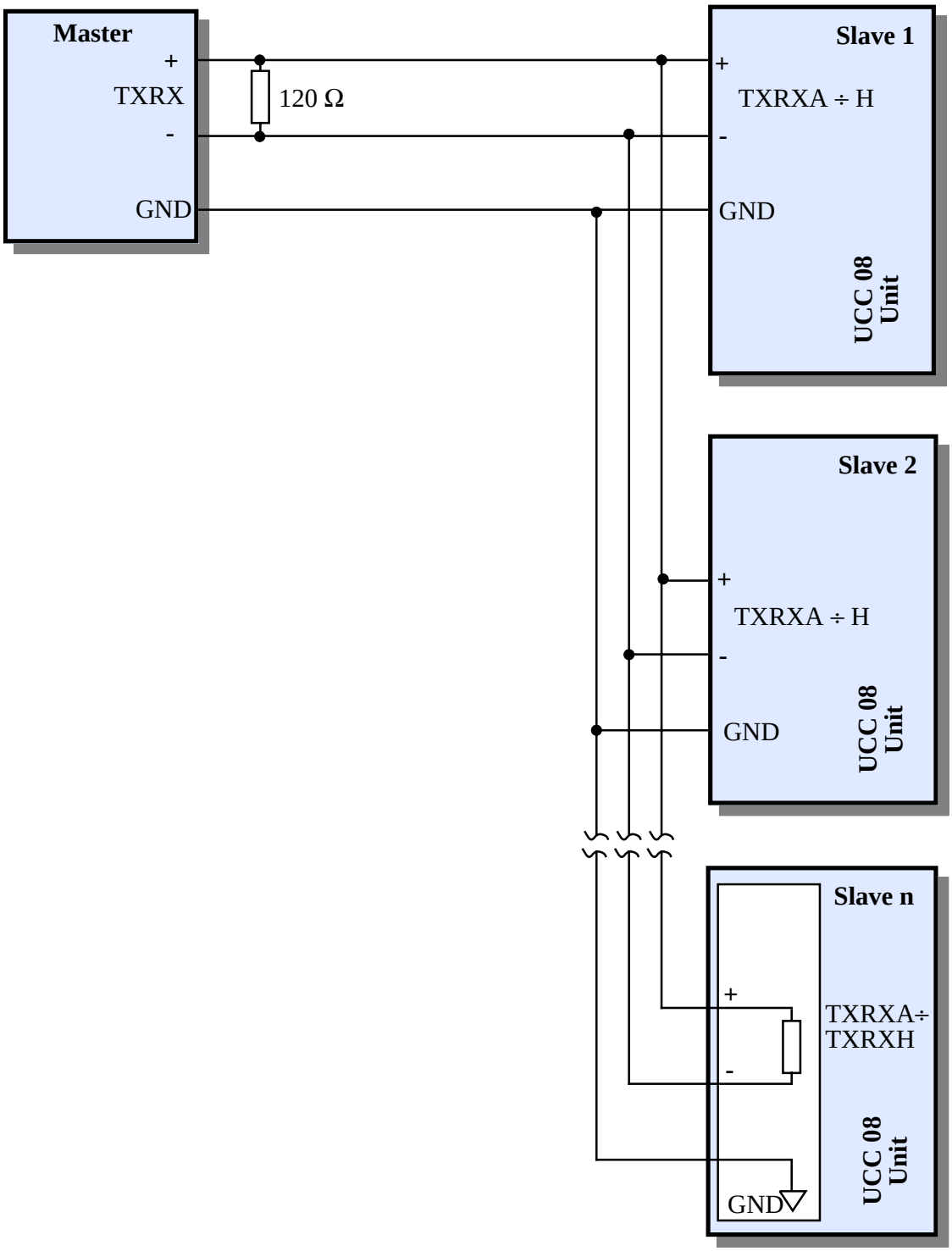


FIGURE 18: RS 485 NETWORK CONNECTION EXAMPLE

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 **UCC 08** 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.

3 PINS JUMPERS

JUMPERS	CONNECTION	PURPOSE	DEF.
J2	position 1-2	Enables normal 8 bit addressing mode. **J1 must be disconnected and switches from 1 to 5 of DIP2 must be OFF.**	
	position 2-3	Enables extended 16 bit addressing mode **(J1 must be disconnected and switches from 1 to 5 of DIP2 must be OFF)** or extended enhances 22 bit addressing mode **(in this case switches from 1 to 5 make bits A16÷A20 of the address).**	
J3	position 1-2	Connects interrupt of serial lines A and B to signal /INT of **BUS ABACO®**.	
	position 2-3	Connects interrupt of serial lines A and B to signal /NMI of **BUS ABACO®**.	
	not connected	Serial lines A and B cannot generate any kind of interrupt on **BUS ABACO®**.	*
J4	position 1-2	Connects interrupt of serial lines C and D to signal /INT of **BUS ABACO®**.	
	position 2-3	Connects interrupt of serial lines C and D to signal /NMI of **BUS ABACO®**.	
	not connected	Serial lines C and D cannot generate any kind of interrupt on **BUS ABACO®**.	*
J5	position 1-2	Connects interrupt of serial lines E and F to signal /INT of **BUS ABACO®**.	
	position 2-3	Connects interrupt of serial lines E and F to signal /NMI of **BUS ABACO®**.	
	not connected	Serial lines E and F cannot generate any kind of interrupt on **BUS ABACO®**.	*
J6	position 1-2	Connects interrupt of serial lines G and H to signal /INT of **BUS ABACO®**.	
	position 2-3	Connects interrupt of serial lines G and H to signal /NMI of **BUS ABACO®**.	
	not connected	Serial lines G and H cannot generate any kind of interrupt on **BUS ABACO®**.	*
J7	position 1-2	Selects RS 422 serial communication on port H (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port H (2 wires half duplex).	
J8	position 1-2	Selects RS 422 serial communication on port G (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port G (2 wires half duplex).	

FIGURE 19: 3 PINS JUMPERS TABLE - PART 1

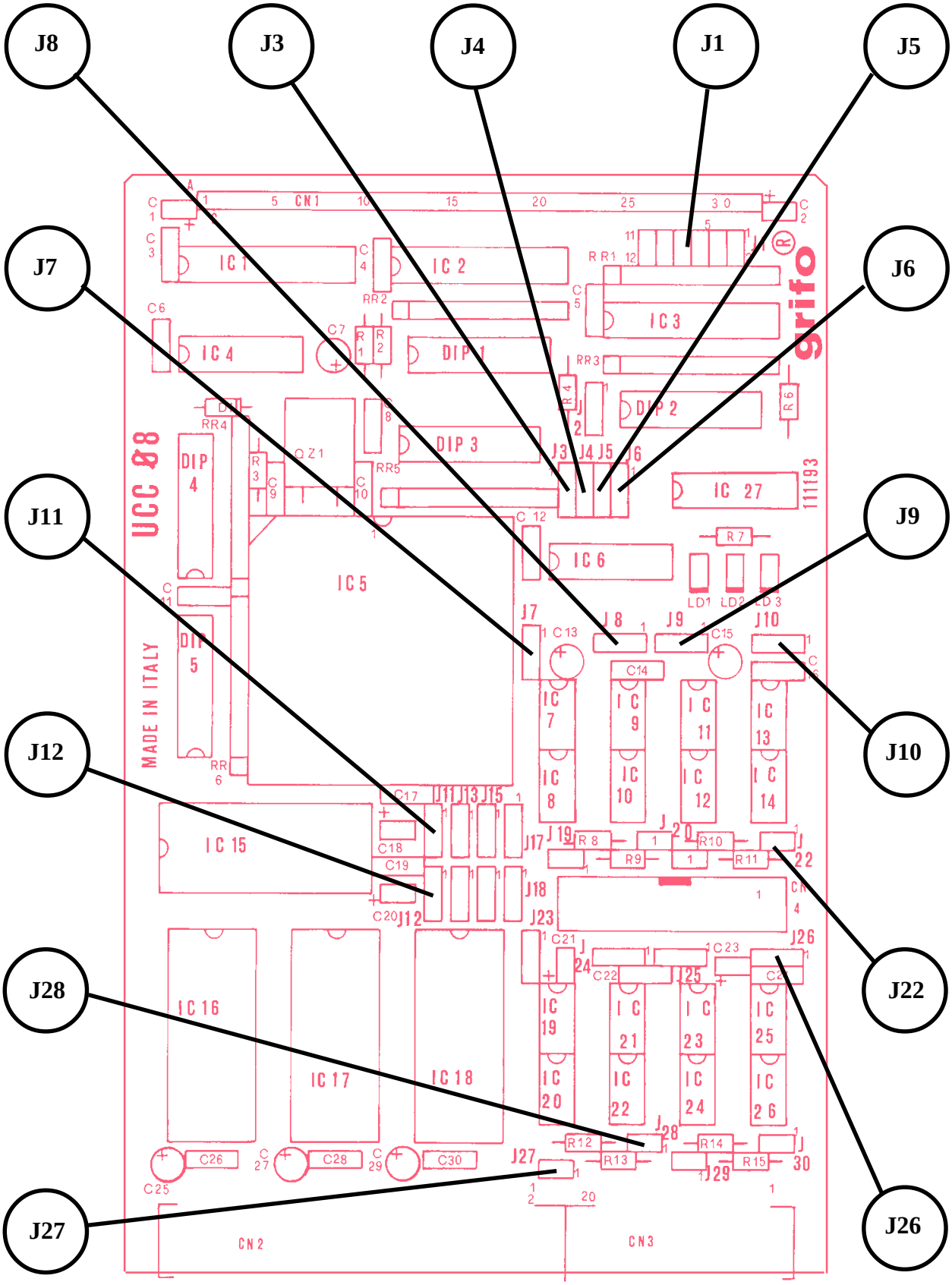


FIGURE 20: JUMPERS LOCATION TABLE - PART 1

JUMPERS	CONNECTION	PURPOSE	DEF.
J9	position 1-2	Selects RS 422 serial communication on port F (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port F (2 wires half duplex).	
J10	position 1-2	Selects RS 422 serial communication on port E (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port E (2 wires half duplex).	
J11	position 1-2	Connectes reception signal of port H to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port H to RS 422 or RS 485 serial communication protocol.	
J12	position 1-2	Connectes reception signal of port G to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port G to RS 422 or RS 485 serial communication protocol.	
J13	position 1-2	Connectes reception signal of port F to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port F to RS 422 or RS 485 serial communication protocol.	
J14	position 1-2	Connectes reception signal of port E to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port E to RS 422 or RS 485 serial communication protocol.	
J15	position 1-2	Connectes reception signal of port D to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port D to RS 422 or RS 485 serial communication protocol.	
J16	position 1-2	Connectes reception signal of port C to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port C to RS 422 or RS 485 serial communication protocol.	
J17	position 1-2	Connectes reception signal of port B to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port B to RS 422 or RS 485 serial communication protocol.	
J18	position 1-2	Connectes reception signal of port A to RS 232 serial communication protocol.	*
	position 2-3	Connectes reception signal of port A to RS 422 or RS 485 serial communication protocol.	

FIGURE 21: 3 PINS JUMPERS TABLE - PART 2

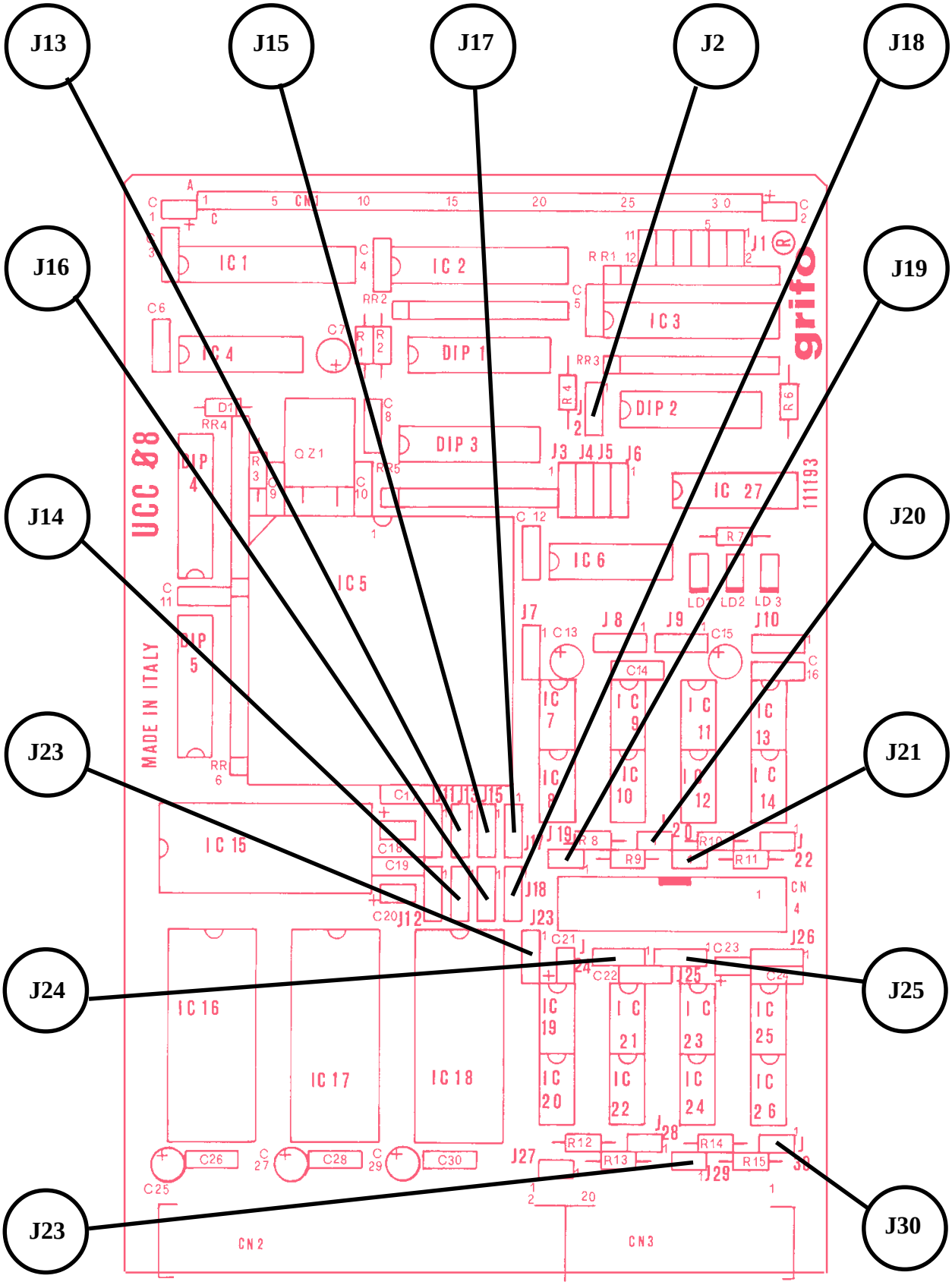


FIGURE 22: JUMPERS LOCATION TABLE - PART 2

JUMPERS	CONNECTION	PURPOSE	DEF.
J23	position 1-2	Selects RS 422 serial communication on port D (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port D (2 wires half duplex).	
J24	position 1-2	Selects RS 422 serial communication on port C (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port C (2 wires half duplex).	
J25	position 1-2	Selects RS 422 serial communication on port B (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port B (2 wires half duplex).	
J26	position 1-2	Selects RS 422 serial communication on port A (4 wires full duplex or half duplex).	*
	position 2-3	Selects RS 485 serial communication on port A (2 wires half duplex).	

FIGURE 23: 3 PINS JUMPERS TABLE - PART 3

6 PINS JUMPER

JUMPERS	CONNECTION	PURPOSE	DEF.
J1	position 1-2	If connected, signal /M1 is used as MSB in the card address for enhanced extended mode.	
	position 3-4	If connected, signal A20 is used as bit 20 in the card address for enhanced extended mode.	
	position 5-6	If connected, signal A19 is used as bit 19 in the card address for enhanced extended mode.	
	position 7-8	If connected, signal A18 is used as bit 18 in the card address for enhanced extended mode.	
	position 9-10	If connected, signal A17 is used as bit 17 in the card address for enhanced extended mode.	
	position 11-12	If connected, signal A16 is used as bit 16 in the card address for enhanced extended mode.	
	no jumper connected	No signal is used in the card address for enhanced extended mode.	*

FIGURE 24: 6 PINS JUMPERS TABLE

I/O CONNECTION

To prevent possible connecting problems between **UCC 08** 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 and RS 485 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".

SERIAL COMMUNICATION SELECTION

Please remember that if not differently specified during the order phase, the card is delivered in its default configuration with eight RS 232 serial line.
The serial lines available can be buffered in RS 232, RS 422 or RS 485, 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 desired bits per character, parity, stop bits, etc. at standard or non standard baud rates.
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 the configuration, following the below description (jumpers not mentioned in the below description have no influence on communication):

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

J18	=	position 1-2	IC18	= indifferent
J30	=	(*1)	IC25	= MAX 483 or SN75176
J26	=	position 1-2	IC26	= MAX 483 or SN75176

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

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

In point to point connections, signal MPOA 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 A CONFIGURED IN RS 232 (default)

J18	=	position 2-3	IC18	= driver MAX 235
J30	=	indifferent	IC25	= indifferent
J26	=	indifferent	IC26	= indifferent

- SERIAL LINE A CONFIGURED IN RS 485 (options RS 485)

J18	=	position 1-2	IC18	=	indifferent
J30	=	(*1)	IC25	=	MAX 483 or SN75176
J26	=	position 2-3	IC26	=	no component

In this modality the signals to use are pins 2 and 4 of connector CN3, that become transmission or reception lines according to the status of signal MPOA, managed by software, as follows:

MPOA = low level	= logic state 0	->	transmitter enabled
MPOA = 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.

- SERIAL LINE B CONFIGURED IN RS 422 (options RS 422)

J17	=	position 1-2	IC18	=	indifferent
J29	=	(*1)	IC23	=	MAX 483 or SN75176
J25	=	position 1-2	IC24	=	MAX 483 or SN75176

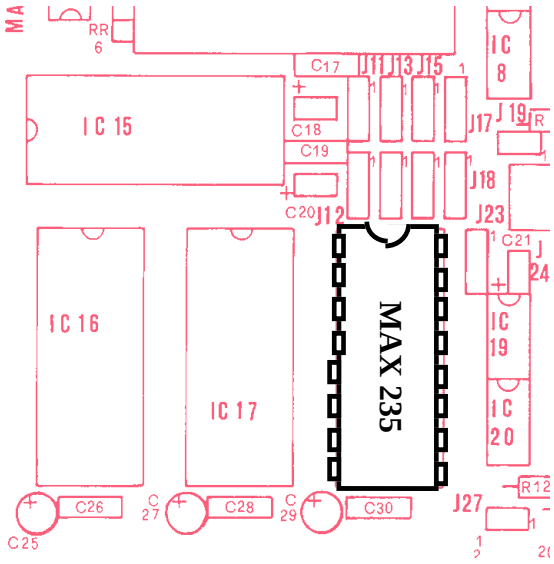
Status of signal MPOB, which is software managed, allows to enable or disable the transmitter as follows:

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

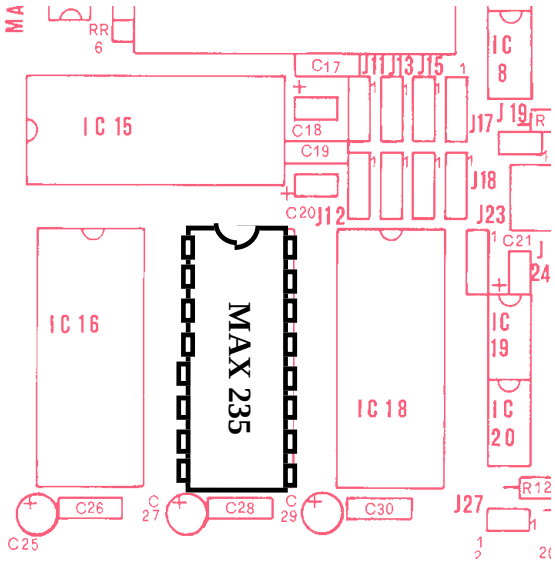
In point to point connections, signal MPOB can be always kept low (trasnmitter always enabled), while in multi point connections transmitter must be enabled only when a transmission is requested.

- SERIAL LINE B CONFIGURED IN RS 232 (default)

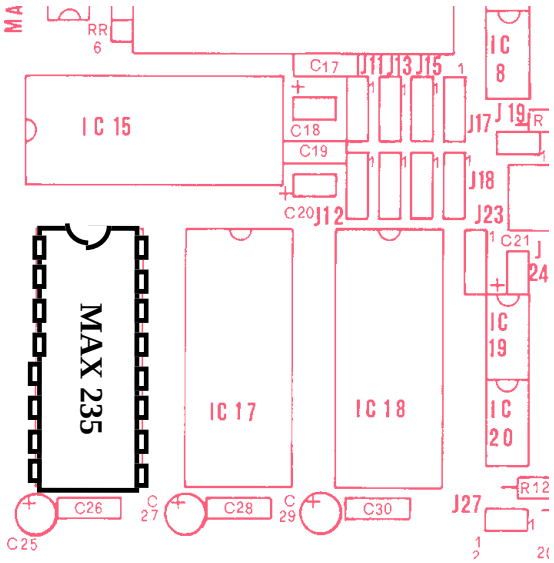
J17	=	position 2-3	IC18	=	driver MAX 235
J29	=	indifferent	IC23	=	indifferent
J25	=	indifferent	IC24	=	indifferent



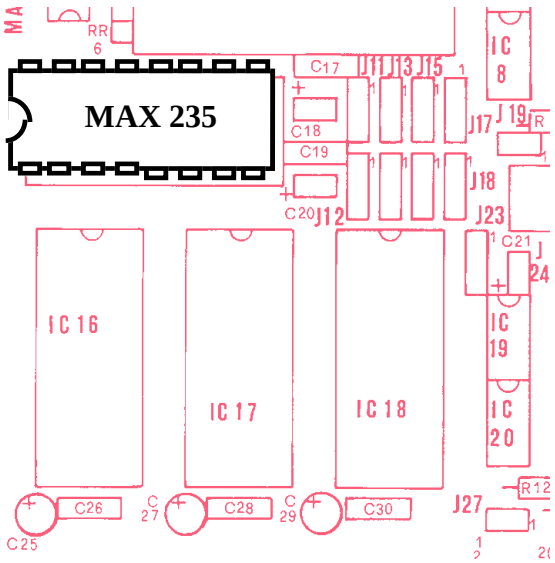
Serial lines A and B in RS 232



Serial lines C and D in RS 232



Serial lines E and F in RS 232



Serial lines G and H in RS 232

FIGURE 25: DRIVER FOR RS 232 SERIAL COMMUNICATION LOCATION

- SERIAL LINE B CONFIGURED IN RS 485 (options RS 485)

J17	=	position 1-2	IC18	= indifferent
J29	=	(*1)	IC23	= MAX 483 or SN75176
J25	=	position 2-3	IC24	= no component

In this modality the signals to use are pins 8 and 10 of connector CN3, that become transmission or reception lines according to the status of signal MPOB, managed by software, as follows:

MPOB = low level	= logic state 0	->	transmitter enabled
MPOB = 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.

- SERIAL LINE C CONFIGURED IN RS 422 (options RS 422)

J16	=	position 1-2	IC17	= indifferent
J28	=	(*1)	IC21	= MAX 483 or SN75176
J24	=	position 1-2	IC22	= MAX 483 or SN75176

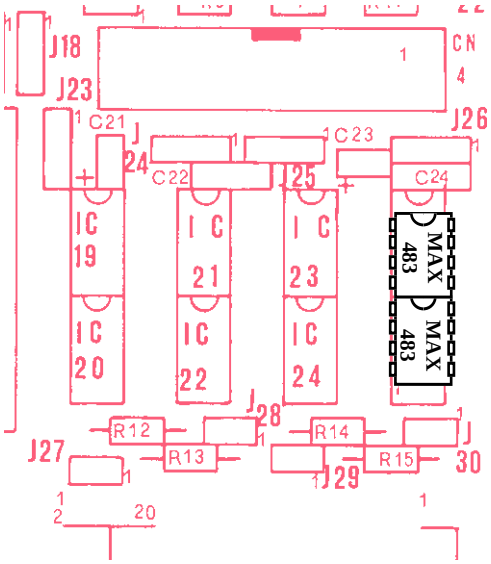
Status of signal MPOC, which is software managed, allows to enable or disable the transmitter as follows:

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

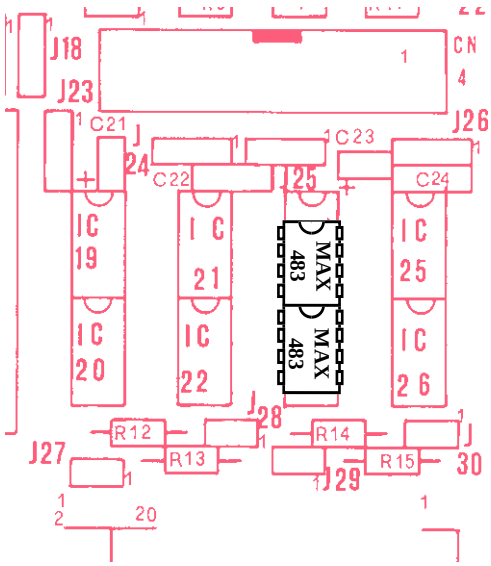
In point to point connections, signal MPOC can be always kept low (trasnmitter always enabled), while in multi point connections transmitter must be enabled only when a transmission is requested.

- SERIAL LINE C CONFIGURED IN RS 232 (default)

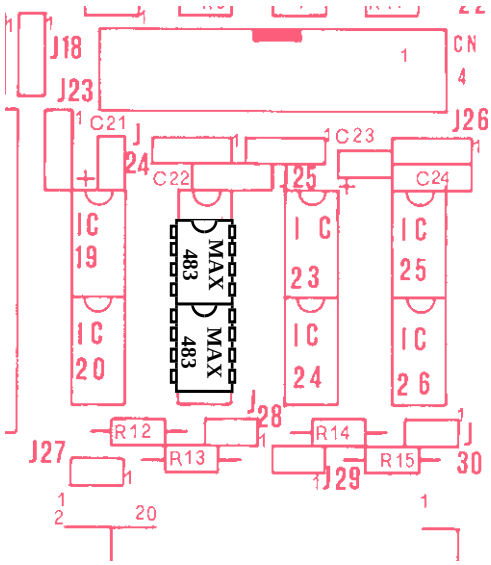
J16	=	position 2-3	IC17	= driver MAX 235
J28	=	indifferent	IC21	= indifferent
J24	=	indifferent	IC22	= indifferent



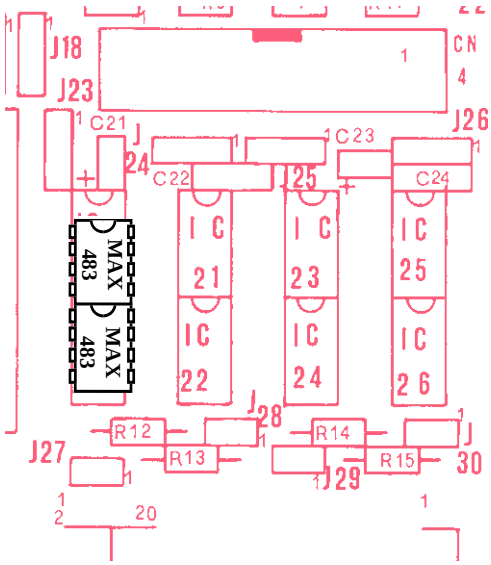
Serial line A in RS 422



Serial line B in RS 422



Serial line C in RS 422



Serial line D in RS 422

FIGURE 26: RS 422 DRIVERS FOR SERIAL LINES A÷D LOCATION

- SERIAL LINE C CONFIGURED IN RS 485 (options RS 485)

J16	=	position 1-2	IC17	= indifferent
J28	=	(*1)	IC21	= MAX 483 or SN75176
J24	=	position 2-3	IC22	= no component

In this modality the signals to use are pins 12 and 14 of connector CN3, that become transmission or reception lines according to the status of signal MPOC, managed by software, as follows:

MPOC = low level	= logic state 0	->	transmitter enabled
MPOC = 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.

- SERIAL LINE D CONFIGURED IN RS 422 (options RS 422)

J15	=	position 1-2	IC17	= indifferent
J27	=	(*1)	IC19	= MAX 483 or SN75176
J23	=	position 1-2	IC20	= MAX 483 or SN75176

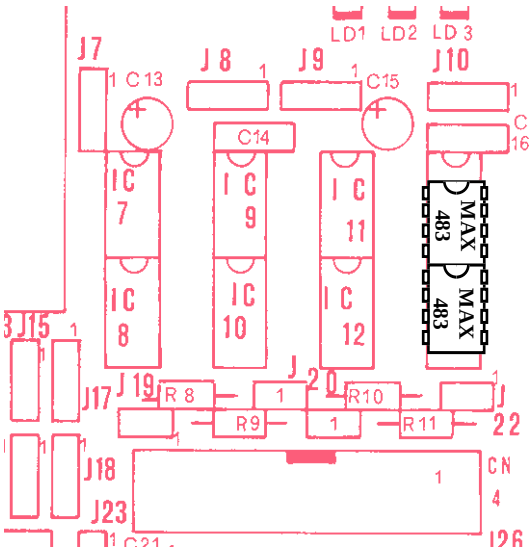
Status of signal MPOD, which is software managed, allows to enable or disable the transmitter as follows:

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

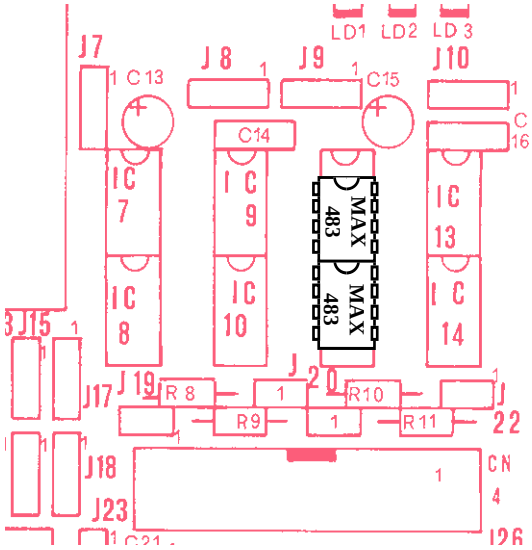
In point to point connections, signal MPOD can be always kept low (trasnmitter always enabled), while in multi point connections transmitter must be enabled only when a transmission is requested.

- SERIAL LINE D CONFIGURED IN RS 232 (default)

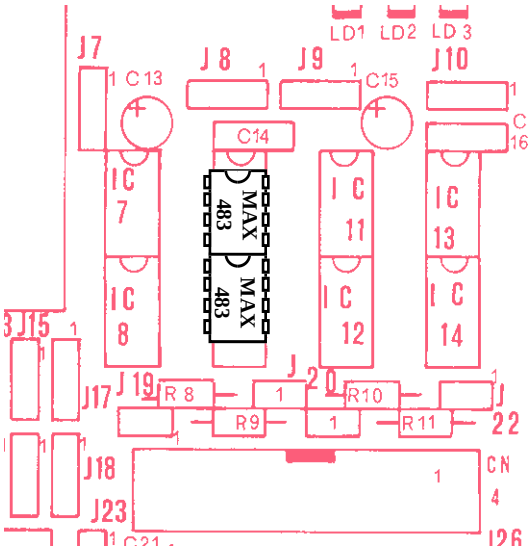
J15	=	position 2-3	IC17	= driver MAX 235
J27	=	indifferent	IC19	= indifferent
J23	=	indifferent	IC20	= indifferent



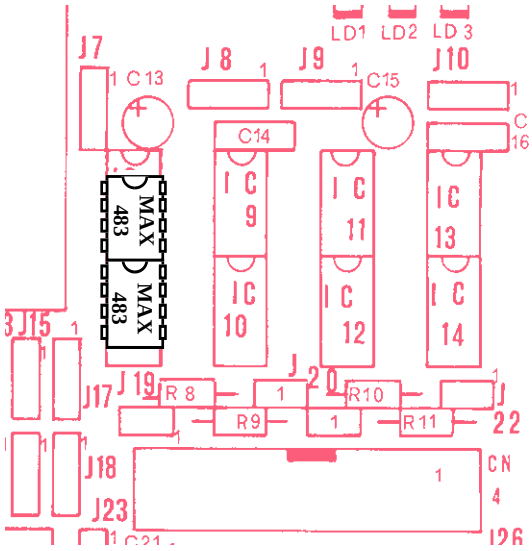
Serial line E in RS 422



Serial line F in RS 422



Serial line G in RS 422



Serial line H in RS 422

FIGURE 27: RS 422 DRIVERS FOR SERIAL LINES E÷H LOCATION

- SERIAL LINE D CONFIGURED IN RS 485 (options RS 485)

J15	=	position 1-2	IC17	= indifferent
J27	=	(*1)	IC19	= MAX 483 or SN75176
J23	=	position 2-3	IC20	= no component

In this modality the signals to use are pins 18 and 20 of connector CN3, that become transmission or reception lines according to the status of signal MPOD, managed by software, as follows:

MPOD = low level	= logic state 0	->	transmitter enabled
MPOD = 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.

- SERIAL LINE E CONFIGURED IN RS 422 (options RS 422)

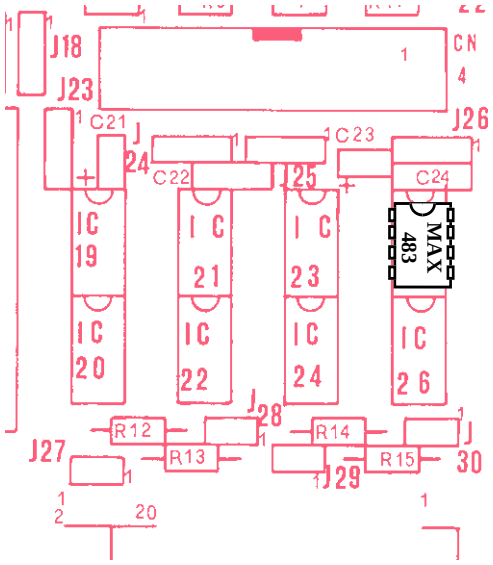
J14	=	position 1-2	IC16	= indifferent
J22	=	(*1)	IC14	= MAX 483 or SN75176
J10	=	position 1-2	IC13	= MAX 483 or SN75176

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

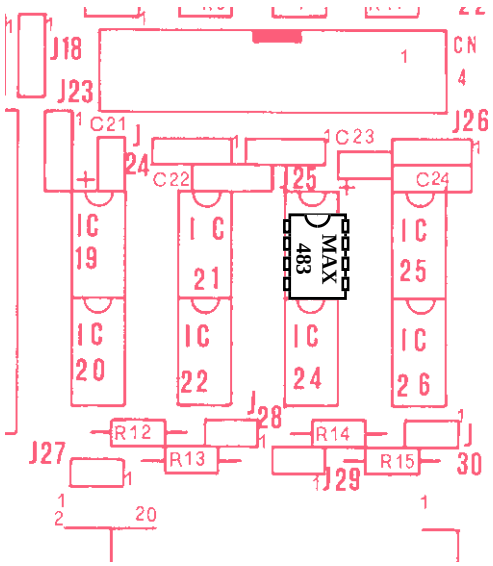
In point to point connections, signal MPOE can be always kept low (trasnmitter always enabled), while in multi point connections transmitter must be enabled only when a transmission is requested.

- SERIAL LINE E CONFIGURED IN RS 232 (default)

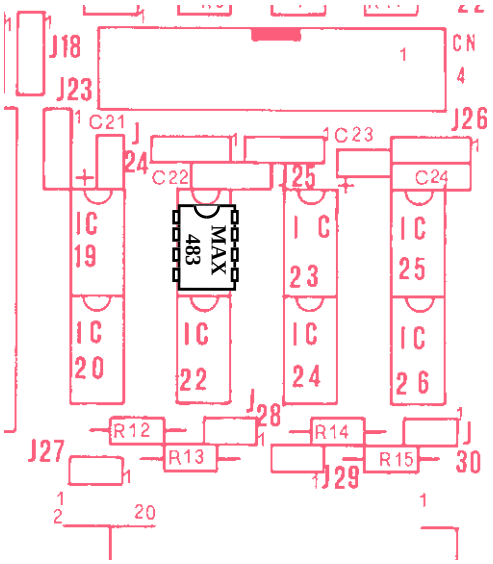
J14	=	position 2-3	IC16	= driver MAX 235
J22	=	indifferent	IC14	= indifferent
J10	=	indifferent	IC13	= indifferent



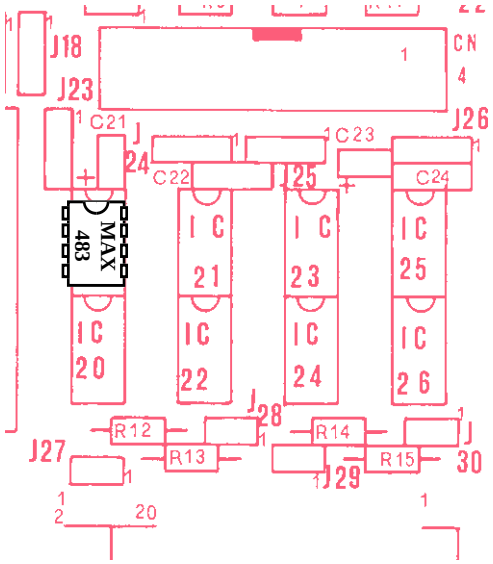
Serial linee A in RS 485



Serial linee B in RS 485



Serial line C in RS 485



Serial line D in RS 485

FIGURE 28: RS 485 DRIVERS FOR SERIAL LINES A÷D LOCATION

- SERIAL LINE E CONFIGURED IN RS 485 (options RS 485)

J14	=	position 1-2	IC16	=	indifferent
J22	=	(*1)	IC14	=	MAX 483 or SN75176
J10	=	position 2-3	IC23	=	no component

In this modality the signals to use are pins 2 and 4 of connector CN4, that become transmission or reception lines according to the status of signal MPOE, managed by software, as follows:

MPOE = low level	= logic state 0	->	transmitter enabled
MPOE = 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.

- SERIAL LINE F CONFIGURED IN RS 422 (options RS 422)

J13	=	position 1-2	IC16	=	indifferent
J21	=	(*1)	IC12	=	MAX 483 or SN75176
J9	=	position 1-2	IC11	=	MAX 483 or SN75176

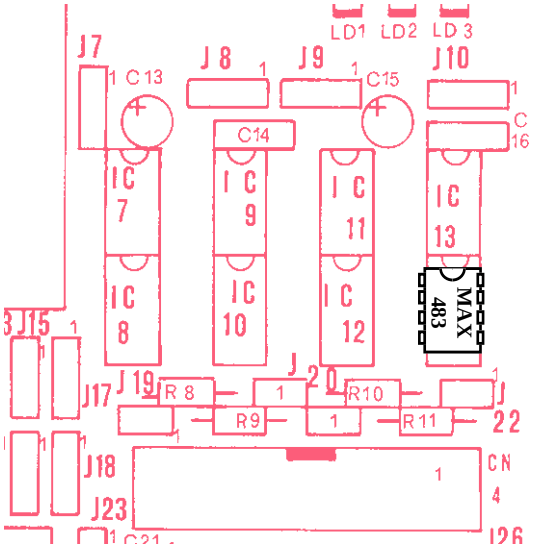
Status of signal MPOF, which is software managed, allows to enable or disable the transmitter as follows:

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

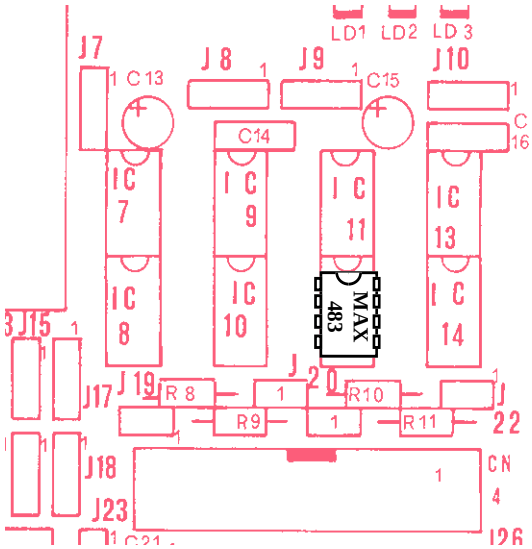
In point to point connections, signal MPOF can be always kept low (trasnmitter always enabled), while in multi point connections transmitter must be enabled only when a transmission is requested.

- SERIAL LINE F CONFIGURED IN RS 232 (default)

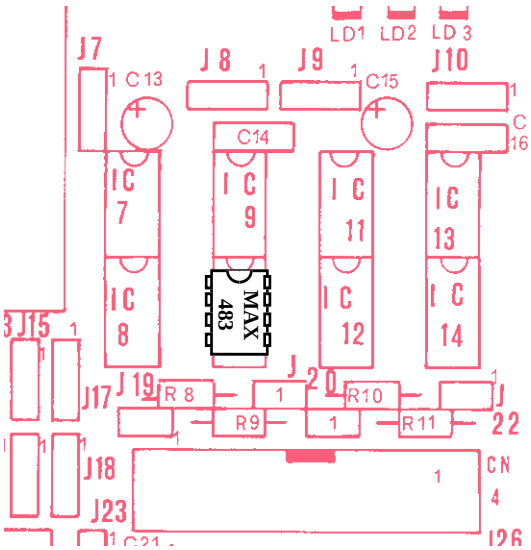
J13	=	position 2-3	IC16	=	driver MAX 235
J21	=	indifferent	IC12	=	indifferent
J9	=	indifferent	IC11	=	indifferent



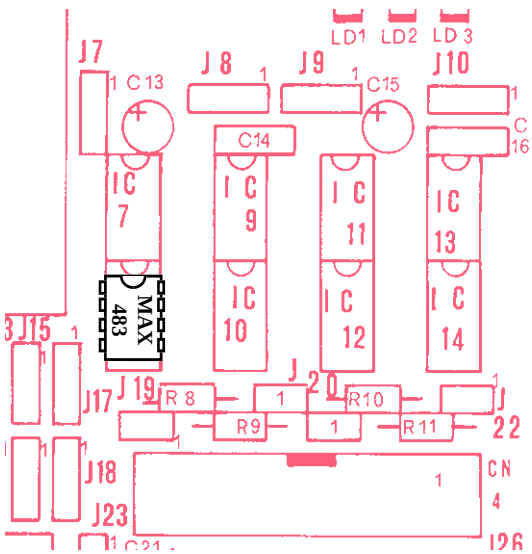
Serial line E in RS 485



Serial line F in RS 485



Serial line G in RS 485



Serial line H in RS 485

FIGURE 29: RS 485 DRIVERS FOR SERIAL LINES E÷H LOCATION

- SERIAL LINE F CONFIGURED IN RS 485 (options RS 485)

J13	=	position 1-2	IC16	=	indifferent
J21	=	(*1)	IC12	=	MAX 483 or SN75176
J9	=	position 2-3	IC11	=	no component

In this modality the signals to use are pins 3 and 10 of connector CN4, that become transmission or reception lines according to the status of signal MPOF, managed by software, as follows:

MPOF = low level	= logic state 0	->	transmitter enabled
MPOF = 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.

- SERIAL LINE G CONFIGURED IN RS 422 (options RS 422)

J12	=	position 1-2	IC15	=	indifferent
J20	=	(*1)	IC10	=	MAX 483 or SN75176
J8	=	position 1-2	IC9	=	MAX 483 or SN75176

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

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

In point to point connections, signal MPOG 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 G CONFIGURED IN RS 232 (default)

J12	=	position 2-3	IC15	=	driver MAX 235
J20	=	indifferent	IC10	=	indifferent
J8	=	indifferent	IC9	=	indifferent

- SERIAL LINE G CONFIGURED IN RS 485 (options RS 485)

J12	=	position 1-2	IC15	= indifferent
J20	=	(*1)	IC10	= MAX 483 or SN75176
J8	=	position 2-3	IC9	= no component

In this modality the signals to use are pins 12 and 14 of connector CN4, that become transmission or reception lines according to the status of signal MPOG, managed by software, as follows:

MPOG = low level	= logic state 0	->	transmitter enabled
MPOG = 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.

- SERIAL LINE H CONFIGURED IN RS 422 (options RS 422)

J11	=	position 1-2	IC15	= indifferent
J19	=	(*1)	IC8	= MAX 483 or SN75176
J7	=	position 1-2	IC7	= MAX 483 or SN75176

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

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

In point to point connections, signal MPOH 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 H CONFIGURED IN RS 232 (default)

J11	=	position 2-3	IC15	= driver MAX 235
J19	=	indifferent	IC8	= indifferent
J7	=	indifferent	IC7	= indifferent

- SERIAL LINE H CONFIGURED IN RS 485 (options RS 485)

J11	=	position 1-2	IC15	= indifferent
J19	=	(*1)	IC8	= MAX 483 or SN75176
J7	=	position 2-3	IC7	= no component

In this modality the signals to use are pins 18 and 20 of connector CN4, that become transmission or reception lines according to the status of signal MPOH, managed by software, as follows:

MPOH = low level	= logic state 0	->	transmitter enabled
MPOH = 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 circuit on the line by using the indicated jumpers. 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, signalsMPOA÷MPOH 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 7÷15 or to appendix A about SCC 2698.

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 2698 serial lines -> They generate an /NMI or an /INT interrupt on BUS **ABACO®**, according to the connection of jumpers J3÷J6. For further information please see jumpers chapter.

Setting the conditions that generate the interrupts is completely software manageable by programming SCC 2698 registers.
Please remark that **UCC 08** structure allows to use more that one card with interrupt actived at the same time but vectored interrupts cannot be used.

CONFIGURATION INPUT

UCC 08 board is provided with three 8-pins dip switch (DIP3, DIP4 and DIP5), 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 three dip switches of connected to three MPI (Multi Purpose Inputs) of SCC 2698, according to the following table:

DIP3	->	**MPI3**
DIP4	->	**MPI2**
DIP5	->	**MPI1**

Please refer to chapter “PERIPHERAL DEVICES SOFTWARE DESCRIPTION” for information about how to read them, while to locate them on the board please refer to figure 4.

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

UCC 08 board is mapped into a **64** bytes I/O addressing space, that can be allocated starting from different base addresses according to how the board is configured.
Three different mapping modalities can be used:

- Normal addressing
- Extended addressing
- Enhanced extended addressing

Normal addressing allows to allocate the 64 board's registers inside an addressing space of 256 bytes, so to select the address switches 7 and 8 of DIP2 must be used, while switches form 1 to 6 must be always OFF and J1 must be always disconnected.
Extended addressing allows to allocate the registers inside a 64 KBytes addressing space, so also the dip switch DIP1 is used in addition to the one above described, used exactly in the same way.
Enhanced extended addressing, at last, opens a 16M Bytes addressing space so DIP1 and DIP2 are completely used to decide the address. In addition, in this mode J1 pins allow to include or exclude signals from A16 to A20 and /M1 from the composition of the address MSB. /M1 signal is useful to put the card in the same bus with other cards that user vectored interrupts.
This feature allows to use several **UCC 08** cards on the same BUS **ABACO®**, 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.
Here follows the corrispondance between dips configuration and address signals, to easily locate such component please refer to figure 4.

- Normal addressing mode:

DIP2.7	->	Address A6
DIP2.8	->	Address A7

- Extended addressing mode; above addressing signals plus the following:

DIP1.1	->	Address A8
DIP1.2	->	Address A9
DIP1.3	->	Address A10
DIP1.4	->	Address A11
DIP1.5	->	Address A12
DIP1.6	->	Address A13
DIP1.7	->	Address A14
DIP1.8	->	Address A16

- Enhanced extended addressing mode; previous addressing signals plus the following:

DIP2.1	->	Address A16 (*)
DIP2.2	->	Address A17 (*)
DIP2.3	->	Address A18 (*)
DIP2.4	->	Address A19 (*)
DIP2.5	->	Address A20 (*)

(*) if connected through J1

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, here follows a possible mapping in normal mode:
To address **UCC 08** board's registers from address 080H, it must be configured as follows:

J1	->	No connection
J2	->	Connection 1-2
DIP1.x	->	Indifferent
DIP2.1	->	OFF
DIP2.2	->	OFF
DIP2.3	->	OFF
DIP2.4	->	OFF
DIP2.5	->	OFF
DIP2.6	->	OFF
DIP2.7	->	ON
DIP2.8	->	OFF

As an example, here follows a possible mapping in extended mode:
To address **UCC 08** board's registers from address 04000H, it must be configured as follows:

J1	->	No connection
J2	->	Connession 2-3

DIP1.1	->	ON
DIP1.2	->	ON
DIP1.3	->	OFF
DIP1.4	->	ON
DIP1.5	->	ON
DIP1.6	->	ON
DIP1.7	->	ON
DIP1.8	->	ON

DIP2.1	->	OFF
DIP2.2	->	OFF
DIP2.3	->	OFF
DIP2.4	->	OFF
DIP2.5	->	OFF
DIP2.6	->	OFF
DIP2.7	->	ON
DIP2.8	->	ON

To locate the above mentioned componentes on the board please refer to figure 4.

I/O ADDRESSES

Indication <**baseaddr**> means the base address of the board decided with dip switches and J1, as previously described.

DEV.	ADDRESS	READ	WRITE	MEANING
SCC 2698	<baseadd>+00H	MR1A, 2A	MR1A, 2A	Mode registers 1 and 2 serial A
	<baseadd>+01H	SRA	CSRA	Status and clock registers serial A
	<baseadd>+02H	Reserved	CRA	Command register serial A
	<baseadd>+03H	RHRA	THRA	Transmit and receive holding serial A
	<baseadd>+04H	IPCRA	ACRA	Input port change and auxiliary control A
	<baseadd>+05H	ISRA	IMRA	Interrupt status and mask port A
	<baseadd>+06H	CTUA	CTURA	Counter/timer high data register port A
	<baseadd>+07H	CTLA	CTLRA	Counter/timer low data register port A
	<baseadd>+08H	MR1B, 2B	MR1B, 2B	Mode registers 1 and 2 serial B
	<baseadd>+09H	SRB	CSRB	Status and clock registers serial B
	<baseadd>+0AH	Reserved	CRB	Command register serial B
	<baseadd>+0BH	RHRB	THRB	Transmit and receive holding serial B
	<baseadd>+0CH	Reserved	Reserved	Reserved: never access these registers
	<baseadd>+0DH	Input port A	OPCRA	Input status and output registers port A
	<baseadd>+0EH	Start C/T A	Reserved	Enable counter/timer port A
	<baseadd>+0FH	Stop C/T A	Reserved	Disable counter/timer port A
	<baseadd>+10H	MR1C, 2C	MR1C, 2C	Mode registers 1 and 2 serial C
	<baseadd>+11H	SRC	CSRC	Status and clock registers serial C
	<baseadd>+12H	Reserved	CRC	Command register serial C
	<baseadd>+13H	RHRC	THRC	Transmit and receive holding serial C
	<baseadd>+14H	IPCRB	ACRB	Input port change and auxiliary control C
	<baseadd>+15H	ISRB	IMRB	Interrupt status and mask port C
	<baseadd>+16H	CTUB	CTURB	Counter/timer high data register port C
	<baseadd>+17H	CTLB	CTLRB	Counter/timer low data register port C
	<baseadd>+18H	MR1D, 2D	MR1D, 2D	Mode registers 1 and 2 serial D
	<baseadd>+19H	SRD	CSRD	Status and clock registers serial D
	<baseadd>+1AH	Reserved	CRD	Command register serial D
	<baseadd>+1BH	RHRD	THRD	Transmit and receive holding serial D
	<baseadd>+1CH	Reserved	Reserved	Reserved: never access these registers
	<baseadd>+1DH	Input port B	OPCRB	Input status and output registers port B

FIGURE 30: INTERNAL REGISTERS ADDRESSING TABLE - PART 1

DEV.	ADDRESS	READ	WRITE	MEANING
SCC 2698	<baseadd>+1EH	Start C/T B	Reserved	Enable counter/timer port B
	<baseadd>+1FH	Stop C/T B	Reserved	Disable counter/timer port B
	<baseadd>+20H	MR1E, 2E	MR1E, 2E	Mode registers 1 and 2 serial E
	<baseadd>+21H	SRE	CSRE	Status and clock registers serial E
	<baseadd>+22H	Reserved	CRE	Command register serial E
	<baseadd>+23H	RHRE	THRE	Transmit and receive holding serial E
	<baseadd>+24H	IPCRC	ACRC	Input port change and auxiliary control C
	<baseadd>+25H	ISRC	IMRC	Interrupt status and mask port C
	<baseadd>+26H	CTUC	CTURC	Counter/timer high data register port C
	<baseadd>+27H	CTLC	CTLRC	Counter/timer low data register port C
	<baseadd>+28H	MR1E, 2F	MR1E, 2F	Mode registers 1 and 2 serial F
	<baseadd>+29H	SRF	CSRF	Status and clock registers serial F
	<baseadd>+2AH	Reserved	CRF	Command register serial F
	<baseadd>+2BH	RHRF	THRF	Transmit and receive holding serial F
	<baseadd>+2CH	Reserved	Reserved	Reserved: never access these registers
	<baseadd>+2DH	Input port C	OPCRC	Input status and output registers port C
	<baseadd>+2EH	Start C/T C	Reserved	Enable counter/timer port C
	<baseadd>+2FH	Stop C/T C	Reserved	Disable counter/timer port C
	<baseadd>+30H	MR1G, 2G	MR1G, 2G	Mode registers 1 and 2 serial G
	<baseadd>+31H	SRG	CSRG	Status and clock registers serial G
	<baseadd>+32H	Reserved	CRG	Command register serial G
	<baseadd>+33H	RHRG	THRG	Transmit and receive holding serial G
	<baseadd>+34H	IPCRD	ACRD	Input port change and auxiliary control D
	<baseadd>+35H	ISRD	IMRD	Interrupt status and mask port d
	<baseadd>+36H	CTUD	CTURD	Counter/timer high data register port D
	<baseadd>+37H	CTLD	CTLRD	Counter/timer low data register port D
	<baseadd>+38H	MR1G, 2G	MR1G, 2G	Mode registers 1 and 2 serial G
	<baseadd>+39H	SRG	CSRG	Status and clock registers serial G
	<baseadd>+3AH	Reserved	CRG	Command register serial G
	<baseadd>+3BH	RHRG	THRG	Transmit and receive holding serial G
	<baseadd>+3CH	Reserved	Reserved	Reserved: never access these registers
	<baseadd>+3DH	Input port D	OPCRD	Input status and output registers port D
	<baseadd>+3EH	Start C/T D	Reserved	Enable counter/timer port D
	<baseadd>+3FH	Stop C/T D	Reserved	Disable counter/timer port D

FIGURE 31: INTERNAL REGISTERS ADDRESSING TABLE - PART 2

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

UCC 08 board is provided with 24configuraion inputs that can be read by software as described here. The dip switches DIP3, DIP4 and DIP5 are connected to as many MPI (Multi Purpose Inputs) of SCC 2698, and can be acquired by software performing a read operation from status registers of the two sections with the following correspondace:

DIP3.1	->	Bit 5 Input Port A
DIP3.2	->	Bit 7 Input Port A
DIP3.3	->	Bit 5 Input Port B
DIP3.4	->	Bit 7 Input Port B
DIP3.5	->	Bit 5 Input Port C
DIP3.6	->	Bit 7 Input Port C
DIP3.7	->	Bit 5 Input Port D
DIP3.8	->	Bit 7 Input Port D
DIP4.1	->	Bit 4 Input Port A
DIP4.2	->	Bit 6 Input Port A
DIP4.3	->	Bit 4 Input Port B
DIP4.4	->	Bit 6 Input Port B
DIP4.5	->	Bit 4 Input Port C
DIP4.6	->	Bit 6 Input Port C
DIP4.7	->	Bit 4 Input Port D
DIP4.8	->	Bit 6 Input Port D
DIP5.1	->	Bit 1 Input Port A
DIP5.2	->	Bit 3 Input Port A
DIP5.3	->	Bit 1 Input Port B
DIP5.4	->	Bit 3 Input Port B
DIP5.5	->	Bit 1 Input Port C
DIP5.6	->	Bit 3 Input Port C
DIP5.7	->	Bit 1 Input Port D
DIP5.8	->	Bit 3 Input Port D

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**.
MPI ports registers of SCC 2698 are described in appendix A.

HANDSHAKE

Each **UCC 08** serial port is provided with two handshake signals commonly indicated with RTSY RS 232 and CTSy RS 232 in this manual, where y stands for the serial port name (from A to H). Here follows the correspondance between these communication control lines and UART pins and internal registers.

SIGNAL	PIN SCC 2698	BIT AND REGISTER
CTSA	MPI0A	Bit 0 of Input Port A or Bit 0 of IPCRA
CTSB	MPI0B	Bit 2 of Input Port A or Bit 2 of IPCRA
CTSC	MPI0C	Bit 0 of Input Port B or Bit 0 of IPCRB
CTSD	MPI0D	Bit 2 of Input Port B or Bit 2 of IPCRB
CTSE	MPI0E	Bit 0 of Input Port C or Bit 0 of IPCRC
CTSF	MPI0F	Bit 2 of Input Port C or Bit 2 of IPCRC
CTSG	MPI0G	Bit 0 of Input Port D or Bit 0 of IPCRD
CTSH	MPI0H	Bit 2 of Input Port D or Bit 2 of IPCRD
RTSA	MPOA	Bits from 0 to 2 of OPCRA and bits from 4 to 7 of CRA
RTSB	MPOB	Bits from 4 to 6 of OPCRA and bits from 4 to 7 of CRB
RTSC	MPOC	Bits from 0 to 2 of OPCRB and bits from 4 to 7 of CRC
RTSD	MPOD	Bits from 4 to 6 of OPCRB and bits from 4 to 7 of CRD
RTSE	MPOE	Bits from 0 to 2 of OPCRC and bits from 4 to 7 of CRE
RTSF	MPOF	Bits from 4 to 6 of OPCRC and bits from 4 to 7 of CRF
RTSG	MPOG	Bits from 0 to 2 of OPCRD and bits from 4 to 7 of CRG
RTSH	MPOH	Bits from 4 to 6 of OPCRD and bits from 4 to 7 of CRH

FIGURE 32: CORRESPONDANCE BETWEEN HANDSHAKES, SIGNALS AND SCC 2698 REGISTERS

Please remark that both output handshakes (RTS) and input handshakes (CTS) can be managed in autonomy by UART even with different working modalities software defined by the user.
For further information about the above signals and relative registers please refer to appendix A.

EXTERNAL CARDS

UCC 08 can be connected to a wide range of block modules and operator interface systems 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 connectors. Even single EURO 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®**, or search it on **grifo®** CD or web site, if required.

GPC® 188F

General Purpose Controller 80C188

80C188 µP 20MHz; 1 RS 232 line; 1 RS 232, RS 422-485 or Current Loop line; 24 TTL I/O lines; 1M EPROM or 512K FLASH; 1M RAM Lithium battery backed; 8K serial EEPROM; RTC; Watch Dog; 8 Dip switch; 3 Timer Counter; 8 13 bit A/D lines; Power failure; activity LEDs; single power supply +5Vdc.

GPC® 15A

General Purpose Controller 84C15

Full CMOS card, 10÷20 MHz 84C15 CPU; 512K EPROM or FLASH; 128K RAM; 8K RAM and RTC backed; 8K serial EEPROM; 1 RS 232 line; 1 RS 232 line or RS 422-485 or Current Loop line; 32 or 40 TTL I/O lines; CTC; Watch dog; 2 Dip switches; Buzzer.

GPC® 150

General Purpose Controller 84C15

Microprocessor Z80 at 16 MHz; implementation completely CMOS; 512K EPROM or FLASH; 512K SRAM; RTC; Back-Up through external Lithium battery; 4M serail FLASH ; 1 serial line RS 232 plus 1 RS 232 or RS 422-485 or current loop; 40 I/O TTL; 2 timer/counter; 2 watch dog; dip switch; EEPROM; A/D converter with resolution 12 bit; activity LED.

GPC® 15R

General Purpose Controller 84C15

84C15 µP, 10÷16 MHz; 1 RS 232 line; 1 RS 232 or RS 422-485 or C. L. line; 16÷24 TTL I/O lines; 16 Opto-in; 8 Relays; 4 Opto Coupled Timers Counters; 512K EPROM or FLASH; 512K RAM and RTC backed; 8K serial EEPROM; 8K Backed RAM modul; Buzzer; 1 Activity LED; Watch dog; 4÷12 readable DIPs; LCD Interface.

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.

GPC® 323

General Purpose Controller 51 family

80C32 µP, 14 MHz; Full CMOS; 1 RS 232 line (software); 1 RS 232 or RS 422-485 or Current Loop line; 24 TTL I/O lines; 11 A/D 12 bits lines; 3 Timers Counters; 64K EPROM; 64K RAM; 32K RAM and RTC backed; 32K DIL EEPROM; 8K serial EEPROM; Buzzer; 2 Activity LED; Watch dog; 5 readable DIPs; LCD Interface.

GPC® 553

General Purpose Controller 80C552

80C552 µP, 22÷33 MHz; 1 RS 232 line (software); 1 RS 232 or RS 422-485 or Current Loop line; 16 TTL I/O lines; 8 A/D 10 bits lines; 3 Timers Counters; 64K EPROM; 64K RAM; 32K RAM and RTC backed; 32K DIL EEPROM; 8K serial EEPROM; 2 PWM lines; 1 Activity LED; Watch dog; 5 readable DIPs; LCD Interface.

GPC® 153

General Purpose Controller Z80

84C15 µP, 10÷16 MHz; Full CMOS; 1 RS 232 line; 1 RS 232 or RS 422-485 or Current Loop line; 16 TTL I/O lines; 8 A/D 12 bits lines; 2÷4 Timers Counters; 512K EPROM or FLASH; 512K RAM and RTC backed; 8K serial EEPROM; Buzzer; 1 Activity LED; Watch dog; 8 readable DIPs; LCD Interface.

GPC® 183

General Purpose Controller Z180

Z180 µP, 10÷16 MHz; Full CMOS; 1 RS 232 line; 1 RS 232 or RS 422-485 or Current Loop line; 24 TTL I/O lines; 11 A/D 12 bits lines; 2 Timers Counters; 512K EPROM or FLASH; 512K RAM and RTC backed; 8K serial EEPROM; Buzzer; 2 Activity LED; Watch dog; 4 readable DIPs; LCD Interface.

GPC® 184

General Purpose Controller Z80195

Microprocessor Z80195 at 22 MHz; implementation completely CMOS; 512K EPROM or FLASH; 512K RAM; Back-Up with Lithium battery internal or external; 1 serial line RS 232 + 1 RS 232 or RS 422-485 or current loop + 1 TTL; 18 I/O TTL; 4 timer/counter 8 bits; 2 timer 16 bits; Watch Dog; Real Time Clock; activity LED; EEPROM; interface for **ABACO®** I/O BUS.

GPC® 324/D

“4” Type General Purpose Controller 80C32/320

80C32 or 80C320 µP, 14÷22 MHz; Full CMOS; 1 RS 232 line; 1 RS 232 or RS 422-485 or Current Loop line; 4÷16 TTL I/O lines; 3 Timers Counters; 64K EPROM; 64K RAM; 32K RAM backed; 32K DIL E2; 8K serial EEPROM; Watch dog; 1 readable DIP; LCD Interface; Abaco® I/O BUS; 5Vdc Power supply; Size: 100x50 mm.

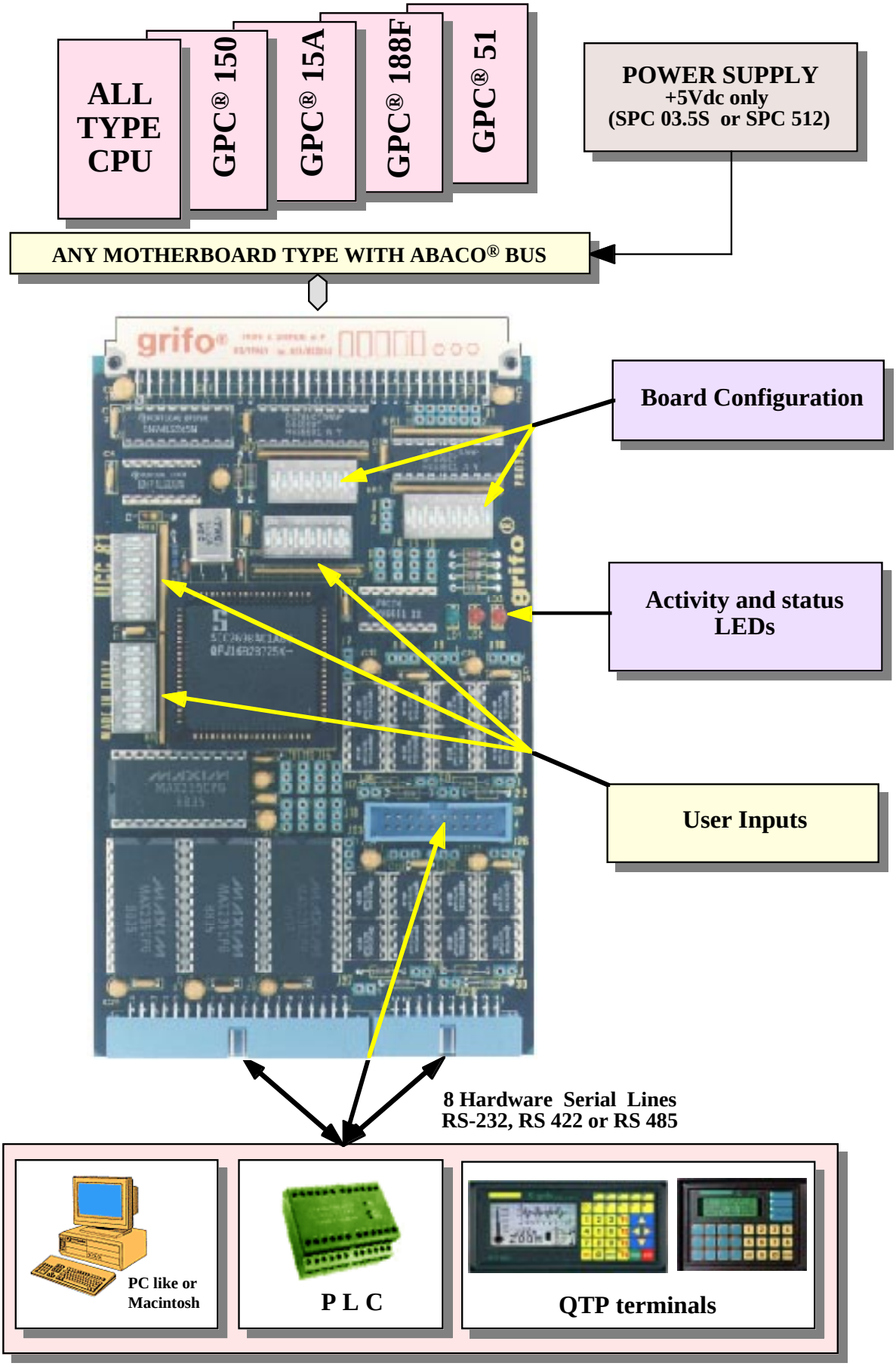


FIGURE 33: CONNECTIONS EXAMPLE

GPC® 554

General Purpose Controller 80C552

Microprocessor 80C552 at 22 MHz; implementation completely CMOS; 32K EPROM; 32 K SRAM; 32 K EEPROM or SRAM; EEPROM; 2 RS 232 serial lines; 16 I/O TTL; 2 PWM lines; 16 bits Timer/Counter; Watch Dog; 6 signals A/D converter with resolution 10 bit; interface for **ABACO®** I/O BUS.

GPC® 154

"4" Type General Purpose Controller Z80

84C15 µP, 10÷16 MHz; Full CMOS; 1 RS 232 line; 1 RS 232 or RS 422-485 line; 16 TTL I/O lines; 2÷4 Timers Counters; 512K EPROM or FLASH; 512K RAM and RTC backed; 8K serial EEPROM; Watch dog; 2 readable DIPs; LCD Interface; Abaco® I/O BUS; 5Vdc Power supply; Size: 100x50 mm.

GPC® 884

General Purpose Controller Am188ES

Microprocessor AMD Am188ES up to 40 MHz16 bits; implementation completely CMOS; serie 4 format; 512K EPROM or FLASH; 512K SRAM backed with Lithium battery; RTC; 1 RS 232 serial line + 1 RS 232 or RS 422-485 or current loop; 16 I/O TTL; 3 timer/counter; watch dog; EEPROM; 11 signals A/D converter with 12 bit resolution; interface for **ABACO®** I/O BUS.

GPC® 114

General Purpose Controller 68HC11

Microprocessor 68HC11A1 at 8 MHz; implementation completely CMOS; serie 4 format; 32K EPROM; 32K SRAM backed with Lithium battery; 32K EPROM, SRAM, EEPROM; RTC; 1 serial line RS 232 or RS 422-485; 10 I/O TTL; 3 timer/counter; watch dog; 8 signals A/D converter with resolution 8 bit; 1 asunchronous serial line; extremly low power consumption; interface for **ABACO®** I/O BUS.

GPC® AM4

General Purpose Controller ATmega103

Microprocessor ATmega103 at 5.5 MHz; implementation completely CMOS; 128K internal FLASH; 32K SRAM; Back-Up with Lithium battery internal or external; 1 serial line RS 232 or RS 422-485 or current loop; 16 I/O TTL; 8 linee A/D resolution 10 bits; 2 timer/counter; Watch Dog; Real Time Clock; 4K internal EEPROM; interface for ISP programming; interface forr **ABACO®** I/O BUS.

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 **UCC 08**.

Manual TEXAS INSTRUMENTS:	*The TTL Data Book - SN54/74 Families*
Manual TEXAS INSTRUMENTS:	*RS-422 and RS-485 Interface Circuits*
Manual MAXIM:	*New Releases Data Book - Volume IV*
Data Sheet PHILIPS:	*SCC2698B (Enhanced Octal UART)*

The described manual can be requested directly to manufacturer or local dealers. Alternatively this information and/or upgrades can be found in specific internet web pages, of the listed companies.

APPENDIX A: DATA SHEET OF SCC 2698B

INTEGRATED CIRCUITS

DATA SHEET

SCC2698B
Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

Product specification
Supersedes data of 1998 Sep 04

2000 Jan 31

Philips Semiconductors

PHILIPS

DESCRIPTION

The SCC2698B Enhanced Octal Universal Asynchronous Receiver/Transmitter (Octal UART) is a single chip MOS-LSI communications device that provides eight full-duplex asynchronous receiver/transmitter channels in a single package. It is fabricated with CMOS technology which combines the benefits of high density and low power consumption.

The operating speed of each receiver and transmitter can be selected independently as one of 26 fixed baud rates, a 16X clock derived from a programmable counter/timer, or an external 1X or 16X clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the Octal UART particularly attractive for dual-speed channel applications such as clustered terminal systems.

The receiver is quadruple buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a handshaking (RTS/CTS) capability is provided to disable a remote UART transmitter when the receiver buffer is full.

The UART provides a power-down mode in which the oscillator is frozen but the register contents are stored. This results in reduced power consumption on the order of several magnitudes. The Octal UART is fully TTL compatible and operates from a single +5V power supply.

The SCC2698B is an upwardly compatible version of the 2698A Octal UART. In PLCC packaging, it is enhanced by the addition of receiver ready or FIFO full status outputs, and transmitter empty status outputs for each channel on 16 multipurpose I/O pins. The multipurpose pins of the 2698B RIO pins, thus DMA and modem control is provided.

FEATURES

- Eight full-duplex independent asynchronous receiver/transmitters
- Quadruple buffered receiver data register
- Programmable data format:
  - 5 to 8 data bits plus parity
  - Odd, even, no parity or force parity
  - 1, 1.5 or 2 stop bits programmable in 1/16-bit increments
- Baud rate for the receiver and transmitter selectable from:
  - 26 fixed rates: 50 to 38.4K baud
    Non-standard rates to 115.2K baud
  - User-defined rates from the programmable counter/timer associated with each of four blocks
  - External 1x or 16x clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - Normal (full-duplex), automatic echo, local loop back, remote loopback
- Four multi-function programmable 16-bit counter/timers
- Four interrupt outputs with eight maskable interrupting conditions for each output
- Receiver ready/FIFO full and transmitter ready status available on 16 multi-function pins in PLCC package
- On-chip crystal oscillator
- TTL compatible
- Single +5V power supply with low power mode
- Eight multi-purpose output pins
- Sixteen multi-purpose I/O pins
- Sixteen multi-purpose Input pins with pull-up resistors

ORDERING INFORMATION

PACKAGES	COMMERCIAL	INDUSTRIAL	DWG #
	V_{CC} = +5V ±5%, T_A = 0°C to +70°C	V_{CC} = +5V ±5%, T_A = –40°C to +85°C	
84-Pin Plastic Leaded Chip Carrier (PLCC)	SCC2698BC1A84	SCC2698BE1A84	SOT189-3

NOTE: Pin Grid Array (PGA) package version is available from Philips Components Military Division.

ABSOLUTE MAXIMUM RATINGS[1]

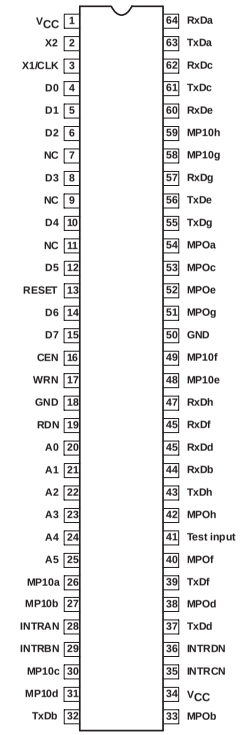
SYMBOL	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature range[2]	Note 4	°C
T_{STG}	Storage temperature range	–65 to +150	°C
V_{CC}	Voltage from V_{DD} to GND[3]	–0.5 to +7.0	V
V_S	Voltage from any pin to ground[3]	–0.5 to V_{CC} +0.5	V
P_D	Power dissipation	1	W

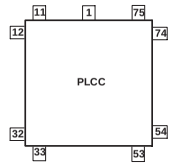
NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range. See ordering information table for applicable temperature range and operating supply range.

Philips Semiconductors Product specification

Enhanced octal universal asynchronous receiver/transmitter (Octal UART) SCC2698B

PIN CONFIGURATIONS





Pin	Function	Pin	Function	Pin	Function
1	TxDa	29	A3	57	RxDd
2	MPP2g	30	MPP2b	58	RxDf
3	RxDa	31	A4	59	RxDh
4	MPP2h	32	A5	60	MPI1e
5	V_{CC}	33	MPI0a	61	MPI0e
6	X2	34	MPI0b	62	MPI1f
7	X1/CLK	35	INTRAN	63	MPI0f
8	D0	36	INTRBN	64	MPP1e
9	D1	37	MPI0c	65	GND
10	D2	38	MPI1c	66	MPP1f
11	D3	39	MPI0d	67	MPOg
12	D4	40	MPI1d	68	MPP2e
13	D5	41	TxDb	69	MPOe
14	MPI1a	42	MPP1c	70	MPP2f
15	RESET	43	MPOb	71	MPOc
16	D6	44	MPP1d	72	MPOa
17	D7	45	V_{CC}	73	TxDg
18	CEN	46	INTRCN	74	TxDe
19	WRN	47	INTRDN	75	RxDg
20	GND	48	MPP2c	76	MPI0g
21	MPI1b	49	TxDd	77	MPI0h
22	RDN	50	MPP2d	78	MPI1g
23	A0	51	MPOd	79	RxDe
24	MPP1a	52	TxDf	80	MPIh
25	A1	53	MPOf	81	TxDc
26	MPP1b	54	MPOh	82	MPP1g
27	A2	55	TxDh	83	RxDc
28	MPP2a	56	RxDb	84	MPP1h

SD00184

Figure 1. Pin Configurations

BLOCK DIAGRAM

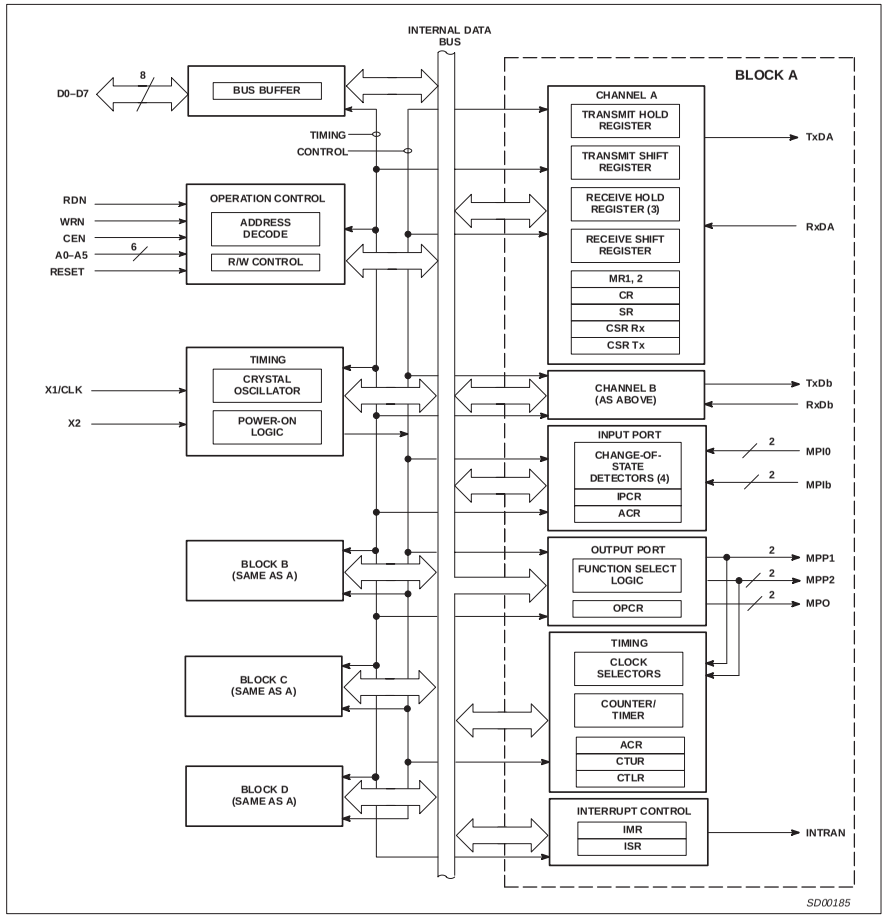


Figure 2. Block Diagram

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
D0–D7	8–13, 16, 17	I/O	**Data Bus:** Active–High 8-bit bidirectional 3-State data bus. Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the Octal UART take place over this bus. The direction of the transfer is controlled by the WRN and RDN inputs when the CEN input is low. When the CEN input is High, the data bus is in the 3-State condition.
CEN	18	I	**Chip Enable:** Active-Low input. When Low, data transfers between the CPU and the Octal UART are enabled on D0–D7 as controlled by the WRN, RDN and A0–A5 inputs. When CEN is High, the Octal UART is effectively isolated from the data bus and D0–D7 are placed in the 3-State condition.
WRN	19	I	**Write Strobe:** Active-Low input. A Low on this pin while CEN is Low causes the contents of the data bus to be transferred to the register selected by A0–A5. The transfer occurs on the trailing (rising) edge of the signal.
RDN	22	I	**Read Strobe:** Active-Low input. A Low on this pin while CEN is Low causes the contents of the register selected by A0–A5 to be placed on the data bus. The read cycle begins on the leading (falling) edge of RDN.
A0–A5	23, 25, 27, 29, 31, 32	I	**Address Inputs:** Active-High address inputs to select the Octal UART registers for read/write operations.
RESET	15	I	**Reset:** Master reset. A High on this pin clears the status register (SR), clears the interrupt mask register (IMR), clears the interrupt status register (ISR), clears the output port configuration register (OPCR), places the receiver and transmitter in the inactive state causing the TxD output to go to the marking (High) state, and stops the counter/timer. Clears power-down mode and interrupts. Clears Test Modes, sets MR pointer to MR1.
INTRAN–INTRDN	35, 36, 46, 47	O	**Interrupt Request:** This active-Low open drain output is asserted on occurrence of one or more of eight maskable interrupting conditions. The CPU can read the interrupt status register to determine the interrupting condition(s). These pins require a pullup device and may be wire ORed.
X1/CLK	7	I	**Crystal 1:** Crystal or external clock input. When using the crystal oscillator, this pin serves as the connection for one side of the crystal. If a crystal is not used, an external clock is supplied at this input. An external clock (or crystal) is required even if the internal baud rate generator is not utilized. This clock is used to drive the internal baud rate generator, as an optional input to the timer/counter, and to provide other clocking signals required by the chip.
X2	6	I	**Crystal 2:** Connection for other side of crystal. If an external source is used instead of a crystal, this connection should be left open (see Figure 9).
RxDa–RxDh	3, 56, 83, 57, 79, 58, 75, 59	I	**Receiver Serial Data Input:** The least significant bit is received first. If external receiver clock is specified, this input is sampled on the rising edge of the clock. If internal clock is used, the RxD input is sampled on the rising edge of the RxC1x signal as seen on the MPO pin.
TxDa–TxDh	1, 41, 81, 49, 74, 52, 73, 55	O	**Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the marking (High) condition when the transmitter is idle or disabled and when the Octal UART is operating in local loopback mode. If external transmitter is specified, the data is shifted on the falling edge of the transmitter clock. If internal clock is used, the TxD output changes on the falling edge of the TxC1x signal as seen on the MPO pin.
MPOa–MPOh	72, 43, 71, 51, 69, 53, 67, 54	O	**Multi-Purpose Output:** Each of the four DUARTS has two MPO pins (one per UART). One of the following eight functions can be selected for this output pin by programming the OPCR (output port configuration register). Note that reset conditions MPO pins to RTSN. **RTSN –** Request to send active-Low output. This output is asserted and negated via the command register. By appropriate programming of the mode registers, (MR1[7])=1 RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted or when the receiver FIFO and shift register are full. RTSN is an internal signal which normally represents the condition of the receiver FIFO not full, i.e., the receiver can request more data to be sent. However, it can also be controlled by the transmitter empty and the commands 8h and 9h written to the CR (command register). **C/TO –** The counter/timer output. **TxC1X –** The 1X clock for the transmitter. **TxC16X –** The 16X clock for the transmitter. **RxC1X –** The 1X clock for the receiver. **RxC16X –** The 16X clock for the receiver. **TxRDY –** Transmitter holding register empty signal. **RxRDY/FFULL –** Receiver FIFO not empty/full signal.
MPI0a–MPI0h	33, 34, 37, 39, 61, 63, 76, 77	I	**Multi-Purpose Input 0:** This pin (one in each UART) is programmable. Its state can always be read through the IPCR bit 0, or the IPR bit 0. **CTSN:** By programming MR2[4] to a 1, this input controls the clear-to-send function for the transmitter. It is active low. This pin is provided with a change-of-state detector.

PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
MPI1a–MPI1h	14, 21, 38, 40, 60, 62, 78, 80	I	**Multi-Purpose Input 1:** This pin (one for each UART) is programmable. Its state can always be determined by reading the IPCR bit 1 or IPR bit 1. **C/TCLK –** This input will serve as the external clock for the counter/timer when ACR[5] is set to 0. This occurs only for channels a, c, e, and g since there is one counter/timer for each DUART block. This pin is provided with a change-of-state detector.
MPP1a–MPP1h	24, 26, 42, 44, 64, 66, 82, 84	I/O	**Multi-Purpose Pin 1:** This pin (one for each UART) is programmed to be an input or an output according to the state of OPCR[7]. (0 = input, 1 = output). The state of the multi-purpose pin can always be determined by reading the IPR. When programmed as an input, it will be the transmitter clock (TxCLK). It will be 1x or 16x according to the clock select registers (CSR[3.0]). When programmed as an output, it will be the status register TxRDY bit. These pins have a small pull-up device.
MPP2a–MPP2h	28, 30, 48, 50, 68, 70, 2, 4	I/O	**Multi-Purpose Pin 2:** This pin (one for each UART) is programmed to be an input or an output according to the state of OPCR[7]. (0 = input, 1 = output). The state of the multi-purpose pin can always be determined by reading the IPR. When programmed as an input, it will be the receiver clock (RxCLK). It will be 1x or 16x according to the clock select registers (CSR[7:4). When programmed as an output, it will be the ISR status register RxRDY/FIFO full bit. These pins have a small pull-up device.
Test Input	–	I	**Test Input:** This pin is used as an input for test purposes at the factory while in test mode. This pin can be treated as 'N/C' by the user. It can be tied high, or left open.
V_{CC}	5, 45	I	**Power Supply:** +5V supply input.
GND	20, 65	I	**Ground**

BLOCK DIAGRAM

As shown in the block diagram, the Octal UART consists of: data bus buffer, interrupt control, operation control, timing, and eight receiver and transmitter channels. The eight channels are divided into four different blocks, each block independent of each other (see Figure 3). Figure 2 represents the DUART block.

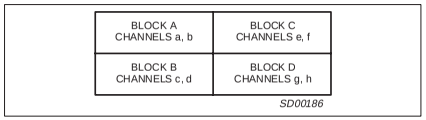


Figure 3. Channel Architecture

Channel Blocks

There are four blocks (Figure 3), each containing two sets of receiver/transmitters. In the following discussion, the description applies to Block A which contains channels a and b. However, the same information applies to all channel blocks.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the Octal UART.

Interrupt Control

A single interrupt output per DUART (INTRN) is provided which is asserted on occurrence of any of the following internal events:

–Transmit holding register ready for each channel

–Receive holding register ready or FIFO full for each channel

–Change in break received status for each channel

–Counter reached terminal count

–Change in MPI input

Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR can be programmed to select only certain conditions, of the above, to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. However, the bits of the ISR are not masked by the IMR. The transmitter ready status and the receiver ready or FIFO full status can be provided on MPP1a, MPP1b, MPP2a, and MPP2b by setting OPCR[7]. these outputs are not masked by IMR.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are shown in Table 1.

Mode registers 1 and 2 are accessed via an auxiliary pointer. The pointer is set to MR1 by RESET or by issuing a reset pointer command via the command register. Any read or write of the mode register while the pointer is at MR1 switches the pointer to MR2 after the read or write. The pointer then remains at MR2 so that subsequent accesses are to MR2. To access MR1, the command 0001 of the command register must be executed.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer for each block, and two clock selectors.

Crystal Clock

The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/ CLK and X2 inputs with a minimum of external components. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. If an external clock is used instead of a crystal, X1 must be driven and X2 left floating as shown in Figure 9. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock frequency, within the limits specified in the electrical specifications, must be supplied even if the internal BRG is not used.

Philips Semiconductors Product specification

Enhanced octal universal asynchronous receiver/transmitter (Octal UART) SCC2698B

Table 1. Register Addressing

Units A and B								Units E and F							
A5	A4	A3	A2	A1	A0	READ (RDN=0)	WRITE (WRN=0)	A5	A4	A3	A2	A1	A0	READ (RDN=0)	WRITE (WRN=0)
0	0	0	0	0	0	MR1a, MR2a	MR1a, MR2a	1	0	0	0	0	0	MR1e, MR2e	MR1e, MR2e
0	0	0	0	0	1	SRa	CSRa	1	0	0	0	0	1	SRe	CSRe
0	0	0	0	1	0	BRG Test[2]	CRa	1	0	0	0	1	0	Reserved[1]	CRe
0	0	0	0	1	1	RHRa	THRa	1	0	0	0	1	1	RHRe	THRe
0	0	0	1	0	0	IPCRA	ACRA	1	0	0	1	0	0	IPCRC	ACRC
0	0	0	1	0	1	ISRA	IMRA	1	0	0	1	0	1	ISRC	IMRC
0	0	0	1	1	0	CTUA	CTPUA	1	0	0	1	1	0	CTUC	CTPUC
0	0	0	1	1	1	CTLA	CTPLA	1	0	0	1	1	1	CTLC	CTPLC
0	0	1	0	0	0	MR1b, MR2b	MR1b, MR2b	1	0	1	0	0	0	MR1f, MR2f	MR1f, MR2f
0	0	1	0	0	1	SRb	CSRb	1	0	1	0	0	1	SRf	CSRf
0	0	1	0	1	0	1X/16X Test[2]	CRb	1	0	1	0	1	0	Reserved[1]	CRf
0	0	1	0	1	1	RHRb	THRb	1	0	1	0	1	1	RHRf	THRf
0	0	1	1	0	0	Reserved[1]	Reserved[1]	1	0	1	1	0	0	Reserved[1]	Reserved[1]
0	0	1	1	0	1	Input port A	OPCRA	1	0	1	1	0	1	Input port C	OPCRC
0	0	1	1	1	0	Start C/T A	Reserved[1]	1	0	1	1	1	0	Start C/T C	Reserved[1]
0	0	1	1	1	1	Stop C/T A	Reserved[1]	1	0	1	1	1	1	Stop C/T C	Reserved[1]
Units C and D								Units G and H							
0	1	0	0	0	0	MR1c, MR2c	MR1c, MR2c	1	1	0	0	0	0	MR1g, MR2g	MR1g, MR2g
0	1	0	0	0	1	SRc	CSRc	1	1	0	0	0	1	SRg	CSRg
0	1	0	0	1	0	Reserved[1]	CRc	1	1	0	0	1	0	Reserved[1]	CRg
0	1	0	0	1	1	RHRc	THRc	1	1	0	0	1	1	RHRg	THRg
0	1	0	1	0	0	IPCRB	ACRB	1	1	0	1	0	0	IPCRD	ACRD
0	1	0	1	0	1	ISRB	IMRB	1	1	0	1	0	1	ISRD	IMRD
0	1	0	1	1	0	CTUB	CTPUB	1	1	0	1	1	0	CTUD	CTPUD
0	1	0	1	1	1	CTLB	CTPLB	1	1	0	1	1	1	CTLD	CTPLD
0	1	1	0	0	0	MR1d, MR2d	MR1d, MR2d	1	1	1	0	0	0	MR1h, MR2h	MR1h, MR2h
0	1	1	0	0	1	SRd	CSRd	1	1	1	0	0	1	SRh	CSRh
0	1	1	0	1	0	Reserved[1]	CRd	1	1	1	0	1	0	Reserved[1]	CRh
0	1	1	0	1	1	RHRd	THRd	1	1	1	0	1	1	RHRh	THRh
0	1	1	1	0	0	Reserved[1]	Reserved[1]	1	1	1	1	0	0	Reserved[1]	Reserved[1]
0	1	1	1	0	1	Input port B	OPCRB	1	1	1	1	0	1	Input port D	OPCRD
0	1	1	1	1	0	Start C/T B	Reserved[1]	1	1	1	1	1	0	Start C/T D	Reserved[1]
0	1	1	1	1	1	Stop C/T B	Reserved[1]	1	1	1	1	1	1	Stop C/T D	Reserved[1]

NOTE:

1. Reserved registers should never be read during normal operation since they are reserved for internal diagnostics.

ACR = Auxiliary control register
CR = Command register
CSR = Clock select register
CTL = Counter/timer lower
CTPL = Counter/timer preset lower register
CTU = Counter/timer upper
CTPU = Counter/timer preset upper register
MR = Mode register
SR = Status Register
THR = Tx holding register
RHR = Rx holding register
IPCR = Input port change register
ISR = Interrupt status register
IMR = Interrupt mask register
OPCR = Output port configuration register

2. See Table 5 for BRG Test frequencies in this data sheet, and *"Extended baud rates for SCN2681, SCN68681, SCC2691, SCC2692, SCC68681 and SCC2698B"* Philips Semiconductors ICs for Data Communications, IC-19, 1994.

BRG

The baud rate generator operates from the oscillator or external clock input and is capable of generating 26 commonly used data communications baud rates ranging from 50 to 115.2K baud. Thirteen of these are available simultaneously for use by the receiver and transmitter. Eight are fixed, and one of two sets of five can be selected by programming ACR[7]. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The clock selectors allow the independent selection, by the receiver and transmitter, of any of these baud rates or an external timing signal.

Counter–Timer

The four Counter/Timers are programmable 16 bit dividers that are used for generating miscellaneous clocks or generating timeout

periods. These clocks may be used by any or all of the receivers and transmitters in the OCTART or may be directed to an I/O pin for miscellaneous use.

Counter/Timer programming

The counter timer is a 16–bit programmable divider that operates in one of three modes: counter, timer, and time out.

- Timer mode generates a square wave.
- Counter mode generates a time delay.
- Time out mode counts time between received characters.

The C/T uses the numbers loaded into the Counter/Timer Lower Register (CTPL) and the Counter/Timer Upper Register (CTPU) as its divisor. The counter timer is controlled with six commands: Start/Stop C/T, Read/Write Counter/Timer lower register and Read/Write Counter/Timer upper register. These commands have slight differences depending on the mode of operation. Please see the detail of the commands under the CTPL/CTPU register descriptions.

Baud Rate Generation

When these timers are selected as baud rates for receiver or transmitter via the Clock Select register their output will be configured as a 16x clock. Therefore one needs to program the timers to generate a clock 16 times faster than the data rate. The formula for calculating 'n', the number loaded to the CTPU and CTPL registers, based on a particular input clock frequency is shown below.

For the timer mode the formula is as follows:

$$n = \frac{\text{Clock input frequency}}{2 \quad 16 \quad \text{Baud rate desired}}$$

NOTE: 'n' may not assume values of 0 and 1.

The frequency generated from the above formula will be at a rate 16 times faster than the desired baud rate. The transmitter and receiver state machines include divide by 16 circuits, which provide the final frequency and provide various timing edges used in the qualifying the serial data bit stream. Often this division will result in a non–integer value: 26.3 for example. One may only program integer numbers to a digital divider. There for 26 would be chosen. If 26.7 were the result of the division then 27 would be chosen. This gives a baud rate error of 0.3/26.3 or 0.3/26.7 that yields a percentage error of 1.14% or 1.12% respectively, well within the ability of the asynchronous mode of operation. Higher input frequency to the counter reduces the error effect of the fractional division

One should be cautious about the assumed benign effects of small errors since the other receiver or transmitter with which one is communicating may also have a small error in the precise baud rate. In a "clean" communications environment using one start bit, eight data bits and one stop bit the total difference allowed between the transmitter and receiver frequency is approximately 4.6%. Less than eight data bits will increase this percentage.

Receiver and Transmitter

The Octal UART has eight full-duplex asynchronous receiver/transmitters. The operating frequency for the receiver and transmitter can be selected independently from the baud rate generator, the counter/timer, or from an external input.

Registers associated with the communications channel are the mode registers (MR1 and MR2), the clock select register (CSR), the command register (CR), the status register (SR), the transmit holding register (THR), and the receive holding register (RHR).

Transmitter

The SCC2698 is conditioned to transmit data when the transmitter is enabled through the command register. The SCC2698 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at MPO or MPP1 and INTRN. When the transmitter is initially enabled the TxRDY and TxEMT bits will be set in the status register. When a character is loaded to the transmit FIFO the TxEMT bit will be reset. The TxEMT will not set until: 1) the transmit FIFO is empty and the transmit shift register has finished transmitting the stop bit of the last character written to the transmit FIFO, or 2) the transmitter is disabled and then re–enabled. The TxRDY bit is set whenever the transmitter is enabled and the TxFIFO is not full. Data is transferred from the holding register to transmit shift register when it is idle or has completed transmission of the previous character. Characters cannot be loaded into the TxFIFO while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the TxFIFO, the TxD output remains High and the TxEMT bit in the Status Register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the TxFIFO.

If the transmitter is disabled, it continues operating until the character currently being transmitted and any characters in the TxFIFO including parity and stop bit(s) have been completed.

The transmitter can be forced to send a continuous Low condition by issuing a send break command from the command register. The transmitter output is returned to the normal high with a stop break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation.

If CTS option is enabled (MR2[4] = 1), the CTSN input at MPI0 must be Low in order for the character to be transmitted. The transmitter will check the state of the CTS input at the beginning of each character transmitted. If it is found to be High, the transmitter will delay the transmission of any following characters until the CTS has returned to the low state. CTS going high during the serialization of a character will not affect that character.

Transmitter "RS485 turnaround"

The transmitter can also control the RTSN outputs, MPO via MR2[5]. When this mode of operation is set, the meaning of the MPO signal will usually be 'end of message'. See description of the MR2[5] bit for more detail.

Transmitter Flow control

The transmitter may be controlled by the CTSN input when enabled by MR2(4). The CTSN input would be connected to RTSN output of the receiver to which it is communicating. See further description in the MR 1 and MR2 register descriptions.

Receiver

The SCC2698 is conditioned to receive data when enabled through the command register. The receiver looks for a High–to–Low (mark–to–space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7–1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed. The receiver then continues to sample the input at one–bit time intervals at the theoretical center of the bit. When the proper number of data bits and parity bit (if any) have been assembled, with one half–stop bit the character will be considered complete. The least significant bit is received first. The data is then transferred to the Receive FIFO and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at MPO or MPP2 and INTRN. If the character length is less than 8 bits, the most significant unused bits in the RxFIFO are set to zero.

Receiver FIFO

The RxFIFO consists of a First–In–First–Out (FIFO) stack with a capacity of 3 characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three (3) stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RxFIFO outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

Receiver Status Bits

There are five (5) status bits that are evaluated with each byte (or character) received: received break, framing error, parity error, overrun error, and change of break. The first three are appended to each byte and stored in the RxFIFO. The last two are not necessarily related to the byte being received or a byte that is in the RxFIFO. They are however developed by the receiver state machine.

The received break, framing error, parity error and overrun error (if any) are strobed into the RxFIFO at the received character boundary, before the RxRDY status bit is set. For character mode (see below) status reporting the SR (Status Register) indicates the condition of these bits for the character that is the next to be read from the FIFO

The "received break" will always be associated with a zero byte in the RxFIFO. It means that zero character was a break character and not a zero data byte. The reception of a break condition will always set the "change of break" (see below) status bit in the Interrupt Status Register (ISR). The Change of break condition is reset by a reset error status command in the command register

Break Detection

If a break condition is detected (RxD is Low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RxFIFO and the received break bit in the SR is set to 1. The change of break bit also sets in the ISR The RxD input must return to high for two (2) clock edges of the X1 crystal clock for the receiver to recognize the end of the break condition and begin the search for a start bit.

This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.

Framing Error

A framing error occurs when a non–zero character whose parity bit (if used) and stop; bit are zero. If RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if the start bit of the next character had been detected.

The parity error indicates that the receiver–generated parity was not the same as that sent by the transmitter.

The framing, parity and received break status bits are reset when the associated data byte is read from the RxFIFO since these "error" conditions are attached to the byte that has the error

Overrun Error

The overrun error occurs when the RxFIFO is full, the receiver shift register is full, and another start bit is detected. At this moment the receiver has 4 valid characters and the start bit of the 5th has been seen. At this point the host has approximately 6/16–bit time to read a byte from the RxFIFO or the overrun condition will be set. The 5th character then overruns the 4th and the 6th the 5th and so on until an open position in the RxFIFO is seen. ("seen" meaning at least one byte was read from the RxFIFO.)

Overrun is cleared by a use of the "error reset" command in the command register.

The fundamental meaning of the **overrun** is that data has been lost. Data in the RxFIFO remains valid. The receiver will begin placing characters in the RxFIFO as soon as a position becomes vacant.

Note: Precaution must be taken when reading an overrun FIFO. There will be 3 valid characters in the receiver FIFO. There will be one character in the receiver shift register. However it will NOT be known if more than one "over–running" character has been received since the overrun bit was set. The 4th character is received and read as valid but it will not be known how many characters were lost between the two characters of the 3rd and 4th reads of the RxFIFO

The "Change of break" means that either a break has been detected or that the break condition has been cleared. This bit is available in the ISR. The break change bit being set in the ISR and the received break bit being set in the SR will signal the beginning of a break. At the termination of the break condition only the change of break in the ISR will be set. After the break condition is detected the termination of the break will only be recognized when the RxD input has returned to the high state for **two** successive edges of the 1x clock; 1/2 to 1 bit time (see above).

The receiver is disabled by reset or via CR commands. A disabled receiver will not interrupt the host CPU under any circumstance in the **normal** mode of operation. If the receiver is in the multi–drop or special mode, it will be partially enabled and thus may cause an interrupt. Refer to section on Wake–Up and the register description for MR1 for more information.

Receiver Status Modes (block and character)

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character–by–character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical–OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RxFIFO is read. Therefore the status register should be read prior to reading the FIFO.

Receiver Flow Control

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re–asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed. The receiver then continues to sample the input at one–bit time intervals at the theoretical center of the bit. When the proper number of data bits and parity bit (if any) have been assembled, with one half–stop bit the character will be considered complete. The least significant bit is received first. The data is then transferred to the Receive FIFO and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at MPO or MPP2 and INTRN. If the character length is less than 8 bits, the most significant unused bits in the RxFIFO are set to zero.

Receiver FIFO

The RxFIFO consists of a First–In–First–Out (FIFO) stack with a capacity of 3 characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three (3) stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RxFIFO outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

Receiver Status Bits

There are five (5) status bits that are evaluated with each byte (or character) received: received break, framing error, parity error, overrun error, and change of break. The first three are appended to each byte and stored in the RxFIFO. The last two are not necessarily related to the byte being received or a byte that is in the RxFIFO. They are however developed by the receiver state machine.

The received break, framing error, parity error and overrun error (if any) are strobed into the RxFIFO at the received character boundary, before the RxRDY status bit is set. For character mode (see below) status reporting the SR (Status Register) indicates the condition of these bits for the character that is the next to be read from the FIFO

The "received break" will always be associated with a zero byte in the RxFIFO. It means that zero character was a break character and not a zero data byte. The reception of a break condition will always set the "change of break" (see below) status bit in the Interrupt Status Register (ISR). The Change of break condition is reset by a reset error status command in the command register

Break Detection

If a break condition is detected (RxD is Low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RxFIFO and the received break bit in the SR is set to 1. The change of break bit also sets in the ISR The RxD input must return to high for two (2) clock edges of the X1 crystal clock for the receiver to recognize the end of the break condition and begin the search for a start bit.

This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.

Framing Error

A framing error occurs when a non–zero character whose parity bit (if used) and stop; bit are zero. If RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if the start bit of the next character had been detected.

The parity error indicates that the receiver–generated parity was not the same as that sent by the transmitter.

The framing, parity and received break status bits are reset when the associated data byte is read from the RxFIFO since these "error" conditions are attached to the byte that has the error

Overrun Error

The overrun error occurs when the RxFIFO is full, the receiver shift register is full, and another start bit is detected. At this moment the receiver has 4 valid characters and the start bit of the 5th has been seen. At this point the host has approximately 6/16–bit time to read a byte from the RxFIFO or the overrun condition will be set. The 5th character then overruns the 4th and the 6th the 5th and so on until an open position in the RxFIFO is seen. ("seen" meaning at least one byte was read from the RxFIFO.)

Overrun is cleared by a use of the "error reset" command in the command register.

The fundamental meaning of the **overrun** is that data has been lost. Data in the RxFIFO remains valid. The receiver will begin placing characters in the RxFIFO as soon as a position becomes vacant.

Note: Precaution must be taken when reading an overrun FIFO. There will be 3 valid characters in the receiver FIFO. There will be one character in the receiver shift register. However it will NOT be known if more than one "over–running" character has been received since the overrun bit was set. The 4th character is received and read as valid but it will not be known how many characters were lost between the two characters of the 3rd and 4th reads of the RxFIFO

The "Change of break" means that either a break has been detected or that the break condition has been cleared. This bit is available in the ISR. The break change bit being set in the ISR and the received break bit being set in the SR will signal the beginning of a break. At the termination of the break condition only the change of break in the ISR will be set. After the break condition is detected the termination of the break will only be recognized when the RxD input has returned to the high state for **two** successive edges of the 1x clock; 1/2 to 1 bit time (see above).

The receiver is disabled by reset or via CR commands. A disabled receiver will not interrupt the host CPU under any circumstance in the **normal** mode of operation. If the receiver is in the multi–drop or special mode, it will be partially enabled and thus may cause an interrupt. Refer to section on Wake–Up and the register description for MR1 for more information.

Receiver Status Modes (block and character)

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character–by–character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical–OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RxFIFO is read. Therefore the status register should be read prior to reading the FIFO.

Receiver Flow Control

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re–asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

receiver indicating that the receiver is ready to receive data. It is also active low and is, thus, called RTSN. RTSN is on pin MPO. A receiver's RTS output will usually be connected to the CTS input of the associated transmitter. Therefore, one could say that RTS and CTS are different ends of the same wire!

MR2(4) is the bit that allows the transmitter to be controlled by the CTS pin (MPI0). When this bit is set to one AND the CTS input is driven high, the transmitter will stop sending data at the end of the present character being serialized. It is usually the RTS output of the receiver that will be connected to the transmitter's CTS input. The receiver will set RTS high when the receiver FIFO is full AND the start bit of the fourth character is sensed. Transmission then stops with four valid characters in the receiver. When MR2(4) is set to one, CTSN must be at zero for the transmitter to operate. If MR2(4) is set to zero, the MPI0 pin will have no effect on the operation of the transmitter.

MR1(7) is the bit that allows the receiver to control MPO. When MPO is controlled by the receiver, the meaning of that pin will be RTS. However, a point of confusion arises in that MPO may also be controlled by the transmitter. When the transmitter is controlling this pin, its meaning is not RTS at all. It is, rather, that the transmitter has finished sending its last data byte. Programming the MPO pin to be controlled by the receiver and the transmitter at the same time is allowed, but would usually be incompatible.

RTS can also be controlled by the commands 1000 and 1001 in the command register. RTS is expressed at the MP0 pin which is still an output port. Therefore, the state of MP0 should be set low (either by commands of the CR register or by writing to the Output Port Configuration Register) for the receiver to generate the proper RTS signal. The logic at the output is basically a NAND of the MP0 bit register and the RTS signal as generated by the receiver. When the RTS flow control is selected via the MR1(7) bit the state of the MP0 register is not changed. Terminating the use of "Flow Control" (via the MR registers) will return the MP0 pin to the control of the MP0 register.

Transmitter Disable Note

When the TxEMT bit is set the sequence of instructions: enable transmitter — load transmit holding register — disable transmitter will often result in nothing being sent. In the condition of the TxEMT being set do not issue the disable until the TxRDY bit goes active again after the character is loaded to the TxFIFO. The data is not sent if the time between the end of loading the transmit holding register and the disable command is less that 3/16 bit time in the 16x mode. One bit time in the 1x mode.

This is sometimes the condition when the RS485 automatic "turn-around" is enabled . It will also occur when only one character is to be sent and it is desired to disable the transmitter immediately after the character is loaded.

In general, when it is desired to disable the transmitter before the last character is sent AND the TxEMT bit is set in the status register be sure the TxRDY bit is active immediately before issuing the transmitter disable instruction. (TxEMT is always set if the transmitter has underrun or has just been enabled), TxRDY sets at the end of the "start bit" time. It is during the start bit that the data in the transmit holding register is transferred to the transmit shift register.

MULTI-PURPOSE INPUT PIN

The inputs to this unlatched 8-bit port for each block can be read by the CPU, by performing a read operation as shown in Table 1. A High input results in a logic one, while a Low input results in a logic zero. When the input port pins are read on the 84-pin LLCC, they will appear on the data bus in alternating pairs (i.e., DB0 = MP10a, DB1 = MPI1a, DB2 = MPI0b, DB3 = MPI1b, DB4 = MPP1a, DB5 = MPP2a, DB6 = MPP1b, DB7 = MPP2b. Although this example is shown for input port 'A', all ports will have a similar order).

The MPI pin can be programmed as an input to one of several Octal UART circuits. The function of the pin is selected by programming the appropriate control register. Change-of-state detectors are provided for MPI0 and MPI1 for each channel in each block. A High-to-Low or Low-to-High transition of the inputs lasting longer than 25 to 50µs sets the MPI change-of-state bit in the interrupt status register. The bit is cleared via a command. The change-of-state can be programmed to generate an interrupt to the CPU by setting the corresponding bit in the interrupt mask register.

The input port pulse detection circuitry uses a 38.4KHz sampling clock, derived from one of the baud rate generator taps. This produces a sampling period of slightly more than 25µs (assuming a 3.6864MHz oscillator input). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples be observed at the new logic level. As a consequence, the minimum duration of the signal change is 25µs if the transition occurs coincident with the first sample pulse. (The 50µs time refers to the condition where the change-of-state is just missed and the first change of state is not detected until after an additional 25µs.)

MULTI-PURPOSE I/O PINS

The multi-purpose pins (MPP) can be programmed as inputs or outputs using OPCR[7]. When programmed as inputs, the functions of the pins are selected by programming the appropriate control registers. When programmed as outputs, the two MPP1 pins (per block) will provide the transmitter ready (TxRDY) status for each channel and the MPP2 pins will provide the receiver ready or FIFO full (RxRDY/FFULL) status for each channel.

MULTI-PURPOSE OUTPUT PIN

This pin can be programmed to serve as a request-to-send output, the counter/timer output, the output for the 1X or 16X transmitter or receiver clocks, the TxRDY output or the RxRDY/FFULL output (see OPCR [2:0] and OPCR [6:4] – MPO Output Select).

REGISTERS

The operation of the Octal UART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. Addressing of the registers is described in Table 1.

The bit formats of the Octal UART registers are depicted in Table 2. These are shown for block A. The bit format for the other blocks is the same.

MR1 – Mode Register 1

MR1 is accessed when the MR pointer points to MR1. The pointer is set to MR1 by RESET or by a set pointer command applied via the CR. After reading or writing MR1, the pointers are set at MR2.

MR1[7] – Receiver Request-to-Send Control
This bit controls the deactivation of the RTSN output (MPO) by the receiver. This output is manually asserted and negated by commands applied via the command register. MR1[7] = 1 causes RTSN to be automatically negated upon receipt of a valid start bit if the receiver FIFO is full. RTSN is reasserted when an empty FIFO position is available. This feature can be used to prevent overrun in the receiver by using the RTSN output signal to control the CTS input of the transmitting device.

MR1[6] – Receiver Interrupt Select
This bit selects either the receiver ready status (RxRDY) or the FIFO full status (FFULL) to be used for CPU interrupts.

MR1[5] – Error Mode Select
This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break). In the character mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the block mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last reset error command was issued.

MR1[4:3] – Parity Mode Select
If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1[4:3] = 11 selects the channel to operate in the special wake-up mode.

MR1[2] – Parity Type Select
This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special 'wake-up' mode, it selects the polarity of the transmitted A/D bit.

MR1[1:0] – Bits Per Character Select
This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2 – Mode Register 2

MR2 is accessed when the channel MR pointer points to MR2, which occurs after any access to MR1. Accesses to MR2 do not change the pointer.

MR2[7:6] – Mode Select
The Octal UART can operate in one of four modes. MR2[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2[7:6] = 01 places the channel in the automatic echo mode, which automatically re-transmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is re-clocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU-to-receiver communication continues normally, but the CPU-to-transmitter link is disabled.

Two diagnostic modes can also be selected. MR2[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2[7:6] = 11. In this mode:

1. Received data is re-clocked and retransmitted on the TXD output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., the transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes; if the deselection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

Philips Semiconductors Product specification

Enhanced octal universal asynchronous receiver/transmitter (Octal UART) SCC2698B

Table 2. Register Bit Formats

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

MR1 (Mode Register 1)

RxRTS Control	RxINT Select	Error Mode*	Parity Mode	Parity Type	Bits per Character
0 = No	0 = RxRDY	0 = Char	00 = With parity	0 = Even	00 = 5
1 = Yes	1 = FFULL	1 = Block	01 = Force parity	1 = Odd	01 = 6
			10 = No parity		10 = 7
			11 = Special mode		11 = 8

NOTE: *In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.

MR2 (Mode Register 2)

Channel Mode	TxRTS Control	CTS Enable Tx	Stop Bit Length*			
00 = Normal			0 = 0.563	4 = 0.813	8 = 1.563	C = 1.813
01 = Auto-echo	0 = No	0 = No	1 = 0.625	5 = 0.875	9 = 1.625	C = 1.875
10 = Local loop	1 = Yes	1 = Yes	2 = 0.688	6 = 0.938	A = 1.688	E = 1.938
11 = Remote loop			3 = 0.750	7 = 1.000	B = 1.750	F = 2.000
NOTE: *Add 0.5 to values shown above for 0–7, if channel is programmed for 5 bits/char.						

CR (Command Register)

Miscellaneous Commands	Disable Tx	Enable Tx	Disable Rx	Enable Rx
See text	0 = No	0 = No	0 = No	0 = No
	1 = Yes	1 = Yes	1 = Yes	1 = Yes
NOTE: Access to the upper four bits of the command register should be separated by three (3) edges of the X1 clock. A disabled transmitter cannot be loaded				

SR (Status Register)

Rec'd Break*	Framing Error*	Parity Error*	Overrun Error	TxEMT	TxRDY	FFULL	RxRDY
0 = No	0 = No	0 = No	0 = No	0 = No	0 = No	0 = No	0 = No
1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes
NOTE: *These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits [7:5] from the top of the FIFO together with bits [4:0]. These bits are cleared by a reset error status command. In character mode, they must be reset when the corresponding data character is read from the FIFO. In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.							

Table 2. Register Bit Formats (Continued)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

CSR (Clock Select Register)

Receiver Clock Select	Transmitter Clock Select
See text	See text

* See Table 5 for BRG Test frequencies in this data sheet, and *"Extended baud rates for SCN2681, SCN68681, SCC2691, SCC2692, SCC68681 and SCC2698B"* Philips Semiconductors ICs for Data Communications, IC-19, 1994.

OPCR (Output Port Configuration Register) This register controls the MPP I/O pins and the MPO multi-purpose output pins.

MPP Function Select	MPOb Pin Function Select	Power-Down Mode*	MPOa Pin Function Select
0 = input	000 = RTSN	0 = Off	000 = RTSN
1 = output	001 = C/TO	1 = On	001 = C/TO
	010 = TxC (1X)		010 = TxC (1X)
	011 = TxC (16X)		011 = TxC (16X)
	100 = RxC (1X)		100 = RxC (1X)
	101 = RxC (16X)		101 = RxC (16X)
	110 = TxRDY		110 = TxRDY
	111 = RxRDY/FF		111 = RxRDY/FF
NOTE: *Only OPCR[3] in block A controls the power-down mode.			

ACR (Auxiliary Control Register)

BRG Select	Counter/Timer Mode and Source	Delta MPI1bINT	Delta MPI0bINT	Delta MPI1aINT	Delta MPI0aINT
0 = set 1 1 = set 2	See Text	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

IPCR (Input Port Change Register)

Delta MPI1b	Delta MPI0b	Delta MPI1a	Delta MPI0a	MPI1b	MPI0b	MPI1a	MPI0a
0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High

ISR (Interrupt Status Register)

MPI Port Change	Delta BREAKb	RxRDY/ FFULLb	TxRDYb	Counter Ready	Delta BREAKa	RxRDY/ FFULLa	TxRDYa
0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes

IMR (Interrupt Mask Register)

MPI Port Change INT	Delta BREAKb INT	RxRDY/ FFULLb INT	TxRDYb INT	Counter Ready INT	Delta BREAKa INT	RxRDY/ FFULLa INT	TxRDYa INT
0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

CTPU (Counter/Timer Upper Register)

C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

CTPU (Counter/Timer Lower Register)

C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

IPR (Input Port Register) MPP and MPI Pins

MPP2b	MPP1b	MPP2a	MPP1a	MPI1b	MPI0b	MPI1a	MPI0a
0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High

NOTE: When TxEMT and TxRDY bits are at one just before a write to the Transmit Holding register, a command to disable the transmitter should be delayed until the TxRDY is at one again. TxRDY will set to one at the end of the start bit time.

Philips Semiconductors Product specification

Enhanced octal universal asynchronous receiver/transmitter (Octal UART) SCC2698B

MR2[5] – Transmitter Request-to-Send Control
CAUTION: When the transmitter controls the OP pin (usually used for the RTSN signal) the meaning of the pin is not RTSN at all! Rather, it signals that the transmitter has finished the transmission (i.e., end of block).

This bit allows deactivation of the RTSN output by the transmitter. This output is manually asserted and negated by the appropriate commands issued via the command register. MR2[5] set to 1 caused the RTSN to be reset automatically one bit time after the character(s) in the transmit shift register and in the THR (if any) are completely transmitted (including the programmed number of stop bits) if a previously issued transmitter disable is pending. This feature can be used to automatically terminate the transmission as follows:

1. Program the auto-reset mode: MR2[5]=1
2. Enable transmitter, if not already enabled
3. Assert RTSN via command
4. Send message
5. Disable the transmitter after the last byte of the message is loaded to the TxFIFO. At the time the disable command is issued, be sure that the transmitter ready bit is on and the transmitter empty bit is off. If the transmitter empty bit is on (indicating the transmitter is underrun) when the disable is issued, the last byte will not be sent.
6. The last character will be transmitted and the RTSN will be reset one bit time after the last stop bit is sent.

NOTE: The transmitter is in an underrun condition when both the TxRDY and the TxEMT bits are set. This condition also exists immediately after the transmitter is enabled from the disabled or reset state. When using the above procedure with the transmitter in the underrun condition, the issuing of the transmitter disable must be delayed from the loading of a single, or last, character until the TxRDY becomes active again after the character is loaded.

MR2[4] – Clear-to-Send Control
The sate of this bit determines if the CTSN input (MPI) controls the operation of the transmitter. If this bit is 0, CTSN has no effect on the transmitter. If this bit is a 1, the transmitter checks the sate of CTSN each time it is ready to send a character. If it is asserted (Low), the character is transmitted. If it is negated (High), the TxD output remains in the marking state and the transmission is delayed until CTSN goes Low. Changes in CTSN, while a character is being transmitted do not affect the transmission of that character. This feature can be used to prevent overrun of a remote receiver.

MR2[3:0] – Stop Bit Length Select
This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1–9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1–1/16 to 2 stop bits can be programmed in increments of 1/16 bit. In all cases, the receiver only checks for a mark condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled). If an external 1X clock is used for the transmitter, MR2[3] = 0 selects one stop bit and MR2[3] = 1 selects two stop bits to be transmitted.

CSR – Clock Select Register

Table 3. Baud Rate

CSR[7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	38.4k
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	MP2 – 16X	MP2 – 16X
1 1 1 1	MP2 – 1X	MP2 – 1X

The receiver clock is always a 16X clock, except for CSR[7:4] = 1111. When MPP2 is selected as the input, MPP2a is for channel a and MPP2b is for channel b. See Table 5.

CSR[7:4] – Receiver Clock Select
When using a 3.6864MHz crystal or external clock input, this field selects the baud rate clock for the receiver as shown in Table 3.

CSR[3:0] – Transmitter Clock Select
This field selects the baud rate clock for the transmitter. The field definition is as shown in Table 3, except as follows:

CSR[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	MPP1 – 16X	MPP1 – 16X
1 1 1 1	MPP1 – 1X	MPP1 – 1X

When MPP1 is selected as the input, MPP1a is for channel a and MPP1b is for channel b.

CR – Command Register

CR is used to write commands to the Octal UART.

CR[7:4] – Miscellaneous Commands
The encoded value of this field can be used to specify a single command as follows:

NOTE: Access to the upper four bits of the command register should be separated by three (3) edges of the X1 clock.

0000	No command.
0001	Reset MR pointer. Causes the MR pointer to point to MR1.
0010	Reset receiver. Resets the receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO pointer is reset to the first location.
0011	Reset transmitter. Resets the transmitter as if a hardware reset had been applied.
0100	Reset error status. Clears the received break, parity error, framing error, and overrun error bits in the status register (SR[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared), and in block mode to clear all error status after a block of data has been received.

0101 Reset break change interrupt. Causes the break detect change bit in the interrupt status register (ISR[2 or 6]) to be cleared to zero.

0110 Start break. Forces the TxD output low (spacing). If the transmitter is empty, the start of the break condition will be delayed up to two bit times. If the transmitter is active, the break begins when transmission of the character is completed. If a character is in the THR, the start of break is delayed until that character or any others loaded after it have been transmitted (TxEMT must be true before break begins). The transmitter must be enabled to start a break

0111 Stop break. The TxD line will go high (marking) within two bit times. TxD will remain high for one bit time before the next character, if any, is transmitted.

1000 Assert RTSN. Causes the RTSN output to be asserted (Low).

1001 Negate RTSN. Causes the RTSN output to be negated (High).

1010 Set Timeout Mode On. The register in this channel will restart the C/T as each receive character is transferred from the shift register to the RHR. The C/T is placed in the counter mode, the START/STOP counter commands are disabled, the counter is stopped, and the Counter Ready Bit, ISR[3], is reset.

1011 Reserved.

1100 Disable Timeout Mode. This command returns control of the C/T to the regular START/STOP counter commands. It does not stop the counter, or clear any pending interrupts. After disabling the timeout mode, a 'Stop Counter' command should be issued.

1101 Reserved.

111x Reserved for testing.

CR[3] – Disable Transmitter
This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CR[2] – Enable Transmitter
Enables operation of the transmitter. The TxRDY status bit will be asserted.

CR[1] – Disable Receiver
This command terminates operation of the receiver immediately – a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special wake–up mode is programmed, the receiver operates even if it is disabled (see Wake-up Mode).

CR[0] – Enable Receiver
Enables operation of the receiver. If not in the special wake-up mode, this also forces the receiver into the search for start bit state.

SR – Channel Status Register

SR[7] – Received Break
This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received; further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half bit time two successive edges of the internal or external 1x clock. **This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.**

When this bit is set, the change in break bit in the ISR (ISR[6 or 2]) is set. ISR[6 or 2] is also set when the end of the break condition, as defined above, is detected. The break detect circuitry is capable of detecting breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must last until the end of the next character in order for it to be detected.

SR[6] – Framing Error (FE)
This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SR[5]– Parity Error (PE)
This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity. In special 'wake-up mode', the parity error bit stores the received A/D bit.

SR[4] – Overrun Error (OE)
This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost. This bit is cleared by a reset error status command.

SR[3] – Transmitter Empty (TxEMT)
This bit will be set when the transmitter underruns, i.e., both the transmit holding register and the transmit shift register are empty. It is set after transmission of the last stop bit of a character, If no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU, or when the transmitter is disabled.

SR[2] – Transmitter Ready (TxRDY)
This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, e.g., characters loaded in the THR while the transmitter is disabled will not be transmitted.

SR[1] – FIFO Full (FFULL)
This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the FIFO and there is no character in the receive shift register. If a character is waiting in the receive shift register because the FIFO is full, FFULL is not reset after reading the FIFO once.

SR[0] – Receiver Ready (RxRDY)
This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, and no more characters are in the FIFO.

Philips Semiconductors Product specification

Enhanced octal universal asynchronous receiver/transmitter (Octal UART) SCC2698B

OPCR – Output Port Configuration Register

OPCR[7] – MPP Function Select
When this bit is a zero, the MPP pins function as inputs, to be used as general purpose inputs or as receiver or transmitter external clock inputs. When this bit is set, the MPP pins function as outputs. MPP1 will be a TxRDY indicator, and MPP2 will be an RxRDY/FFULL indicator.

OPCR[6:4] – MPOb Output Select
This field programs the MPOb output pin to provide one of the following:

000 Request-to-send active-Low output (RTSN). This output is asserted and negated via the command register. Mode RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted out or when the receiver FIFO and receiver shift register are full using MR2[5] and MR1[7], respectively.

001 The counter/timer output. In the timer mode, this output is a square wave with a period of twice the value (in clock periods) of the contents of the CTPU and CTPL. In the counter mode, the output remains high until the terminal count is reached, at which time it goes low. The output returns to the High state when the counter is stopped by a stop counter command.

010 The 1X clock for the transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a non-synchronized 1X clock is output.

011 The 16X clock for the transmitter. This is the clock selected by CSR[3:0], and is a 1X clock if CSR[3:0] = 1111.

100 The 1X clock for the receiver, which is the clock that samples the received data. If data is not being received, a non-synchronized 1X clock is output.

101 The 16X clock for the receiver. This is the clock selected by CSR[7:4], and is a 1X clock if CSR[7:4] = 1111.

110 The transmitter register ready signal, which is the same as SR[2].

111 The receiver ready or FIFO full signal.

OPCR[3] – Power Down Mode Select
This bit, when set, selects the power-down mode. In this mode, the 2698B oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior to placing the 2698B in this mode. This bit is reset with RESET asserted. Note that this bit must be set to a logic 1 after power up. Only OPCR[3] in block A controls the power-down mode.

OPCR[2:0] – MPOa Output Select
This field programs the MPOa output pin to provide one of the same functions as described in OPCR[6:4].

ACR – Auxiliary Control Register

ACR[7] – Baud Rate Generator Set Select
This bit selects one of two sets of baud rates generated by the BRG.

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.

Set 2: 75, 110, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, 19.2k, and 38.4k baud.

The selected set of rates is available for use by the receiver and transmitter.

ACR[6:4] – Counter/Timer Mode and Clock Source Select
This field selects the operating mode of the counter/timer and its clock source (see Table 4).

The MPI1 pin available as the Counter/Timer clock source is MPI1 a,c,e, and g only.

Table 4. ACR[6:4] Operating Mode

[6:4]	Mode	Clock Source
0 0 0	Counter	MPI1a pin
0 0 1	Counter	MPI1a pin divided by 16
0 1 0	Counter	TxC–1XA clock of the transmitter
0 1 1	Counter	Crystal or MPI pin (X1/CLK) divided by 16
1 0 0	Timer	MPI1a pin
1 0 1	Timer	MPI1a pin divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or MPI pin (X1/CLK) divided by 16
NOTE: The timer mode generates a squarewave.		

ACR[3:0] – MPI1b, MPI0b, MPI1a, MPI0a Change-of-State Interrupt Enable
This field selects which bits of the input port change register (IPCR) cause the input change bit in the interrupt status register, ISR[7], to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR – Input Port Change Register

IPCR[7:4] – MPI1b, MPI0b, MPI1a, MPI0a Change-of-State
These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register. The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] – MPI1b, MPI0b, MPI1a, MPI0a Change-of-State
These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the inputs pins during the time the IPCR is read.

ISR – Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output is asserted (Low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR; the true status is provided regardless of the contents of the IMR.

ISR[7] – MPI Change-of-State
This bit is set when a change-of-state occurs at the MPI1b, MPI0b, MPI1a, MPI0a input pins. It is reset when the CPU reads the IPCR.

ISR[6] – Channel b Change in Break
This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

ISR[5] – Receiver Ready or FIFO Full Channel b
The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receiver FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read.

If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when FIFO is read and there is no character in the receiver shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

ISR[4] – Transmitter Ready Channel b
This bit is a duplicate of TxRDY (SR[2]).

ISR[3] – Counter Ready
In the counter mode of operation, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command. It is initialized to '0' when the chip is reset.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time the C/T reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the C/T.

ISR[2] – Channel a Change in Break
This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

ISR[1] – Receiver Ready or FIFO Full Channel a
The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be ready by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receiver FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when FIFO is read and there is no character in the receiver shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

ISR[0] – Transmitter Ready Channel a
This bit is a duplicate of TxRDY (SR[2]).

IMR – Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is a '1', the INTRN output is asserted (Low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask reading of the ISR.

CTPU and CTPL – Counter/Timer Registers

The CTPU and CTPL hold the eight MSBs and eight LSBs, respectively, of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTPU/CTPL registers is H'0002'. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTPU and CTPL. The waveform so generated is often used for a data clock. The formula for calculating the divisor n to load to the CTPU and CTPL for a particular 1X data clock is shown below:

$$n = \frac{\text{C T Clock Frequency}}{2 \times 16 \text{ Baud rate desired}}$$

Often this division will result in a non-integer number; 26.3, for example. One can only program integer numbers in a digital divider. Therefore, 26 would be chosen. This gives a baud rate error of 0.3/26.3 which is 1.14%; well within the ability asynchronous mode of operation.

If the value in CTPU or CTPL is changed, the current half-period will not be affected, but subsequent half-periods will be. The C/T will not be running until it receives an initial 'Start Counter' command (read at address A3–A0 = 1110). After this, while in timer mode, the C/T will run continuously. Receipt of a subsequent start counter command causes the C/T to terminate the current timing cycle and to begin a new cycle using the values in the CTPU and CTPL.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command read with A3–A0 = H'F'). The command, however, does not stop the C/T. The generated square wave is output on MPO if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded in CTPU and CTPL by the CPU. Counting begins upon receipt of a start counter command. Upon reaching the terminal count H'0000', the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains High until the terminal count is reached, at which time it goes Low. The output returns to the High state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTPU and CTPL at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower eight bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower eight bits to the upper eight bits occurs between the times that both halves of the counter is read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTPU and CTPL.

APPENDIX B: ALPHABETICAL INDEX

J

L

M

P

R

S

T

U

W