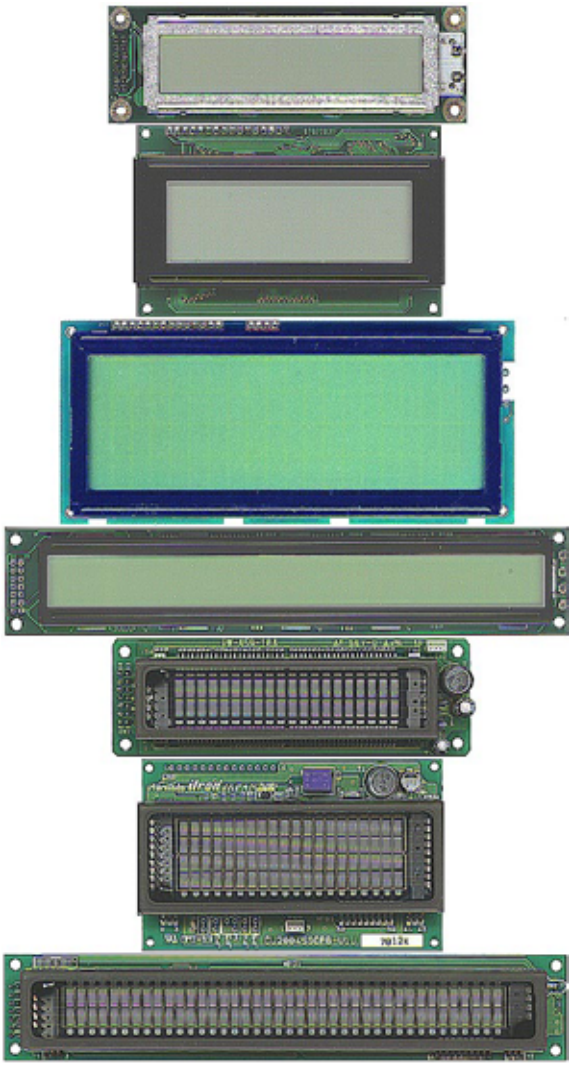
QTP 4x6

Quick Terminal Panel 4x6 Keys

USER MANUAL

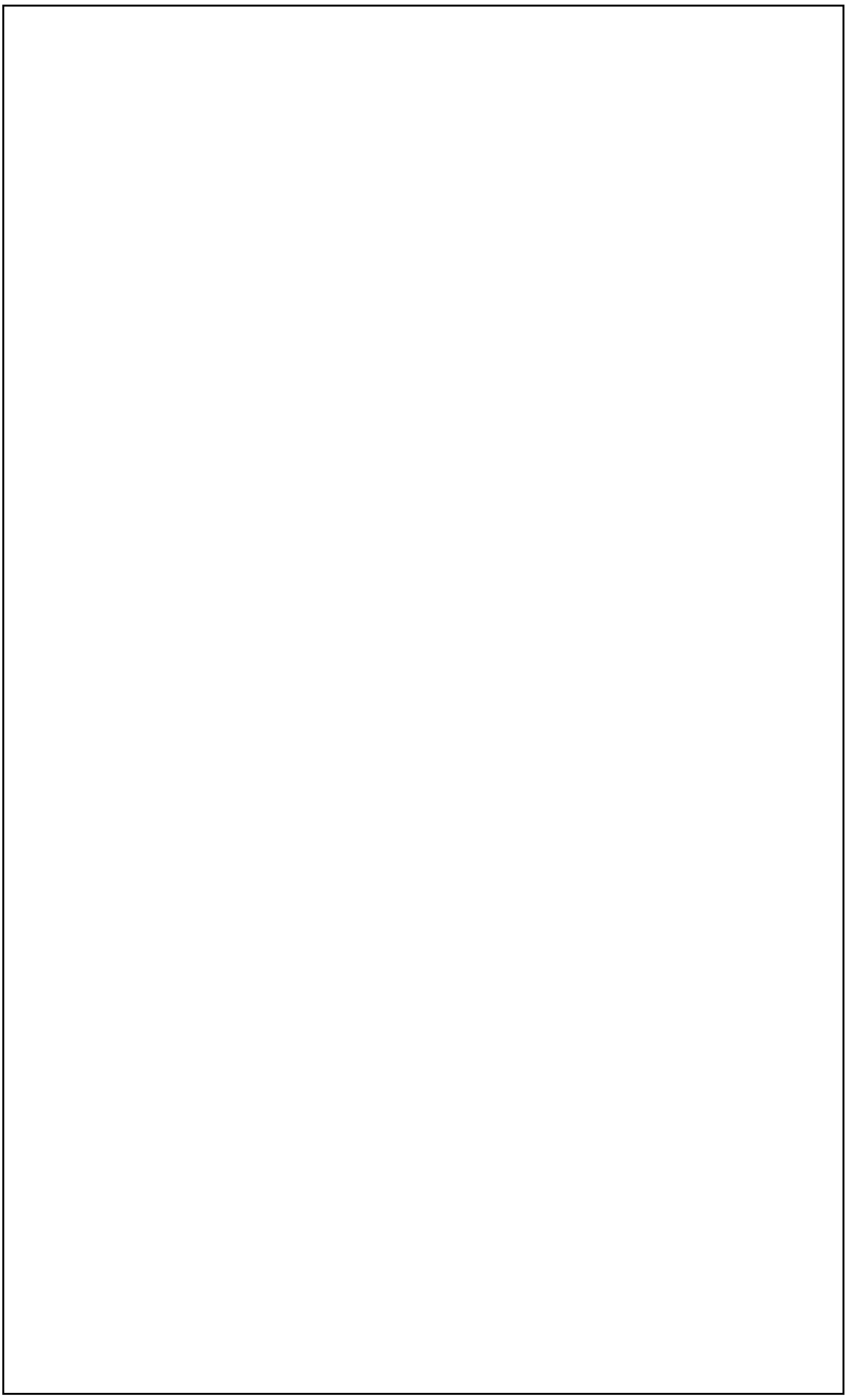


grifo®
ITALIAN TECHNOLOGY

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 (r.a.) FAX: +39 051 893.661

QTP 4x6 Rel. 5.10 *Edition 27 March 2003*

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



QTP 4x6

Quick Terminal Panel 4x6 Keys

USER MANUAL

Front size equal to the size of the display being used; remarkably low price; the **User panel** is available mounting on of these **Displays:**

QTP 4x6-C2: **LCD** display, back lighted or not, with **2** lines of **20** chars
QTP 4x6-C4: **LCD** display, back lighted or not, with **4** lines of **20** chars
QTP 4x6-C4B: **LCD** display, back lighted or not, with **4** lines of **20** chars,**bigger**
QTP 4x6-C24:**LCD** display, back lighted or not, with **2** lines of **40** chars
QTP 4x6-F2: **Fluorescent** display with **2** lines of **20** characters
QTP 4x6-F4: **Fluorescent** display with **4** lines of **20** characters
QTP 4x6-F24: **Fluorescent** display with **2** lines of **40** characters

Buzzer programmable as **BELL** or to sound with keystroke; complete management of **24 keys** trough external 4x6 matrix keyboard; **autorepeat and keyclick** functions; **E²** up to **2 Kbyte** for permanent storage of set-up, messages, key codes,etc.; memorization on **E²** and visualization, also sliding, of up to 97 messages; **RS 232** or **optional RS 422, RS 485** or **Current Loop** serial line; **communication** configurable as **Point-to-point** or **Master-Slave; local set up** for **communication** parameters (Baud Rate, Stop bits,Keyclick, etc.); up to **eight** characters pattern is **user defineable**; up to **255 different characters** in display character memory that can be displayed; possibilty to order some models with **QTP 42144** and **QTP 96192** containers; unique **+5 Vdc** power supply

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 (r.a.) FAX: +39 051 893.661

QTP 4x6 Rel. 5.10 *Edition 27 March 2003*

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

DOCUMENTATION COPYRIGHT BY grifo®, ALL RIGHTS RESERVED

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

IMPORTANT

Although all the information contained herein have been carefully verified, **grifo®** assumes no 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.

FIRMWARE VERSION

This handbook make reference to firmware version **1.3** and following ones. The validity of the information contained in this manual is subordinated to the firmware release number, so the user must always verify the correct correspondence beetween the notations. Inside the device, the firmware release number is written on the label stuck on the CPU or it can be obtained by a proper command sent through the serial line.

GENERAL INFORMATION

QTP 4x6 is a complete low cost operator panel with small overall dimension, specifically designed for industrial use and for direct mounting on automatic machinery. It is a video terminal suitable to be the direct interface between operator and machinery in any of the control, command, visualization operations which could be necessary in many civil and/or industrial applications.
QTP 4x6 is available with **alphanumeric Fluorescent** or LEDs back lite **LCD** displays, with 20 characters for 2 or 4 lines or with 40 characters for 2 lines. **QTP 4x6** is directly connected with the display, so it has an overall frontal dimension equal to the selected display used for visualization. For the mechanic fixing the user must use the onboard displays holes.
QTP 4x6 is the best choice whenever the User needs to show messages and 24 keys are enough to interact with the hardware.
QTP 4x6 gives the possibility to store in the on board serial **EEPROM** up to 97 messages. These messages can be shown on the display, also in sliding mode, simply sending to the serial port a proper sequence of commands. This way, the amount of work for the main CPU is lowered, also the messages to show must not be sent through the serial line to the panel.
It is also possible to read, through the serial line, the messages stored in the **EEPROM**. This allows the User to employ **QTP 4x6** as a small storage device where special data like hardware Set-Up, passwords, ID codes, etc. can be stored.
Some models of **QTP 4x6** can be enclosed inside a proper metallic container, named **QTP 72144** and **QTP 96192**, obtaining a compact apparatus with **IP 56** frontal protection and with remarkable mounting facilities.
The **QTP 4x6** is able to execute an entire range of display commands, including Clear Screen, Position cursor, EEPROM reading or writing, etc., with code compatibility to **ADDS ViewPoint** standard video terminal. Features of **QTP 4x6**, including options, are as follows:

- Front size equal to the size of the display being used
- Remarkably low price
- The **User panel** is available mounting on of these **Displays**
  QTP 4x6-C2: **LCD** display, back lighted or not, with **2** lines of **20** chars
  QTP 4x6-C4: **LCD** display, back lighted or not, with **4** lines of **20** chars
  QTP 4x6-C4B: **LCD** display, back lighted or not, with **4** lines of **20** chars, **bigger** format
  QTP 4x6-C24:**LCD** display, back lighted or not, with **2** lines of **40** chars
  QTP 4x6-F2: **Fluorescent** display with **2** lines of **20** characters
  QTP 4x6-F4: **Fluorescent** display with **4** lines of **20** characters
  QTP 4x6-F24: **Fluorescent** display with **2** lines of **40** characters
- **Buzzer** programmable as **BELL** or to sound with keystroke
- Complete management of **24 keys** trough external 4x6 matrix keyboard
- **Autorepeat and keyclick** functions
- **E²** up to **2 Kbyte** for permanent storage of set-up, messages, key codes,etc.
- Memorization on **E²** and visualization, also sliding, of up to 97 messages
- **RS 232** or **optional RS 422, RS 485** or **Current Loop** serial line
- **Communication** configurable as **Point-to-point** or **Master-Slave**
- **Local set up** for **communication** parameters (Baud Rate, Stop bits,Keyclick, etc.)
- Up to **eight** characters pattern is **user defineable**
- Up to **255 different characters** in display character memory that can be displayed
- Possibilty to order some models with **QTP 42144** and **QTP 96192** containers
- Unique **+5 Vdc** power supply

Here follows a description of the board's functional blocks, with an indication of the operations performed by each one.

SERIAL COMMUNICATION

The communication with remote units is by standard RS 232 serial line, but it can be optionally changed in RS 422, RS 485 or Current Loop.
Communication mode can be point-to-point or Master-Slave, employing the nineth-bit techinque; communication protocol is 8 (point-to-point) or 9 (Master-Slave) Bit, no parity, Baud Rate selectable amongst 1200, 2400, 4800, 9600, 19200 and 38400 Baud and Stop bit selectable amongst 1, 2.
All these parameters are defined through set up mode.

BUZZER

QTP 4x6 has a circuitery that generates a steady sound, based on a capacitive buzzer. The said circuitery can be activated by software through a specific comand for generating a simple beep or it can be linked to a key pressure, just to get the **keyclick** function, or it can signalize possible malfunctions.
In the last case when, after a power on, the card generates an intermittent sound and it doesn't work correctly, there is a wrong condition that must be resolved: please contact **grifo®** technicians.

KEYBOARD

QTP4x6 has an interface for a 4x6 external matrix keytboard with 24 keys made by normally- open contacts. The keys management is completely automatic with comfortable **autorepeat** feature, and there is also the possibility to change the code returned on the serial line for each key stroke, through software by using a proper command. Furthermore there is the possibility to switch on/off the key click function, i.e the buzzer activation each time a key is pressed.
Four of these keys are used to enter in the local setup function, please refer to paragraph “LOCAL SET UP”.
Thanks to the management of this simple keyboard, the **QTP 4x6** can cheaply solve the data exchange problems especially when those data are homogeneous and easy. By using industrial keys and push buttons the problems can be solved even in strong environmental applications and functionality is guaranteed in each operating conditions.

DISPLAY

QTP 4x6 is available with **Fluorescent** or backlite **LCD** alphanumeric displays with different characters number and different characters size. In detail the following displays can be mounted: Fluorescent 20x2, Fluorescent 20x4, Fluorescent 40x2, LCD 20x2, LCD 20x4, LCD 20x4 big or LCD 40x2 characters. The LEDs backlight of LCD models ensures a good visibility even when the environmental lighting changes and if it necessary the user can modify the contrast regulation by acting on a specific trimmer. Another important features of **QTP 4x6** displays is their wide viewing angle that allows a good visibility from each frontal position. Further information on each display are reported in "TECHICAL FEATURES" chapter.
The user must choose the right display (so the right **QTP 4x6** model) that is sufficient for the information to visualize and for his visibility requirements. For specific requirements on current consumption, visibility and price the card can be provided with LCD display not backlighted: for detailed information about these options and their availability, please contact directly **grifo®** offices.

EEPROM

QTP 4x6 has on board EEPROM (the size vary from 256 Bytes to 2 KBytes) for storing set up, communication protocol, keys codes, messagges, and so on. Many of the stored data have vital importance so a serial EEPROM has been choosen to obtain the best warranties on validity and maintenance of the saved information, even when power supply is not available.
It is possible to memorize up to 97 messages of 20 characters that can be first saved on EEPROM and then read or shown on the display at any moments , just giving a proper command to the terminal, with the right message identification number.
QTP 4x6 also manages scrolling messages, to show on an unique line more text than it could be visible without scrolling.
For detailed information about messages please read "COMMANDS FOR MESSAGES MANAGEMENT" paragraph.
The EEPROM size must be chosen considering the application to realize or the specific requirements of the user. Normally the card is equipped with 512 Bytes of EEPROM and the other configuration must be specified from the user, at the moment of the order, by using the following indications:

2048 Bytes EEPROM -> **.MEX** option

QTP 4x6-C2

QTP 4x6-C4

QTP 4x6-C4B

QTP 4x6-C24

QTP 4x6-F2

QTP 4x6-F4

QTP 4x6-F24

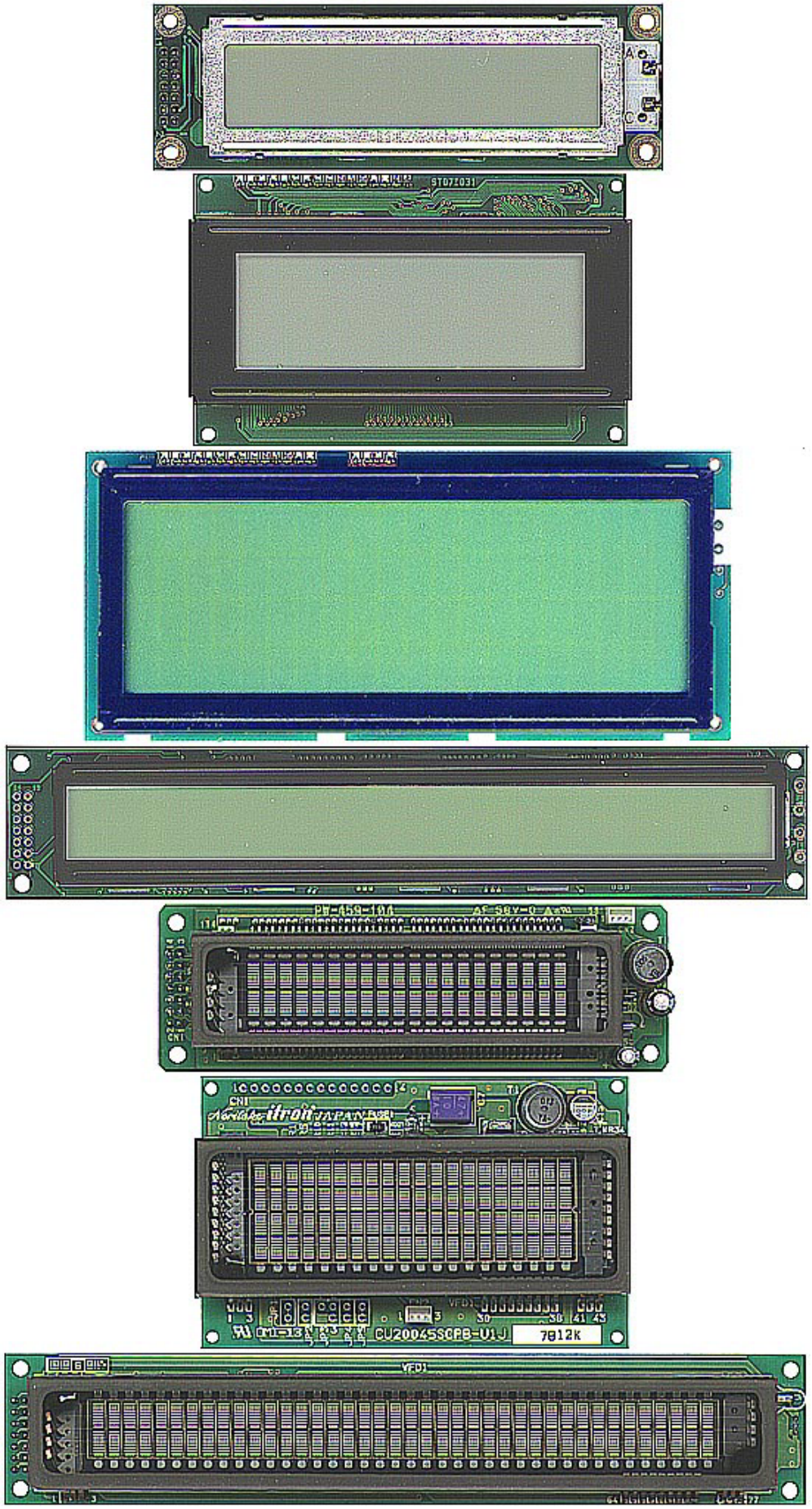


FIGURE 1: QTP 4X6 PHOTO IN ALL AVAILABLE MODELS

TECHNICAL FEATURES

GENERAL FEATURES

On board resources:
10 signals for external 4x6 matrix keyboard management
Buzzer for bell and keyclick
Full duplex RS 232, RS 422, RS 485 or current loop serial line
EEPROM for set up, messagges and so on (2K Bytes max.)
Alphanumeric display in 7 different models
Trimmer for contrast regulation of LCD display

Displays:

LCD :	20x2, 20x4, 20x4 big or 40x2, with LEDs backlite
Fluorescent:	20x2, 20x4 or 40x2

CPU: 89C4051 with 14.7456 MHz Crystal.

Communication protocol:

Baud rate:	1200, 2400, 4800, 9600, 19200, 38400
Stop bit:	1 or 2
Parity:	none
Bits x chr:	8, 9
Default:	*19200 Baud, 1 Stop, No parity, 8 Bits*

Com logic protocol: Selectable between normal and master slave *(Default: normal)*

Receive buffer dimension: 30 characters

PHYSICAL FEATURES

Size:

Display 20x2:	116 x 37 x 30 mm (W x H x D)
Display 20x4:	98 x 61 x 30 mm (W x H x D)
Display 20x4 big:	146 x 63 x 28 mm (W x H x D)
Display 40x2:	182 x 34 x 34 mm (W x H x D)

See outline dimension in APPENDIX C

Characters size:

LCD 20x2:	5 x 7 dots, 3.20 x 4.85 mm (W x H)
LCD 20x4:	5 x 7 dots, 2.95 x 4.75 mm (W x H)
LCD 20x4 big:	5 x 7 dots, 5.00 x 8.50 mm (W x H)
LCD 40x2:	5 x 7 dots, 3.20 x 5.55 mm (W x H)
Fluorescent 20x2:	5 x 7 dots, 2.40 x 4.70 mm (W x H)
Fluorescent 20x4:	5 x 7 dots, 2.40 x 4.70 mm (W x H)
Fluorescent 40x2:	5 x 7 dots, 2.30 x 4.70 mm (W x H)

Weight: 160 g max.

Mounting: Through display mounting hole (outline dimension in APPENDIX C)

Keys connection cable length: 30 cm max.

Keys autorepeat: After 500 ms and then every 100 ms

Temperature range: From 0 to 50 °C

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

Connectors: CN3: 8+8 pins AMP Mod II, 90°, Male
The female connector for CN3 can be directly ordered to **grifo®** with the code **CKS.AMP16** (kit composed by a female AMP Mod II 8+8 pins plus 16 contact to crimp), or to AMP dealer by using P/N 280366 and P/N 182206-2

ELECTRIC FEATURES

Power voltage: +5 Vdc ± 5%

RS 422-485 Termination Network:

pull-up resistor on positive:	**3.3 KΩ**
pull-down resistor on negative:	**3.3 KΩ**
line termination resistor:	**120 Ω**

Hereunder is listed the **QTP 4x6** power consumption referred to the different display types :

DISPLAY Model	*Consumption*
LCD 20x2 backlit: QTP 4x6-C2	200 mA
LCD 20x4 backlit: QTP 4x6-C4	150 mA
LCD 20x4 BIG backlit: QTP 4x6-C4B	180 mA
LCD 40x2 backlit: QTP 4x6-C24	190 mA
Fluorescent 20x2: QTP 4x6-F2	180 mA
Fluorescent 20x4: QTP 4x6-F4	270 mA
Fluorescent 40x2: QTP 4x6-F24	300 mA

FIGURE 2: CURRENT CONSUMPTION TABLE

Please remind that to reduce consumption, the **QTP 4x6** can be ordered also with LCD display not backlited: whenever necessary please contact directly **grifo®** for price and availability.

INSTALLATION

In this chapter there are the information for a right installation and correct use of the terminal **QTP 4x6**. In detail there are the locations and functions of each connector, of the user settable jumpers and of the trimmer. For the connectors it is described the pin outs, the meaning of the connected signals and some connection examples, that simplify and speed the installation phase.

CN3 - INTERFACE CONNECTOR

The conneceor named **CN3** is an AMP Mod II 8+8 pins, 90°, male with 2.54 mm pitch.
It must be used for all the **QTP 4x6** connections in fact it includes the power supply, the serial communication and external matrix keyboard signals. Placing of the signals has been designed to reduce interference and electrical noise and to simplify connections with other systems.
The female connector for CN3 can be directly ordered to **grifo®** (code **CKS.AMP16**) or acquired directly from AMP dealer by using P/N 280366 (female AMP Mod II 8+8 pins) and P/N 182206-2 (crimping contact).
In the following figures are described all these signals, divided according with their functionality.

POWER SUPPLY CONNECTION

The below figure shows the CN3 signals used to power supply the **QTP 4x6**:

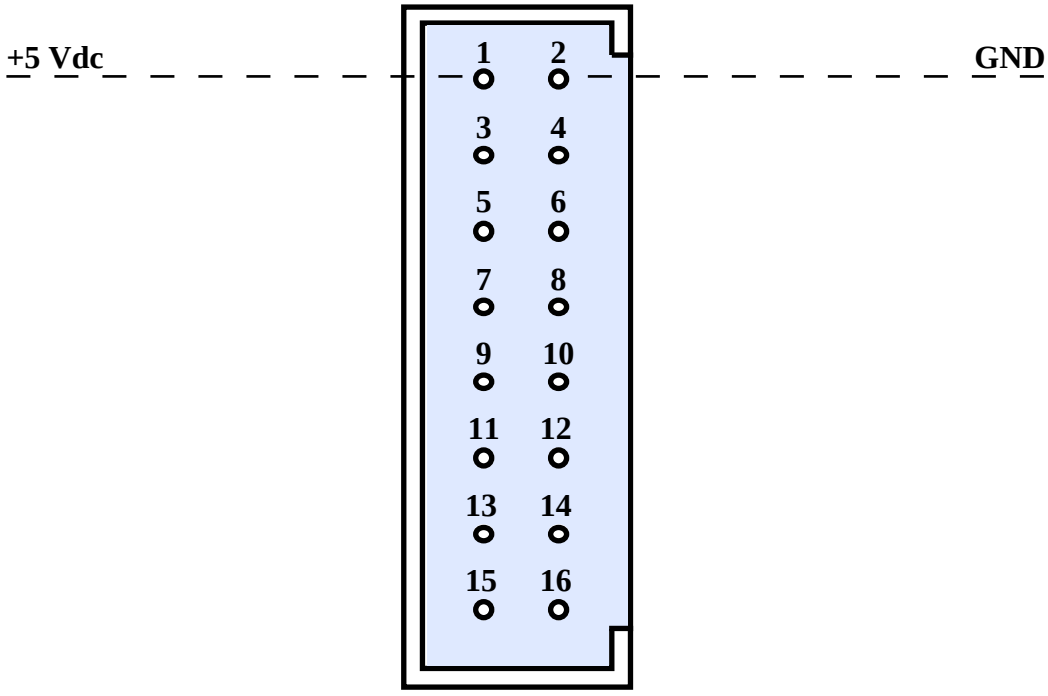


FIGURE 3: CN3 - POWER SUPPLY PINS

Signals description:

+5 Vdc = I - +5 Vdc power supply signal for on board logic.
GND = - Power supply ground signal.

RS 232 SERIAL LINE CONNECTION

The below figure shows the CN3 signals used to connect a serial line RS 232 to **QTP 4x6**. These signals follow the CCITT normative defined for each one of the available electric protocols.

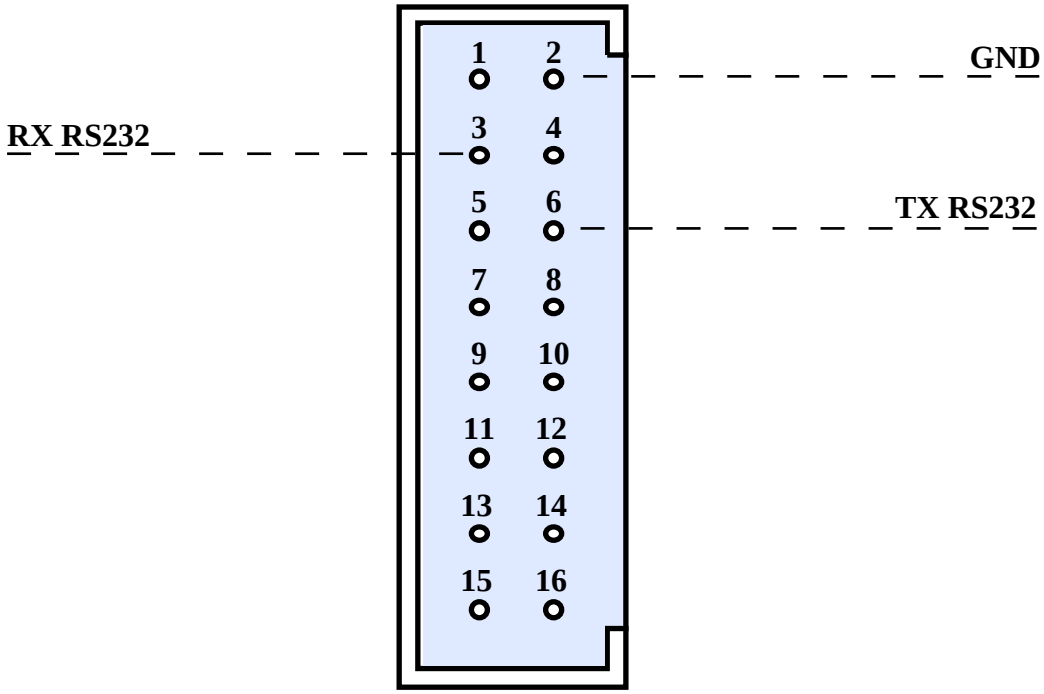


FIGURE 4: CN3 - RS 232 SERIAL LINE PINS

Signals description:

RX RS232	= I	- RS 223 serial receive data.
TX RS232	= O	- RS 223 serial transmit data.
GND	=	- Serial communication ground signal.

The Serial GND is physically connected to GND signal always on CN3 connector: the user will have to connect all the GND signals to pin 2 of CN3.
The following figure shows an RS 232 connection example diagram with a generic master unit:

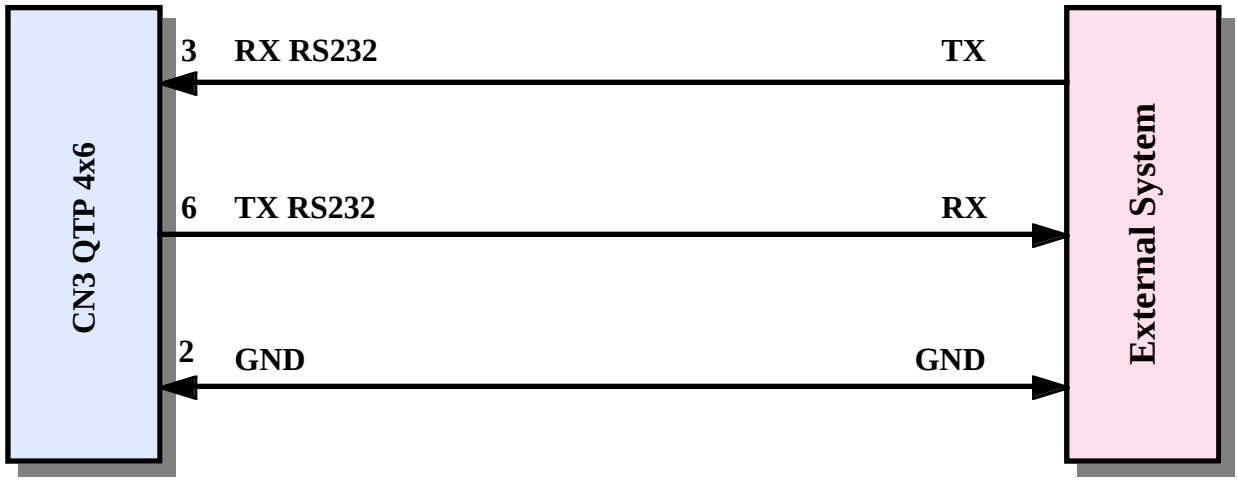


FIGURE 5: RS 232 SERIAL CONNECTION EXAMPLE

RS 422 SERIAL LINE CONNECTION

The below figure shows the CN3 signals used to connect a serial line RS 422 to **QTP 4x6**. These signals follow the CCITT normative defined for each one of the available electric protocols.

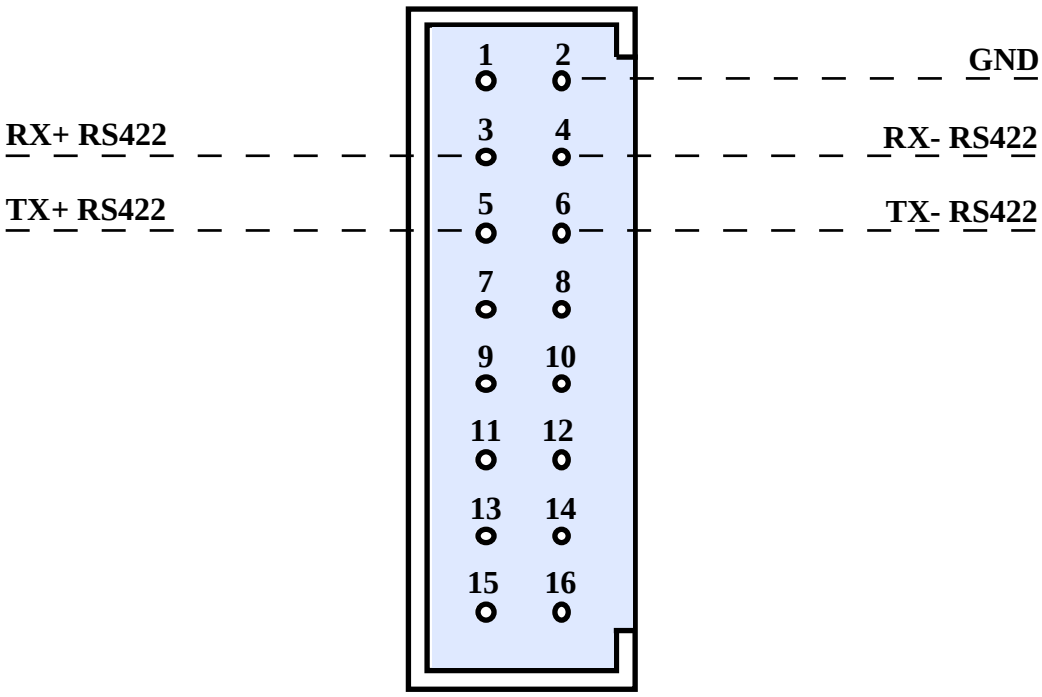


FIGURE 6: CN3 - RS 422 SERIAL LINE PINS

Signals description:

RX- RS422 = I - Receive Data Negative.
RX+ RS422 = I - Receive Data Positive.
TX- RS422 = O - Transmit Data Negative.
TX+ RS422 = O - Transmit Data Positive.
GND = - Ground signal.

The Serial GND is physically connected to GND signal always on CN3 connector: the user will have to connect all the GND signals to pin 2 of CN3.
The following figure shows an RS 422 connection example diagram with a generic master unit.

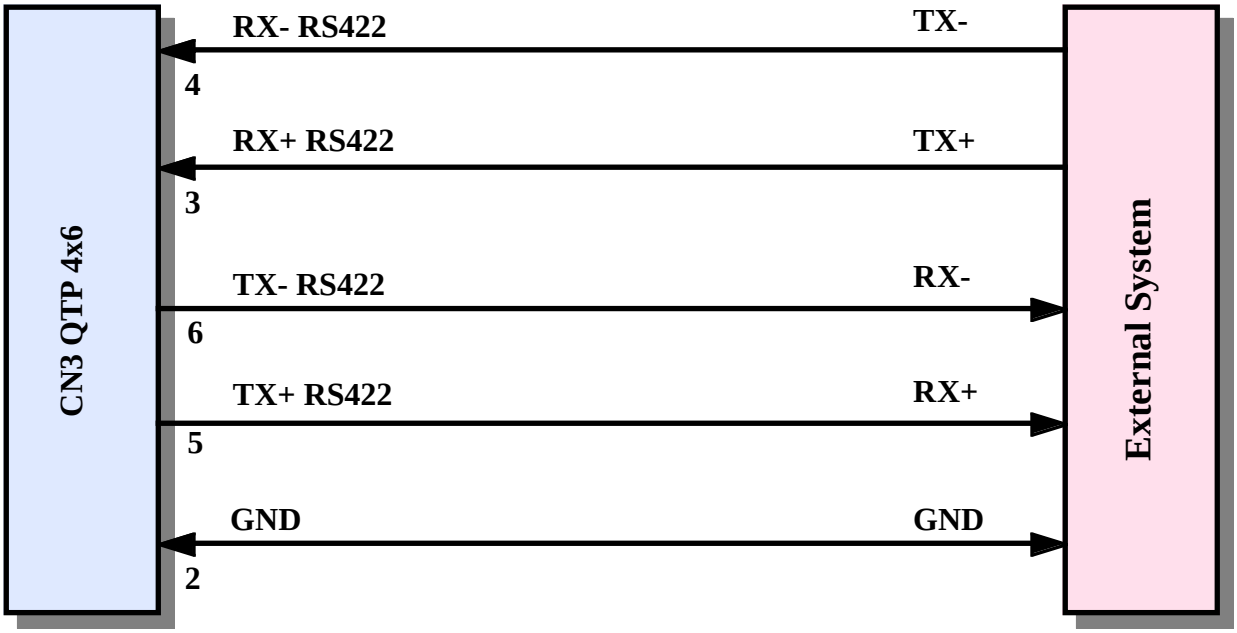


FIGURE 7: RS 422 SERIAL CONNECTION EXAMPLE

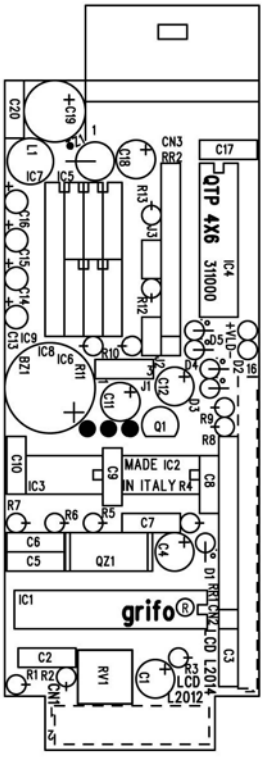


FIGURE 8: COMPONENTS MAP OF COMPONENTS SIDE

RS 485 SERIAL LINE CONNECTION

The below figure shows the CN3 signals used to connect a serial line RS 485 to **QTP 4x6**. These signals follow the CCITT normative defined for each one of the available electric protocols.

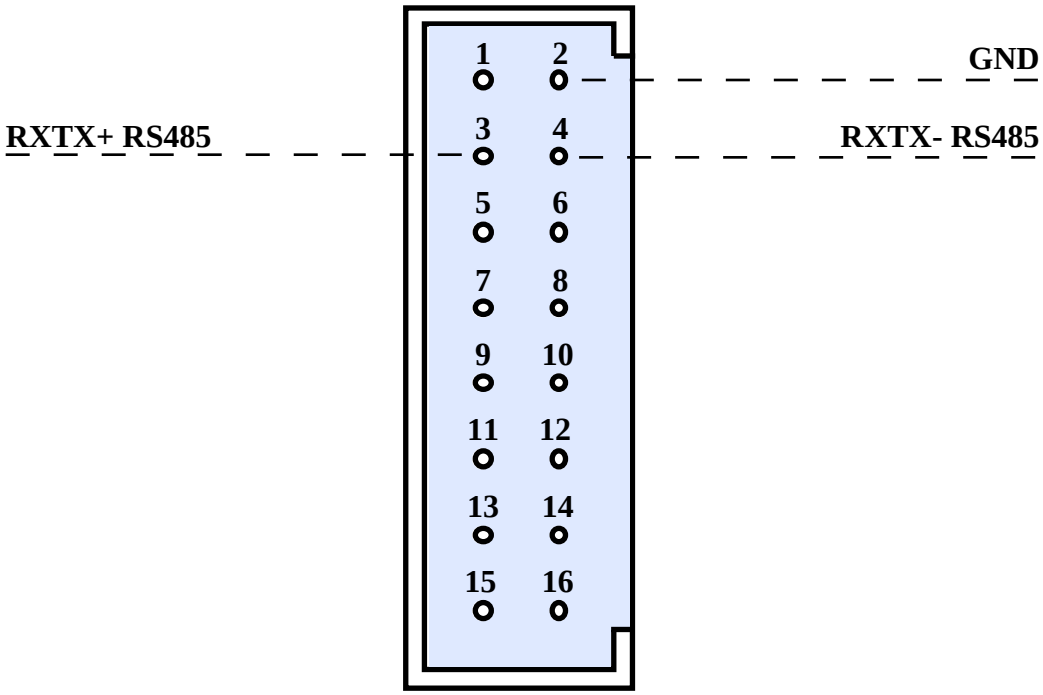


FIGURE 9: CN3 - RS 485 SERIAL LINE PINS

Signals description:

RXTX- RS485 = I/O - Receive and Transmit Data Negative.
RXTX+ RS422 = I/O - Receive and Transmit Data Positive.
GND = - Ground signal.

The Serial GND is physically connected to GND signal always on CN3 connector: the user will have to connect all the GND signals to pin 2 of CN3.
The following figure shows an RS 485 connection example diagram with a generic master unit.

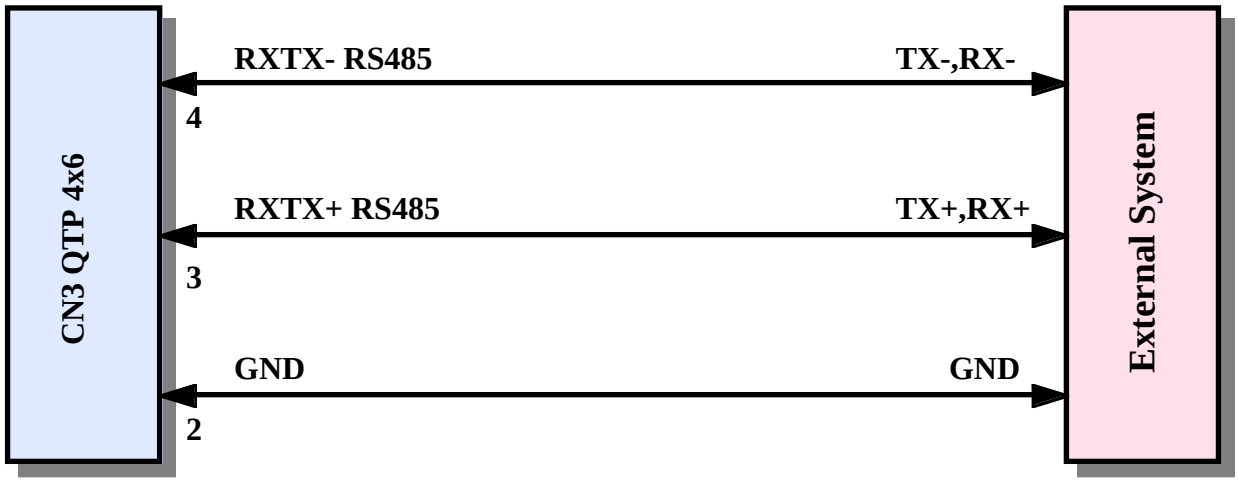


FIGURE 10: RS 485 POINT-TO-POINT SERIAL CONNECTION EXAMPLE

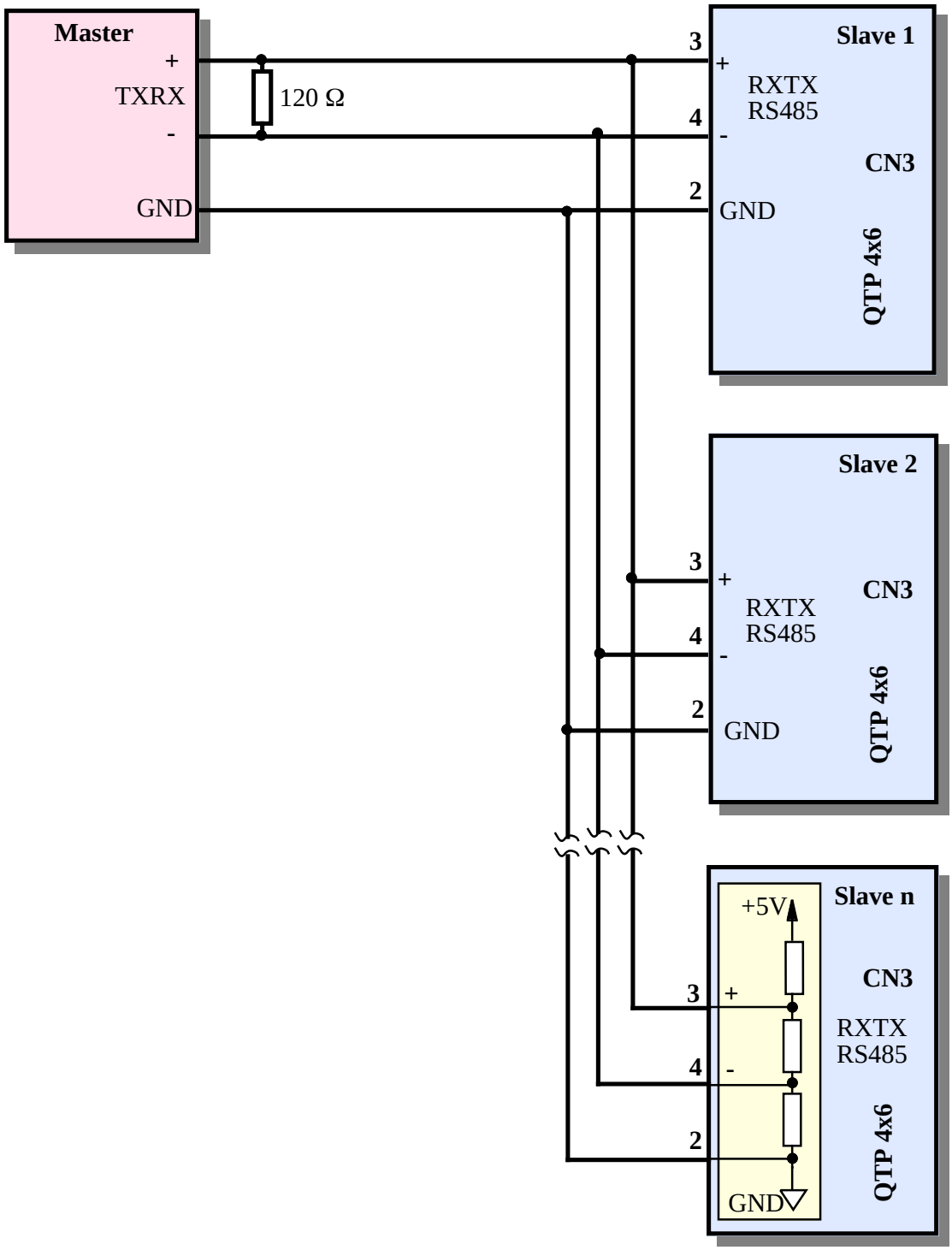


FIGURE 11: RS 485 NETWORK CONNECTION EXAMPLE

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 board of **QTP 4x6** terminal. It can be enabled or disabled through specific jumers, as explained later.
Master termination resistor must be connected if not already present (i.e., RS232-485 converters may already implement it).
For further information please refer to Data-Book TEXAS INSTRUMENTS, "*RS 422 and RS 485 Interface Cicuits*", the introduction to RS 422-485.

CURRENT LOOP SERIAL LINE CONNECTION

The below figure shows the CN3 signals used to connect a serial line current loop to **QTP 4x6**. These signals follow the CCITT normative defined for each one of the available electric protocols.

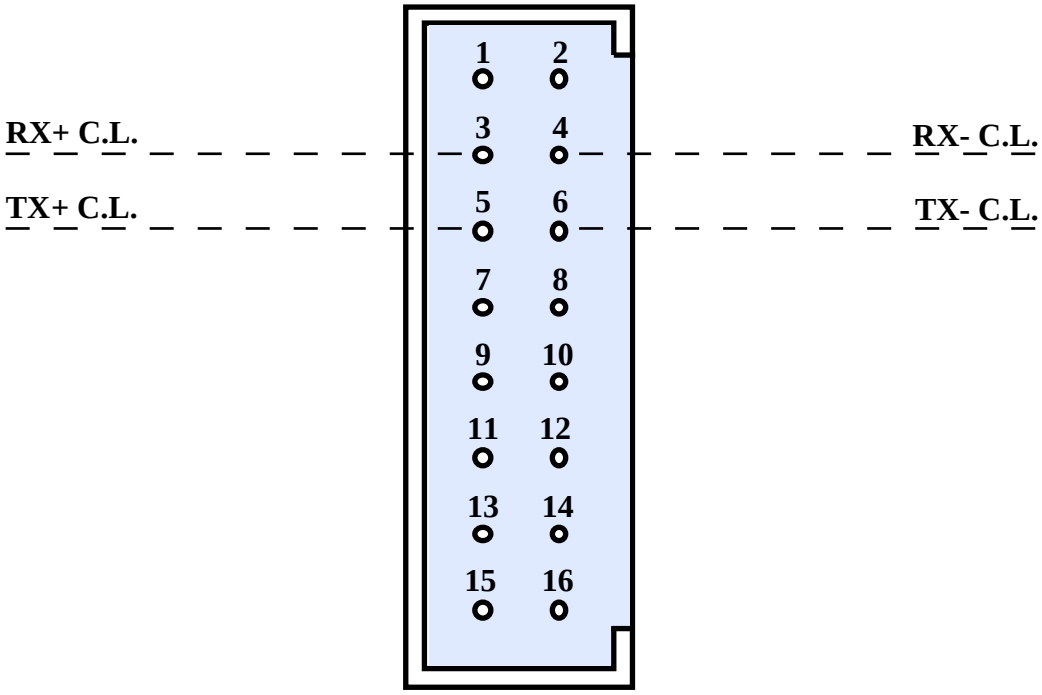


FIGURE 12: CN3 - CURRENT LOOP SERIAL LINE PINS

Signals description:

RX- C.L. = I - Receive Data Negative.
RX+ C.L. = I - Receive Data Positive.
TX- C.L. = O - Transmit Data Negative.
TX+ C.L. = O - Transmit Data Positive.

The following figure shows a current loop connection example diagram with a generic master unit.

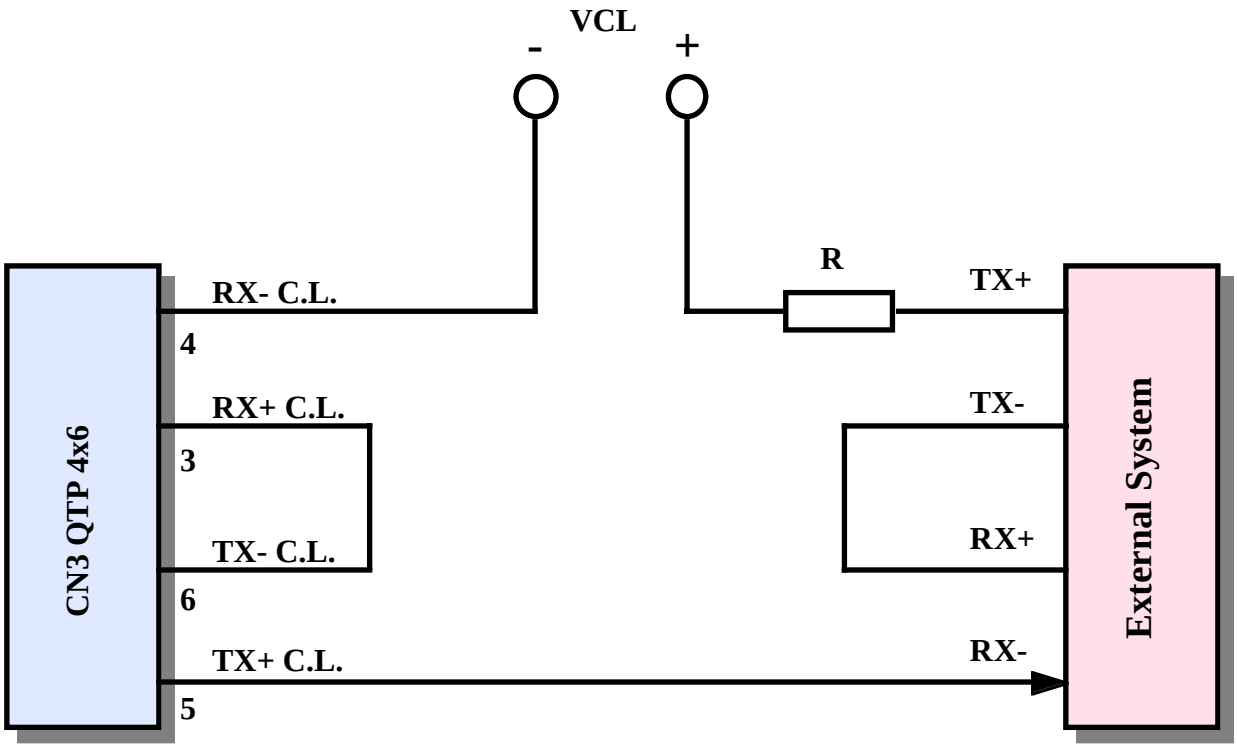


FIGURE 13: CURRENT LOOP 2-WIRES POINT-TO-POINT CONNECTION EXAMPLE

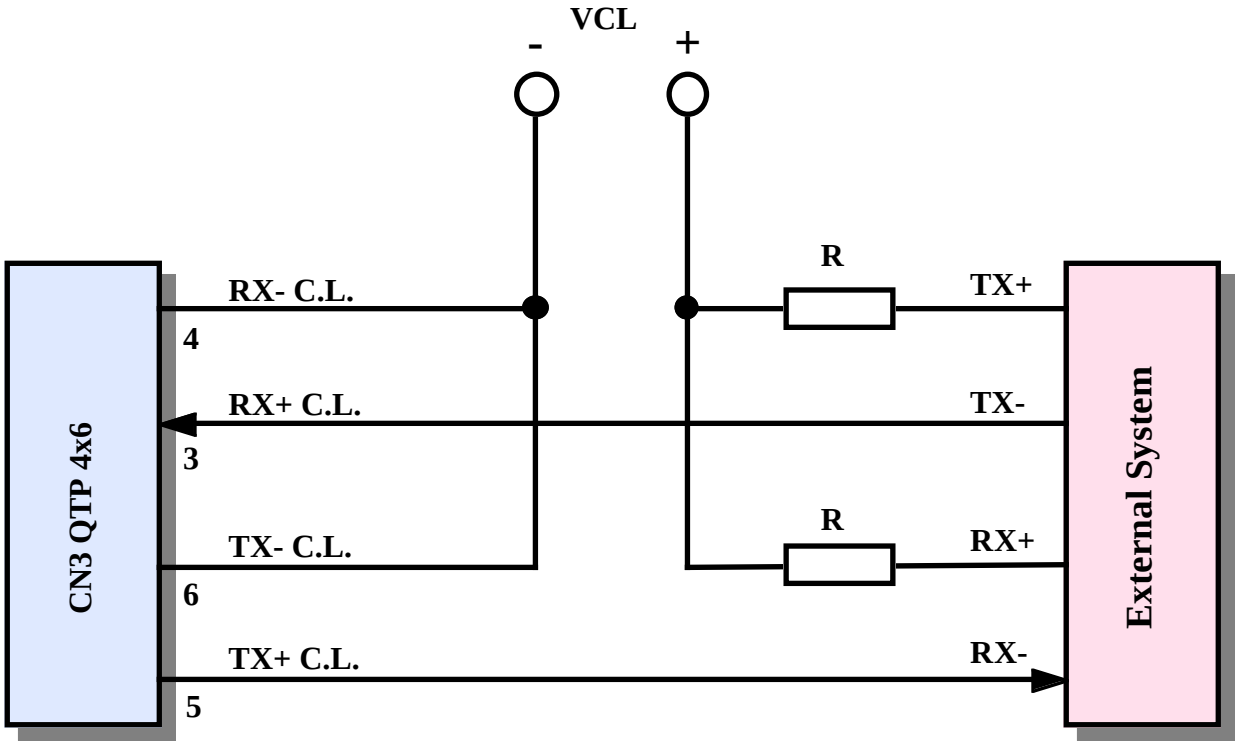


FIGURE 14: CURRENT LOOP 4-WIRES POINT-TO-POINT CONNECTION EXAMPLE

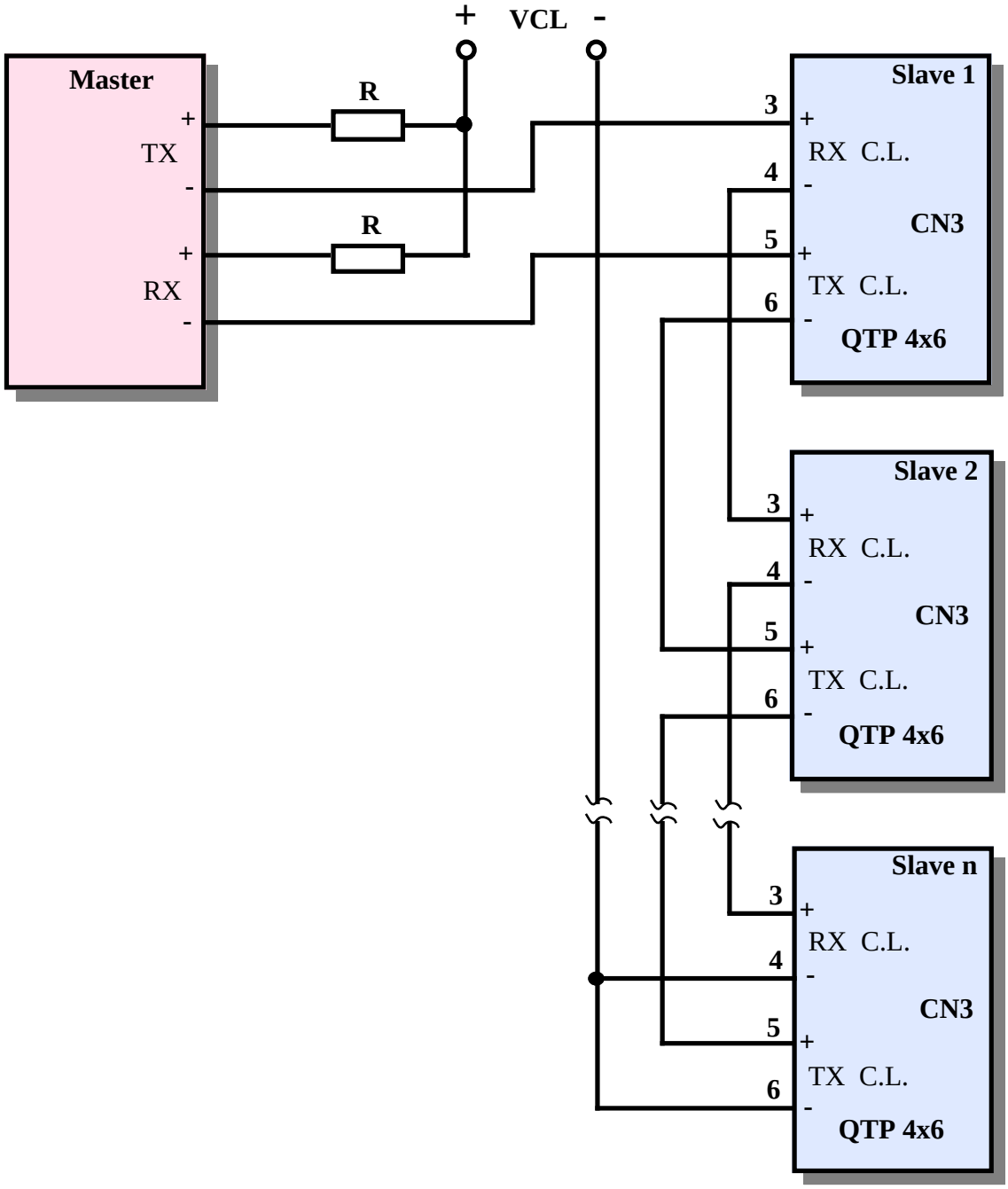


FIGURE 15: CURRENT LOOP NETWORK CONNECTION EXAMPLE

There are two possible passive Current Loop connections: 2 wires and 4 wires. These connections are shown in figures 13÷15 where it is possible to see the voltage that supply power to the loop (**VCL**) and the resistances for current limitation (**R**). The values of these components change in compliance with the number of connected devices and voltage drop on the connection cable. The choice of the values for these components must be done cosidering that:

- circulation of a **20 mA** current must be guaranteed;
- potential drop on each transmitter is about **2.35 V** with a 20 mA current;
- potential drop on each receiver is about **2.52 V** cwith a 20 mA current;
- in case of shortciruit each transmitter can dissipate at most **125 mW**;
- in case of shortciruit each receiver can dissipate at most **90 mW**.

For further info please refer to HEWLETT-PACKARD data book, (**HCPL 4100** and **4200** optocoupler devices).

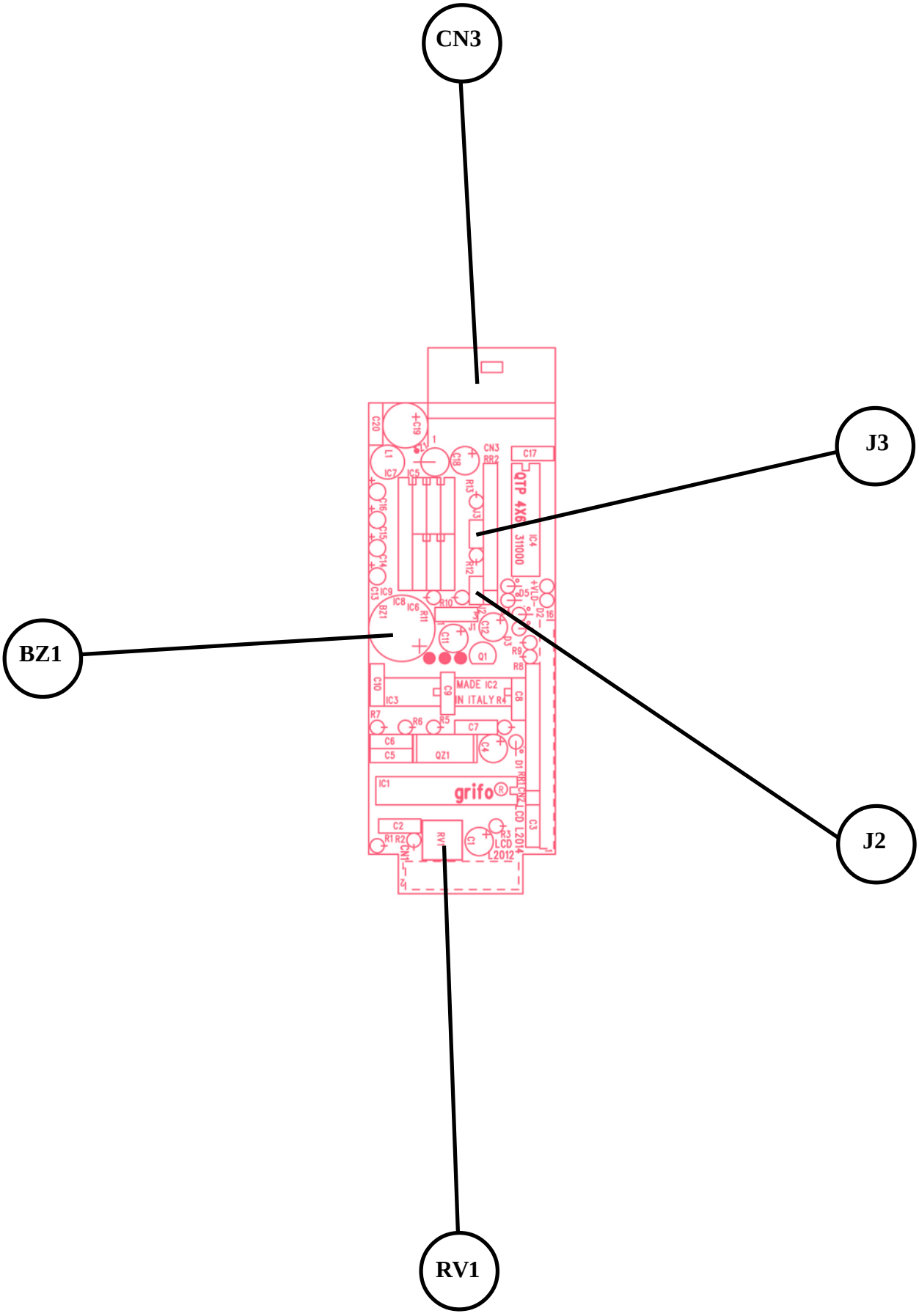


FIGURE 16: JUMPERS, CNNECTORS, TRIMMER, ETC. LOCATION

EXTERNAL MATRIX KEYBOARD CONNECTION

In the following figure the pins of CN3 that carry external keyboard signals to **QTP 4x6** are shown and described.
Signals are TTL and can be connected to any kind of normally-open contact, realizing the matrix described in figure 18; connection cable and contact matrix must be at most 30 cm long unless external favourable conditions allow its extension.

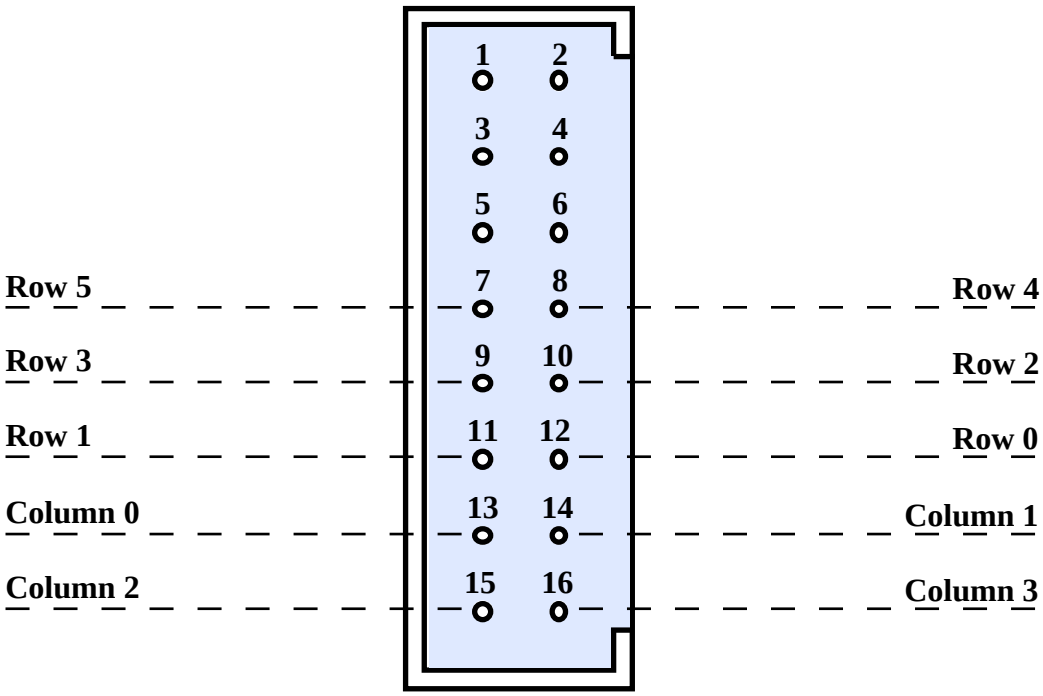


FIGURE 17: CN3 PINS TO CONNECT AN EXTERNAL MATRIX KEYBOARD TO QTP 4X6

Signals description:

Row n = O - Output signal to connect the n-th row of an external 4x6 keyboard.
Column n = I - Input signal to connect the n-th column of an external 4x6 keyboard.

External keys must be connected to allow the pressure of each key to connect the input Row n to an Output line Column n.
Following figure shows how the connection must be made and the identification number of the 24 keys.
The numbers are used in the following paragraphs about keyboard and its management to identify each key.

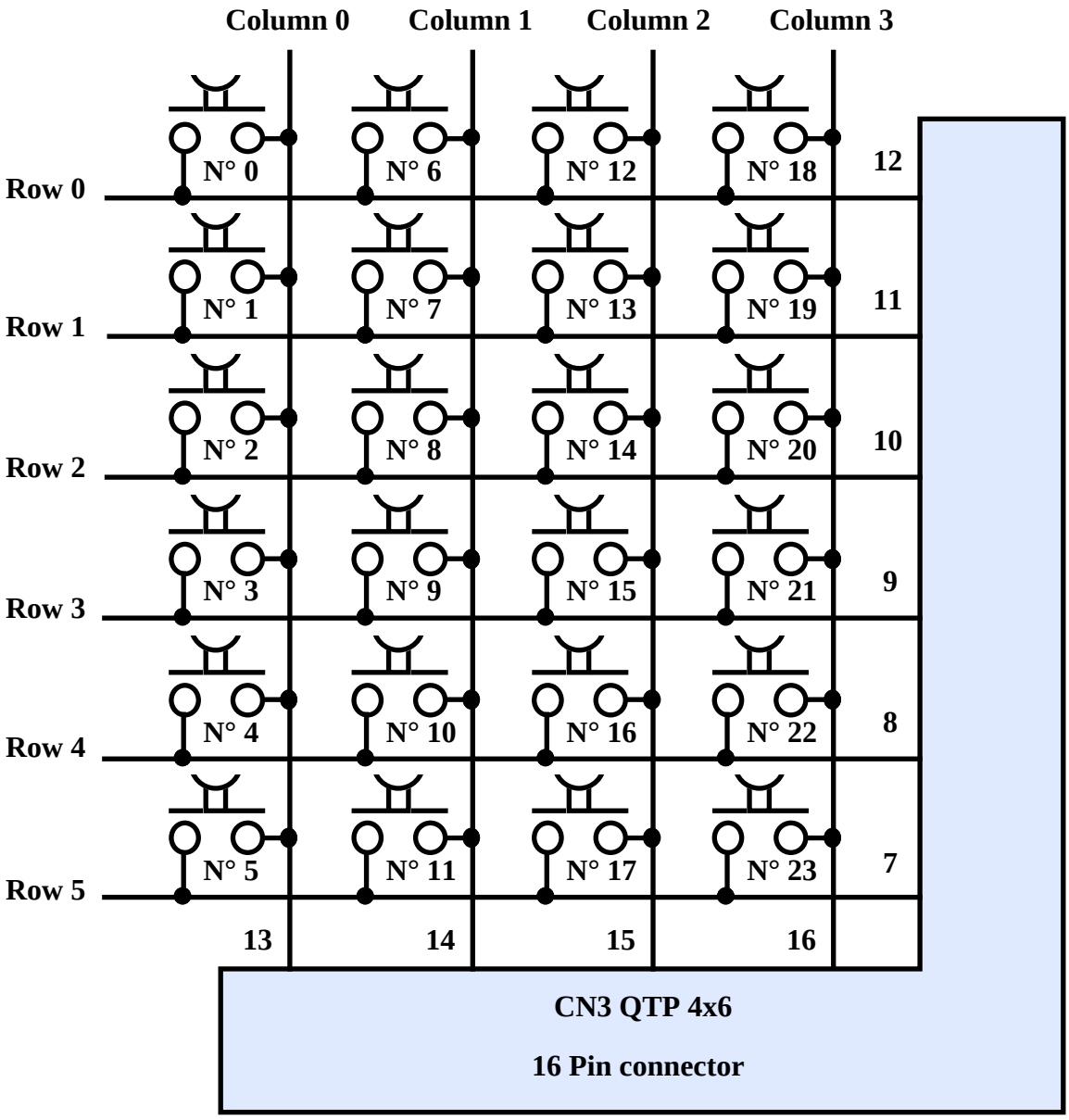


FIGURE 18: EXTERNAL MATRIX KEYBOARD CONNECTION

SERIAL LINE CONFIGURATION

QTP 4x6 is provided with one serial line that can be buffered with four different electric protocols through an hardware configuration. By default the card is delivered in RS 232, all other configuration cannot be performed by the user, who must specify them in the order, using the specific codes:

.RS422	->	optional RS 422 serial line
.RS 485	->	optional RS 485 serial line
.CLOOP	->	optional current loop serial line

QTP 4x6 is also provided with configuration jumpers; two of these, called J2 and J3, can be set by the user because they allow to connect the termination and forcing network in case of serial communication RS 422 and RS 485.
Please remark **that jumpers not described in this manual must not be changed by the user**, or the terminal functions may be faulty.
The following table describes all the right connections of these jumpers with their respective functions. To recognize the valid connections, please refer to the board printed diagram (serigraph) or to figure 8 of this manual, where the pins numeration is listed; for recognizing jumpers location, please refer to figure 16.

JUMPER	CONNECTION	PURPOSE	DEF.
J2 J3	not connected	Do not connect the termination and forcing network to RS 422, RS 485 serial line	*
	connected	Connect the termination and forcing network to RS 422, RS 485 serial line	

FIGURE 19: USER JUMPERS DESCRIPTION TABLE

The "*****" 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.

CONTRAST REGULATION TRIMMER

On **QTP 4x6** board there is a trimmer that defines the contrast on LCD displays. This trimmer, named RV1 is set by **grifo®** to obtain the best display visibility in each working conditions and normally the user must not change its position. In case of specific requirements, as external light very low or very high, RV1 can be changed by little rotation in both directions until the visibility is improved. For recognizing the location of contrast regulation trimmer, please refer to figure 16.

SOFTWARE DESCRIPTION

As already said **QTP 4x6** terminal is a complete video terminal and for this reason any characters received from serial line, if it is not a command, is shown on the display and codes of any pressed external keys, are transmitted to the control master unit. These operations are automatically performed by on board firmware that is programmed and executed by the **QTP 4x6** CPU. The on board firmware manages also a local set up which allows to set the physic communication protocol by using the keys and the display. This manual contains, in addition to the description of the different functions, a complete list of the recognized comand sequences, to be used to benefit of the main features of **QTP 4x6**. For each code or codes sequence, there is a double description i.e: the mnemonic one through the ASCII characters and the numerical one under decimal and hexadecimal form.
The said commands respect the **ADDS View Point** standard so all the sequences begin with **ESC** character corresponding to the **27** decimal code (**1B Hex**).

LOCAL SET UP

Thanks to a proper local set up mode, some parameters of communication protocol and the key click mode can be set by the user with the simple use of 4 external keys.
To enter set up mode the user must power on the **QTP 4x6** and simultaneously he must press the keys number **0** and number **20** for at least half of a second.
When the set up mode is entered, on the display appears the "**** Local Setup ****" string and with keys number 7 and number 13 the configuration parameters shall be changed as below described:

Number 7: It changes the current menu, recognized by the following messages:

"**COMMUNICATION**"	to change the communication type
"**BAUD RATE**"	to change the communication baud rate
"**STOP BIT**"	to change the stop bit number
"**KEYCLICK**"	to change the keyclick mode
"**NAME (Hex)**"	first figura of hexadecimal identification name
"**NAME (Hex)**"	second figura of hexadecimal identification name
"**EEPROM DATA**"	initializes data in EEPROM
"**SAVE and EXIT**"	to exit from set up mode

Number 13: It changes the current value of the selected menu, with the following possibilities:

COMMUNICATION	**Norm.** or **M.-S.** for normal or master-slave protocol (*def.=Norm.*)
BAUD	**38400**, **19200**, **9600**, **4800**, **2400** or **1200** baud (*def.=19200*)
STOP	**1** or **2** with normal protocol (*def.=1*)
	1 with master-slave protocol
KEYCLICK:	**ON** or **OFF** (*def.=ON*)
NAME (Hex)	Changes the figure enclosed in "><" from **0** to **F** (*def.=80H*)
EEPROM DATA	NOINIT or INIT (*def.=NOINIT*)
SAVE and EXIT	exits set up and configures **QTP 4x6** with selected parameters

When set up mode is exit, the selected parameters are saved on EEPROM and they are mantained until another local set up is executed and then terminal starts its normal functionality.
The *default* values before reported are those setted at the end of testing phase, that is the configuration the user receives.

Available options for menus BAUD RATE and STOP BIT define the pyhisical communication protocol with next two parameters unchangeable and set to 8 bits per character and no parity.
Option of remaining menus are described in the following parameters.

NOTE
Please remind that set up mode can be entered only during power up, when previously described condition are recognized in fact if key 0 and key 20 are pressed at the same time during normal operation the set up mode will not start and the code of the pressed keys will be transmitted on the serial line.
The local set up is normally executed only one time after the first installation, so the required four external keys can be connected only during this phase and thereafter the **QTP 4x6** can be used without keys as a simple display unit.

RECEIVE BUFFER

QTP 4x6 is provided of a receive buffer that simplify the management, in fact it reduces the waiting time of the connected master unit. Each received characters is immediately saved inside this buffer (**30** bytes long) and after processed at the end of the currently executed operation.
Naturally when commands that requires a long execution time (delete commands, EEPROM management commands, etc.) are continuosly received, the buffer will become full and it overflows. When overflow occours last location of the buffer is overwritten by each next received characters, and these are defenitively lost.
The master unit must stop the transmission until the **QTP 4x6** has emptied the receive buffer and it is still ready to receive other data. In practice the user must insert suitable delays between the commands transmission, to leave sufficient time to **QTP** for executing the required operations and to avoid the complete filling of the receive buffer.

KEYBOARD ACQUISITION

When **QTP 4x6** recognizes an external key pressure, if normal communcation is used it translates it and then transmits the determinated code on serial line, by using the rules described in the following paragraph.
If master slave communication is used the key code is sent only upon reception of specific request of master unit, as described in the previous paragraph.
Moreover an **auto repeat** function of the stroked key is implemented so when **QTP 4x6** recognizes the pressure on a key for a time grater than **0.5 sec**. it will start the serial transmission of its code about each **0.1 sec**. and it lasts until that specific key is released.
If the **keyclick** function is enabled when the code of the pressed key is transmitted, the on board buzzer also generates a loud beep that sonorously signalize the event to the user.

KEYS CODES

Here are the table wich shows the codes that **QTP 4x6** sends on serial line when a key is pressed; the code here is shown in deciml, hexadecimal and ASCII mnemonic format, like for command sequences:

KEY	*CODE*	*HEX CODE*	*MNEMONIC*
0	65	41	A
1	69	45	E
2	73	49	I
3	50	32	2
4	56	38	8
5	49	31	1
6	66	42	B
7	70	46	F
8	74	4A	J
9	51	33	3
10	57	39	9
11	53	35	5
12	67	43	C
13	71	47	G
14	75	4B	K
15	52	34	4
16	48	30	0
17	27	1B	ESC
18	68	44	D
19	72	48	H
20	76	4C	L
21	54	36	6
22	13	0D	CR
23	55	37	7

FIGURE 20: DEFAULT KEY CODES

Said codes are those transmitted under default condition, i.e. the configuration the user receives, but they can be comfortably reconfigured by using a specific command. This features really simplifies the management software development in fact the master unit can change the codes according with his requirements and it can also disable the keys.

COMMUNICATION MODALITIES

QTP 4x6 features two different serial communication modalities:

Norm. Normal communication uses 8 bits per character, no parity, stop bit and baud rate can be set by the user through local setup. This communication mode is suitable for connections point-to-point in RS 232, RS 422 and current loop.

M.-S. Master Slave communication uses 9 bits per character, no parity, one stop bit and baud rate can be set by the user. This communication mode is suitable for connections point-to-point (all electric protocols) on network (with protocols RS 485, RS 422 and current loop). For further information about master slave mode, please refer to next paragraph.

Local set up allows to select communication modality, as described in the specific paragraph, while electric protocol must be defined when the terminal is ordered.

MASTER SLAVE COMMUNICATION MODE

The Master Slave mode uses the 9 bits communication technique.
In addition to the 8 data bit also a 9th bit is managed as it is needed for recognizing between a call coming from the "**Master**" to any of the "**Slave**" structures and a simple info transmission between Master and the selected device.
When 9th bit is placed at 1, the data byte has to contain the name, or identifying code, of the device towards it needs to communicate, while by placing this particular bit at 0, it is possible to take out or supply info at this device.

When **QTP 4x6** is used, the identifying code must be that one set by the local set up programm on the "NAME (Hex)" entry.
When this byte is sent (with 9th bit set to 1) the **QTP 4x6** recognizes itself and it waits the string containing chars, data or commands. In this string there must only be a comand that involves the return of an information sent via serial line from **QTP 4x6** part; if there is more than one command the results of the remaining ones are ignored.
Between the transmission of a char. and the next one there must be an interval of time shorter than the **Time Out**, as elapsed this delay, the **QTP 4x6** will consider the data string ended and it will begin the answering phase. The Time Out value for each baud rate is below described:

Baud Rate	Time Out	Character transmission time
38400 Baud	550 µsec	287 µsec
19200 Baud	990 µsec	573 µsec
9600 Baud	1540 µsec	1146 µsec
4800 Baud	3080 µsec	2292 µsec
2400 Baud	6105 µsec	4584 µsec
1200 Baud	12100 µsec	9167 µsec

Master unit must wati for:

"character transission time"+"Time out"

before reaching the first character of the answering string returned by the **QTP 4x6**.
The answer consists in a byte containing the code of the pressed key (**FF Hex**, no key is pressed) or a data string related to a reading command sent in the previous request.
Please remark that answer is provided also in case the only identification name is requested, simplify the check for keys pressed or invalid commands.

To explain better the master slave protocol, here follows an example where master unit sends three commands to **QTP 4x6** (reading of version number, a string to show and a check for eventual keys pressed) with baud rate 38.4 KBaud and identification name 80H:

Master	***QTP 4x6***
Sends "Reading of version number" command, that is the characters sequence: 80H with nineth bit set to 1 1BH with nineth bit set to 0 56H with nineth bit set to 0 delay between characters lower than 550 µsec	Receives character of the command and verifies the Time Out of 550 µ sec
Waits for 837 µsec	Recognizes command sequence, executes the command and stores answer for next command
Receives one character of answer	Sends the answer, which is the code of the evental key pressed with nineth bit set to 0
Sends a string to show on the display, that is the character sequence: 80H with nineth bit set to 1 1° character of string with nineth bit set to 0 2° character of string with nineth bit set to 0 : : : : : : : delay between characters lower than 550 µsec	Receives character of the command and verifies the Time Out of 550 µ sec
Waits for 837 µsec	Recognizes command sequence and shows on the display the characters of the string
Receives three characters of answer with the version number previously requested	Trasmette risposta salvata che coincide con il numero di versione richiesto nel comando precedente, con nono bit a 0
Sends key pressed check command, that is the character sequence: 80H with nineth bit set to 1	Receives character of the command and verifies the Time Out of 550 µ sec
Waits for 837 µsec	Recognizes sequence without commands so performs no operation
Receives one character of answer corresponding to code of eventual key pressed	Sends the answer, which is the code of the evental key pressed with nineth bit set to 0

FIGURE 21: EXAMPLE OF MASTER SLAVE COMMUNICATION

Several demo programs, written in different languages, are provided with **QTP 4x6**. They implement master slave communication and can be used directly by the user or modified according to the specific needs.
When the master unit is a PC, the user can also take advantage of comfortable **DLL** libraries that allow to manage high level master slave communication, this means without having to worry about management of nineth bit, timings, eventual electric protocol converters, etc.
Also these libraries are provided with the first purchase, complete of user manual, on a CD.

NOTES:

1) To ensure right command execution, between a call and the next one it is necessary to wait for a time that is related to the number of commands sent and type of operations these latter ones involve.

2) If the Master unit cannot communicate using 9 bits, it is possible to simulate this communication mode by using the parity bit and programming its value opportunally, before any characters tranmission, according to this scheme:

If the character to transmit has EVEN number of "1" bits

*If 9th bit must be **1***	->	*Set parity to ODD*
*If 9th bit must be **0***	->	*Set parity to EVEN*

If the character to transmit has ODD number of "1" bits

*If 9th bit must be **1***	->	*Set parity to EVEN*
*If 9th bit must be **0***	->	*Set parity to ODD*

3) If the scrolling messages mode is enabled, the time between two calls, in addition to the time indicated at point **1**, must be:

QTP 4x6 with display 20x2 or 20x4:	About **12000 µsec**
QTP 4x6 with display 40x2:	About **24000 µsec**

CHARACTERS VISUALIZATION ON THE DISPLAY

QTP 4x6 shows on its display all the received characters having a code included in the range **0÷255** (**00÷FF Hex**) but the one that identifies a command sequence (27 = 1BH).
The character is visualized on the current cursor position and this latter will go to the next position; if it is placed on the last character of the display (right down corner), it will be placed on home position (left up corner).
The corrispondence between codes and displayed characters is defined by the following rules:

Codes	Characters
0 ÷ 15 (00÷0F Hex)	User defineable
16 ÷ 32 (10÷1F Hex)	Special and different according with installed display
27 (1B Hex)	Not shown
32 ÷ 127 (20÷7F Hex)	Standard ASCII
128 ÷ 255 (80÷FF Hex)	Special and different according with installed display

To allow representation of special characters, that have same codes of some one character commands, a specific command has been provided that selects the operating mode of **QTP 4x6** among the two available:

command the special characters are not displayed and the relative commands are executed;
representation the special characters are always displayed.

After a power on it is automatically selected the command mode to make immediately utilizable each functionality.
The commands composed by a sequence of two or more characters, that always start with ESC = 27 = 1BH, are anyhow interpreted and executed independently from the selected operating mode.
Each models of **QTP 4x6** has 8 user characters that can be defined and/or stored and shown on the display, as explained in the further paragraph “USER CHARACTERS COMMANDS”.
About special characters please refer to APPENDIX B and remind that it is possible to get different display models, provided of different special characters, but everything must be directly arranged with **grifo®**.

DATA STORED IN EEPROM

The on board EEPROM of **QTP 4x6** stores a set of data that can be used and/or changed through the specific commands. The menu "EEPROM DATA" of local setup allows the user to decide whether to leave these data unchanged (NOINI option) or to set them to their default value (INIT option) to restore the configuration the board find when delivered.
In detail option INIT sets EEPROM data as follows:

presence byte	->	255 (FFH)
keys codes	->	reported in table of figure 20
models of user defineable characters	->	255 (FFH)
messagges	->	255 (FFH)

exiting from local set up a string is shown on the display with a progress bar; * inform about the status of the operation. Initialization phase duration, and so the number of * printed, depends on EEPROM size and can be up to 20 seconds.
The user must choose the EEPROM initialization option very carefully, in fact all previously saved data are lost.

COMMANDS FOR CURSOR POSITION

Here follows the list of the cursor positioning commands.

CURSOR LEFT

Code: ***21***
Hex code: ***15***
Mnemonic: ***NACK***

The cursor is shifted of one position to the left without modifying the display contents. If the cursor is in Home position, it will be placed in the last position of the last row of the display.

CURSOR RIGHT

Code: ***6***
Hex code: ***6***
Mnemonic: ***ACK***

The cursor is shifted of one position to the right. If the cursor is placed in the last position of the last row, il will be moved to the Home position.

CURSOR DOWN

Code: ***10***
Hex code: ***A***
Mnemonic: ***LF***

The cursor will be moved to the line below but it will remain in the same column. If the cursor is in the last display line, it will be moved to the first display line.

CURSOR UP

Code: ***26***
Hex code: ***1A***
Mnemonic: ***SUB***

The cursor will be moved to the line above but it will remain in the same column. If the cursor is in the first display line, it will be moved to the last display line.

HOME

Code: ***1***
Hex code: ***1***
Mnemonic: ***SOH***

The cursor is moved to Home position i.e first line, first column of the display, or on the other hand the up, left corner

CARRIAGE RETURN

Code: ***13***
Hex code: ***D***
Mnemonic: ***CR***

The cursor is moved to the beginning of the line where it was located.

CARRIAGE RETURN+LINE FEED

Code: ***29***
Hex code: ***1D***
Mnemonic: ***GS***

The cursor is moved to the beginning of line above the one where it was located. If the cursor is at the last display line, it will be moved to the beginning of the first line i.e Home position.

ABSOLUTE CURSOR PLACEMENT

Code: ***27 89 r c***
Hex code: ***1B 59 r c***
Mnemonic: ***ESC Y ASCII(r) ASCII(c)***

The cursor is moved to the absolute position indicated by **r** and **c**.
These characters are the row and column values of the new desidered position referred to coordinate 0, 0 of the Home position, plus a constant offset of **32** (**20 Hex**).
If, for example, the user wants to place the cursor on the second line, third column (row 1, column 2), the following byte sequence must be sent:

27 89 33 34 or **1B 59 21 22 Hex** or **ESC Y ! "**

If row and/or column values are not compatible with the installed display, the command is ignored.

COMMANDS FOR CHARACTERS ERASURE

Below are described all the commands that deletes one or more characters from the display.

BACKSPACE

Code: ***8***
Hex code: ***8***
Mnemonic: ***BS***

This command moves the cursor one character position to the left and it erase the contents of the reached cell.
If the cursor is in Home position, it will be erased the last character of the last row of the display.

CLEAR PAGE

Code: ***12***
Hex code: ***C***
Mnemonic: ***FF***

This command clears all data on the display and it moves the cursor to Home position.

CLEAR LINE

Code: ***25***
Hex code: ***19***
Mnemonic: ***EM***

This command erases all characters displayed on the current line and it moves the cursor to the first column of the said line.

CLEAR END OF LINE

Code: ***27 75***
Hex code: ***1B 4B***
Mnemonic: ***ESC K***

This command erases all characters displayed from the current cursor position to the end of line inclusive. The cursor mantains the previous position.
If, for example, the cursor is at the beginning of a display line, the complete line will be erased.

CLEAR END OF PAGE

Code: ***27 107***
Hex code: ***1B 6B***
Mnemonic: ***ESC k***

This command erases all characters displayed from the current cursor position to the end of display inclusive. The cursor mantains the previous position.
If, for example, the cursor is at Home position, the complete display will be erased.

COMMANDS FOR CURSOR ATTRIBUTES MANAGEMENT

Below are listed the command that define the possible cursor attribute.

CURSOR OFF

Code: ***27 80***
Hex code: ***1B 50***
Mnemonic: ***ESC P***

The cursor is not active and it is not more visible.

STEADY STATIC CURSOR ON

Code: ***27 79***
Hex code: ***1B 4F***
Mnemonic: ***ESC O***

The cursor is activated so it is visible. Now it is a not blinking line placed under the current position character.

NOTE: This command is not available if **QTP 4x6-F4**, with fluorescent 20x4 display, is used.

BLINKING BLOCK CURSOR ON

Code: ***27 81***
Hex code: ***1B 51***
Mnemonic: ***ESC Q***

The cursor is enabled and it is visible. The selected cursor type is a blinking rectangular block that is alternatively visualized with the character displayed on the current cursor position.

COMMANDS FOR EEPROM

In the following paragraphs are described all the commands that manage the data saved on **QTP 4x6** on board EEPROM; there are other commands that indirectly use this memory device but they are described in next paragraphs.

REQUEST FOR EEPROM WRITING POSSIBILITY

Code: ***27 51***
Hex code: ***1B 33***
Mnemonic: ***ESC 3***

This command checks if the **QTP 4x6** is ready for writing data on its on board **EEPROM**. This command must be executed any time there are data to be saved on this type of memory.
When **QTP 4x6** receives this command, it answers with the following codes:

6	*(06 Hex)*	*(ACK)*	->	***QTP* 4x6** *ready*
21	*(15 Hex)*	*(NACK)*	->	***QTP* 4x6** *not ready*

If the **QTP 4x6** sends back the NACK code, it is not yet possible to memorize a new data on EEPROM.

WRITING OF PRESENCE BYTE

Code: ***27 33 78 byte***
Hex code: ***1B 21 4E byte***
Mnemonic: ***ESC ! N ASCII(byte)***

This command sets the card presence byte with the value indicated in the **byte** parameter that must be included in **0÷255** range.
This byte has a reserved allocation on the on board EEPROM that, once it is set with the desidered value, it allows for example, to verify that **QTP 4x6** runs correctly, or if there are some communication problems on the serial line.

NOTE: This command writes data on the on board EEPROM, so before executing it is better to check the EEPROM writing possibility through the proper command; in fact if it is not ready the command is ignored.

READING OF PRESENCE BYTE

Code: ***27 33 110***
Hex code: ***1B 21 6E***
Mnemonic: ***ESC ! n***

The **QTP 4x6** sends back on the serial line the value of its presence byte.
For example, this command can be useful to verify the presence or the correct running of the terminal.

COMMANDS FOR GENERAL FUNCTIONS

In the following paragraphs are described all the general purspose commands that manage some of the **QTP 4x6** features.

READING OF VERSION NUMBER

Code: ***27 86***
Hex code: ***1B 56***
Mnemonic: ***ESC V***

On the serial line is returned a string of 3 characters containing the program managing version that is resident and executed by **QTP 4x6**. For example with a 1.3 firmware version the following characters will be transmitted:

49 46 51 or **31 2E 33 Hex** or **1.3**

BEEP

Code: ***7***
Hex code: ***7***
Menomonic: ***BEL***

The buzzer is enabled for a time of 0.1 second.

OPERATING MODE SELECTION

Code: ***27 65 mode***
Hex code: ***1B 41 mode***
Mnemonic: ***ESC A ASCII(mode)***

It defines the operating mode for the special characters (provided of code less than 32 = 20H) and the single character commands. The selected modality is defined by **mode** value, with the following corrispondence:

0	*(00 Hex)*	->	*Command mode*
255	*(FF Hex)*	->	*Representation mode*

If **mode** value is not one of the above described, the command is ignored. Further information about operating mode are available inside “CHARACTER VISUALIZATION ON THE DISPLAY” paragraph.

COMMANDS FOR KEYBOARD MANAGEMENT

Below are described the commands that can be used to manage the external keys, connected to **QTP 4x6**. Detailed information about keys management and codes transmitted by the terminal, are available in "KEYBOARD ACQUISITION" paragraph.

KEY RECONFIGURATION

Code:	***27 55 key n. code***
Hex code:	***1B 37 key n. code***
Mnemonic:	***ESC 7 ASCII(key n.) ASCII(code)***

When the selected **key n.** is reconfigured, each time it is pressed, the card will send the new specified **code** on serial line.
The value of **key n.** to be reconfigured must be included in the range **0÷23** (**0÷17 Hex**) otherwise the command is ignored, and it will substitute the codes described in figure 18.
The **code** value can vary in the range **0÷254** (**00÷FE Hex**) as the **255** value (**FF Hex**) indicates that the key is disabled and when it will be pressed the **QTP** will not send any codes.
Figure 20 reports the default key codes and the paragraph "DATA STORED IN EEPROM" indicates how to restore these codes in case of unwanted changes.

NOTE: This command writes data on the on board EEPROM, so before executing it is better to check the EEPROM writing possibility through the proper command; in fact if it is not ready the command is ignored.

KEYCLICK ON WITHOUT MEMORIZATION

Code:	***27 53***
Hex code:	***1B 35***
Mnemonic:	***ESC 5***

The **keyclick** function is switched on so there is a sound feedback when a key is pressed. This setting is not saved inside the on board EEPROM so if the terminal is powered off and on it goes back to the previous condition, defined and saved in local set up mode.

KEYCLICK OFF WITHOUT MEMORIZATION

Code:	***27 54***
Hex code:	***1B 36***
Mnemonic:	***ESC 6***

The **keyclick** function is disabled so there is not sound feedback when a key is pressed. This setting is not saved inside the on board EEPROM so if the terminal is powered off and on it goes back to the previous condition, defined and saved in local set up mode.

KEYCLICK ON WITH MEMORIZATION

Code:	***27 33 53***
Hex code:	***1B 21 35***
Mnemonic:	***ESC ! 5***

This command enables **KeyClick** function, so there is an audible feedback when a key is pressed. This setting is stored on the on board EEPROM so if the card is turned off and on, it keeps the current condition.

NOTE
This command writes data on the on board EEPROM, so before executing it is better to check the EEPROM writing possibility through the proper command; in fact if it is not ready the command is ignored.

KEYCLICK OFF WITH MEMORIZATION

Code:	***27 33 54***
Hex code:	***1B 21 36***
Mnemonic:	***ESC ! 6***

This command disables **KeyClick** function, so there is not audible feedback when a key is pressed. This setting is stored on the on board EEPROM so if the card is turned off and on, it keeps the current condition.

NOTE
This command writes data on the on board EEPROM, so before executing it is better to check the EEPROM writing possibility through the proper command; in fact if it is not ready the command is ignored.

COMMANDS FOR USER CHARACTERS

QTP 4x6lets the user define and show up to 8 user characters; those characters can be used to represent on display special characters, pseudo graphic characters, special symbols, etc. that are not still available in the same display (please refer to table in appendix B). The user characters can be defined and saved with a pattern equal to a 5 x 8 pixels matrix, so organized:

	Pixel columns				
Pixel rows	Pat 0.4	Pat 0.3	Pat 0.2	Pat 0.1	Pat 0.0
	Pat 1.4	Pat 1.3	Pat 1.2	Pat 1.1	Pat 1.0
	Pat 2.4	Pat 2.3	Pat 2.2	Pat 2.1	Pat 2.0
	Pat 3.4	Pat 3.3	Pat 3.2	Pat 3.1	Pat 3.0
	Pat 4.4	Pat 4.3	Pat 4.2	Pat 4.1	Pat 4.0
	Pat 5.4	Pat 5.3	Pat 5.2	Pat 5.1	Pat 5.0
	Pat 6.4	Pat 6.3	Pat 6.2	Pat 6.1	Pat 6.0
	Pat 7.4	Pat 7.3	Pat 7.2	Pat 7.1	Pat 7.0

FIGURE 23: USER CHARACTERS MODEL

The user characters representation is really simple in fact it is sufficient to send the proper code (0 to 7 or 8 to 15) with a possible previous setting of representation mode, through OPERATING MODE SELECTION command.
When the user character are saved their pixels patterns are written on EEPROM and then they are reloaded on display any time the terminal is powered on.

NOTE: On **QTP 4x6-F2** and **QTP 4x6-F24** the value of Pat 7.0 pixel defines the status of all the five pixels Pat 7.4÷Pat 7.0, or in other words it defines the status of underline attribute of the defined character.

DEFINITION OF USER CHARACTER

Code: ***27 66 nchr Pat 0 ... Pat 7***
Hex code: ***1B 42 nchr Pat 0 ... Pat 7***
Mnemonic: ***ESC B ASCII(nchr) ASCII(Pat 0) ... ASCII(Pat 7)***

After the two command identification codes, other 9 bytes must be sent to **QTP 4x6** with the following meaning:

nchr	*(0÷7)*	*(00÷7 Hex)*	->	*Number of user character to define*
Pat 0	*(0÷31)*	*(00÷1F Hex)*	->	*First byte of pattern equal to first high row of character.*
:	*:*		->	*:* *:* *:* *:*
Pat 7	*(0÷31)*	*(00÷1F Hex)*	->	*Seventh byte of pattern equal to last low row of character.*

This command loads on the display the pattern of the user character nchr with the value placed in the eight byte byte Pat 0 ÷ Pat 7, as described in figure 23; the pattern is only defined but not saved, so if **QTP 4x6** is turned off and on the user character nchr doesn't mantain the supplied pattern.
For example if you wish to define the user character 5 as an empty rectangle with maximum dimension, the following sequence has to be sent:

27 66 5 31 17 17 17 17 17 17 31 or
1B 42 05 1F 11 11 11 11 11 11 1F Hex

DEFINITION AND MEMORIZATION OF USER CHARACTER

Code: ***27 33 66 nchr Pat 0 ... Pat 7***
Hex code: ***1B 21 42 nchr Pat 0 ... Pat 7***
Mnemonic: ***ESC ! B ASCII(nchr) ASCII(Pat 0) ... ASCII(Pat 7)***

After the three command identification codes, other 9 bytes must be sent to **QTP 4x6** with the following meaning:

nchr	*(0÷7)*	*(00÷7 Hex)*	->	*Number of user character to define and save*
Pat 0	*(0÷31)*	*(00÷1F Hex)*	->	*First byte of pattern equal to first high row of character.*
:	*:*		->	*:* *:* *:* *:*
Pat 7	*(0÷31)*	*(00÷1F Hex)*	->	*Seventh byte of pattern equal to last low row of character.*

This command loads on the display the pattern of the user character nchr with the value placed in the eight byte byte Pat 0 ÷ Pat 7, as described in figure 23; moreover the pattern is also saved on EEPROM, so if **QTP 4x6** is turned off and on the user character nchr mantain the supplied pattern.

NOTE: This command writes data on the on board EEPROM, so before executing it is better to check the EEPROM writing possibility through the proper command; in fact if it is not ready the command is ignored.
Execution time is about 80 msec: if several commands must follow this, it is better to insert a delay to avoid input buffer overflow.

COMMANDS FOR MESSAGE MANAGEMENT

In the following paragraphs are described all the commands that manage messages on **QTP 4x6**. The messages are 20 characters sequence that can be saved on board EEPROM and then reloaded or represented on display, simply by suppling the same message identification number. The most important function of messages is the possibility to show constant information on the display (i.e. allarms, equipment status, etc.) without the transmission of the numerous characters of this information but only the few characters of the commands. Furthermore a comfortable program for PC, named **QTP EDIT**, allows any user to edit the messages, save and load them on PC disks and transmit/receive them directly to/from **QTP** serially connected to PC.
QTP 4x6 can accept one EEPROM with two different size: 512 bytes in base version and 2048 bytes in **.MEX** version. This last is an option that must be specified in the order.

READING OF THE LAST STORAGED MESSAGE NUMBER

Code: ***27 110***
Hex code: ***1B 6E***
Mnemonic: ***ESC n***

This comand returns on the serial line the number of the last messages that can be saved on EEPROM. It varys in compliace with the size of the EEPROM installed on the card, as reported in the below table:

Version	*EEPROM Size*	*N°.last message*
-	512 Bytes	19 (13 Hex)
-	1024 Bytes	44 (2C Hex)
.MEX	2048 Bytes	96 (60 Hex)

FIGURE 24: NUMBER OF MESSAGES STORAGEABLE ON EEPROM

MESSAGE STORAGE

Code: **27 33 67 n.mes. chr. 0... chr.19**
Hex code: **1B 21 43 n.mes. chr. 0... chr.13 Hex**
Mnemonic: **ESC ! C ASCII(n.mes.) ASCII(chr.0)...ASCII(chr.19)**

This command stores the 20 characters message, with number indicated as **n.mes.**, on the on board EEPROM. The 20 chars which form the message must be visualizable on the display so they must be included in the range **16÷255 (10÷FF Hex)**. The message number must be included in the range of **0÷max. n**., where max.n. is the number of the last storaged message just described in figure 13.

NOTE: This command writes data on the on board EEPROM, so before executing it is better to check the EEPROM writing possibility through the proper command; in fact if it is not ready the command is ignored.

MESSAGE READING

Code:	***27 33 69 n.mes.***
Hex code:	***1B 21 45 n.mes.***
Mnemonic:	***ESC ! E ASCII(n.mes.)***

This command reads the 20 characters message corresponding to **n.mes.** by the EEPROM and it sends this message on serial line, beginning from the first char of the string. At the end of the message, the **CR+LF** codes are sent, too.
The message number must be included in the range of **0÷max. no.**, where max.no. is the number of the last storaged message just previously described in figure 24. If this number is not compatible with the **QTP 4x6** installed EEPROM size, this command is ignored.

MESSAGE VISUALIZATION

Code:	***27 33 68 n.mes. n***
Hex code:	***1B 21 44 n.mes. n***
Mnemonic :	***ESC ! D ASCII(n.mes.) ASCII(n)***

This command visualizes **n** 20 characters messages on the display, beginning from current cursor position.
The first of the **n** messages is that one having the number corresponding to **n.mes.** while the remaining messages are those ones immediately subsequents in EEPROM.
The message number must be included in the range **0÷max.no**, where max.no. is the value described in figure 24. If this number is not compatible with the **QTP 4x6** installed EEPROM size, this command is ignored.
The **n** quantity of messages to be visualized depends only on the model of the display and it is included in these ranges:

20x2 display	**n** between 1÷2
20x4 or 40x2 display	**n** between 1÷4

If the **n** value is not compatible with the installed display model, the command is ignored.
The cursor is placed in the next position of the last character visualized; if the last character of the said message occupies the last position of the display, the cursor will be placed in home position.
For example; if you wish to visualize the messages number 10 and 11, it will be necessary to send the following sequence:

27 33 68 10 2 or **1B 21 44 0A 02 Hex** or **ESC ! D LF STX**

SCROLLING MESSAGE VISUALIZATION

Code:	***27 33 83 mess.no. chars***
Hex code:	***1B 21 45 mess.no. chars Hex***
Mnemonic:	***ESC ! E ASCII(mess.no.) ASCII(chars)***

This command visualizes, on the first row of the display, a scrolling message **chars** charactes long; in fact the characters that form the message are shifted from the right to the left, making possible to show on an unique row of the display (the first row), an amount of informations greater than the one normally available.
The message, which is **chars** characters long, begins from the first character of the message whose number is **mess.no.** and is composed by the characters that make the **mess.no.** message and the following ones (making the following messages stored on the EEPROM).
The message number must be included in the range of **0÷max. no.**, where max.no. is the number of the last storable message previously described in figure 24. If this number is not compatible with the **QTP 4x6** installed EEPROM size, this command is ignored.
The value **chars** may have these meanings:

0	Stops the current scrolling (value of **mess.no.** is irrilevant)
20÷200	If a **20** characters per row display is installed
40÷200	If a **40** characters per row display is installed

If **chars** has a value out of these ranges or it extends the sliding messages beyond the limit of the EEPROM storage space, the command is ignored.

The message will slide in the first row of the display, without changing position and attributes of the cursor.

If, for example, the User wants to show a scrolling message 23 characters long, made by message 5 (20 characters) and the first 3 characters of message 6,it will be necessary to send the following sequence:

27 33 83 10 35 or **1B 21 53 0A 23 Hex** or **ESC ! S LF #**

NOTE
Scrolling a message involves a continuous display updating; this operation slows the interpretation of commands coming from the serial port.
So if a great amount of informations must be sent to **QTP 4x6** and a message is scrolling on the display, it is suggestable to wait for some **msec** between the transmission of a 20÷30 bytes data block and the next one, to assure that the terminal has had the time to interpretate correctly the transmitted data.

EXTERNAL CARDS

The typical application of **QTP 4x6** are those that require interaction between operator and controlled machine for data, status, measures, information exchanges. The card can be connected to a wide range of programmable devices provided of one serial communication line produced by **grifo®**, or to many system of other companies.
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® 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; **ABACO®** I/O BUS.

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; **ABACO®** I/O BUS.

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; **ABACO®** I/O BUS.

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® 154

"4" Type General Purpose Controller Z80

84C15 µP, 10 MHz; full CMOS; 1 RS 232 line; 1 RS 232 or RS 422-485 line; 16 TTL I/O lines; 512K EPROM or FLASH; 512K RAM and RTC backed; 8K serial EEPROM; 2÷4 timers/counters;Watch dog; 2 readable DIPs; LCD Interface; **ABACO®** I/O BUS; 5Vdc power supply. Size100x50 mm.

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.

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; type 4 format; 32K EPROM; 32K SRAM backed with Lithium battery; 32K EPROM, SRAM, EEPROM; RTC; 1 serial line RS 232, RS 422 or RS 485; 10 TTL I/O lines; 3 timers/counters; watch dog; 8 A/D converter signals with 8 bits resolution; 1 synchronous serial line; extremly low power consumption; interface for **ABACO®** I/O BUS.

GPC® AM4

General Purpose Controller ATmega103

Microprocessor ATmega103 at 5.5 MHz; CMOS implementation; 128K internal FLASH; 32K SRAM; Back-Up with Lithium battery internal or external; 4K internal EEPROM; 1 serial line RS 232, RS 422, RS 485 or current loop; 16 I/O TTL; 8 linee A/D resolution 10 bits; 3 timers/counters; Watch Dog; Real Time Clock; **ABACO®** I/O BUS expansion. Interface for ISP programming.

MSI 01

Multi Serial Interface 1 line

Interface card for TTL serial line that is buffered in RS 232, RS 422, RS 485, or current loop line. The TTL line is on a mini screw connector and the buffered one is on standard plug connector.

IBC 01

Interface Block Comunication

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

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 SRAM 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.

GPC® 15A

General Purpose Controller 84C15

Full CMOS card, 10÷20 MHz 84C15 CPU; 512K EPROM or FLASH EPROM; 128K RAM; 2K or 8K backed RAM+RTC; 8K serial EEPROM; 1 RS 232 serial line; 1 RS 232, RS 422, RS 485 or current loop line; 40 TTL I/O lines; 2 counters timers; 2 watch dogs; 2 dip switches, buzzer.

GPC® R/T94

General Purpose Relays/transistors 9 inputs 4 outputs

CMOS card, 14 MHz 89C4051 CPU; 4K FLASH; 128 byte RAM; 256 byte SRAM+RTC backed through battery; 1K serial EEPROM; 1 RS 232, RS 422, RS 485 or current loop line; 9 optocoupled NPN inputs; 4 relays outputs (5 A) or transistor (4A 45 Vdc) optocoupled; I/O lines displayed by LEDs; 1 counter timer.+5 Vdc power supply or 8÷24 Vac wide range; plastic container for Ω rails.

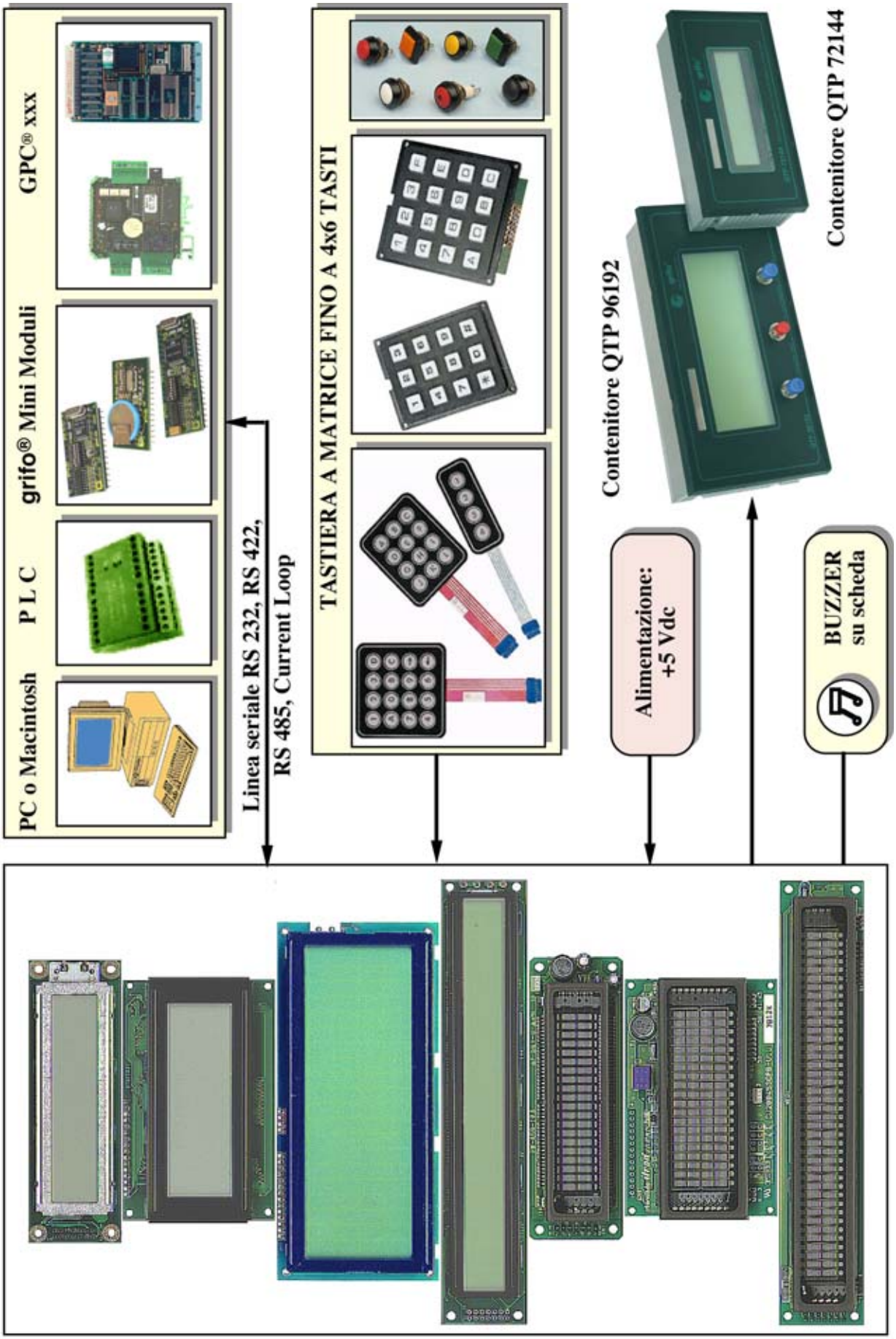


FIGURE 25: AVAILABLE CONNECTIONS DIAGRAM

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® 550

General Purpose Controller 80C552

Microprocessor 80C552 at 22 MHz. 32K EPROM; 32 K RAM; 32 K EEPROM or SRAM; RTC; serial EEPROM; serial lines 1 RS 232 + 1 RS 232 or RS 422-485 or current loop; 40 I/O TTL; 2 lines of PWM; 16 bits timer/counter; watch dog; dip switch; 8 lines 10 bit A/D converter; interface for BUS **ABACO®**; CAN line galvanically isolated. Unique power supply +5 Vdc; EUROCARD format.

SBP 02-xx

Switch BLOCK Power xx version

Low cost switching power supply able to generate voltage from +5 to +40 Vdc and current up to 2.5 A; Input from 12 to 24 Vac; Connection for DIN C Type and Ω rails.

APPENDIX A: COMMAND CODES SUMMARY TABLES

The tables of this appendix list a summary of all the command sequences recognized by **QTP 4x6**.

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*
Home	01	01	SOH
Cursor left	21	15	NACK
Cursor right	06	06	ACK
Cursor down	10	0A	LF
Cursor up	26	1A	SUB
Carriage return	13	0D	CR
Carraige return+line feed	29	1D	GS
Absolute cursor position	27 89 r c	1B 59 r c	ESC Y ASCII(r) ASCII(c)
Back space	08	08	BS
Clear page	12	0C	FF
Clear line	25	19	EM
Clear end of line	27 75	1B 4B	ESC K
Clear end of page	27 107	1B 6B	ESC k
Cursor off	27 80	1B 50	ESC P
Steady cursor on	27 79	1B 4F	ESC O
Blinkling block cursor on	27 81	1B 51	ESC Q
Beep	07	07	BEL
Reading of version number	27 86	1B 56	ESC V
Operating mode selection	27 65 mode	1B 41 mode	ESC A ASCII(mode)

FIGURE A1: COMMAND CODES SUMMARY TABLE (1 OF 2)

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*
Request of EEPROM writing	27 51	1B 33	ESC 3
Writing of presence byte	27 33 78 byte	1B 21 4E byte	ESC ! N ASCII(byte)
Reading of presence byte	27 33 110	1B 21 6E	ESC ! n
Key code reconfiguration	27 55 key n. code	1B 37 key n. code	ESC 7 ASCII(key n.) ASCII(code)
Keyclick on without memorization	27 53	1B 35	ESC 5
Keyclick off without memorization	27 54	1B 36	ESC 6
Keyclick on with memorization	27 33 53	1B 21 35	ESC ! 5
Keyclick off with memorization	27 33 54	1B 21 36	ESC ! 6
Definition of user character	27 66 nchar Pat0...Pat7	1B 42 nchar Pat0...Pat7	ESC B ASCII(nchar) ASCII(Pat0)...ASCII(Pat7)
Definition and memorization of user character	27 33 66 nchar Pat0...Pat7	1B 21 42 nchar Pat0...Pat7	ESC ! B ASCII(nchar) ASCII(Pat0)...ASCII(Pat7)
Reading of max message number	27 110	1B 6E	ESC n
Message storage	27 33 67 n.mess. chr.0…chr.19	1B 21 43 n.mess. chr.0…chr.13	ESC ! C ASCII(n.mess.) ASCII(chr.0)…ASCII(chr.19)
Message reading	27 33 69 n.mess.	1B 21 45 n.mess.	ESC ! E ASCII(n.mess.)
Visualization of n messaggi	27 33 68 n.mess. n	1B 21 44 n.mess. n	ESC ! D ASCII(n.mess.) ASCII(n)
Scrolling message visualization	27 33 83 n.mess. n.char	1B 21 53 n.mess. n.char	ESC ! S ASCII(n.mess.) ASCII(n.char)

FIGURE A2: COMMAND CODES SUMMARY TABLE (2 OF 2)

APPENDIX B: DISPLAY CHARACTERS

The following tables shows the characters sets displayed on **QTP 4x6** for all the possible received characters, according with ordered display and model.
Even the not ASCII characters (or special characters) change when the display type changes and if the user requires a character set different from those described in the following figures, he can take a direct contact with **grifo®**.

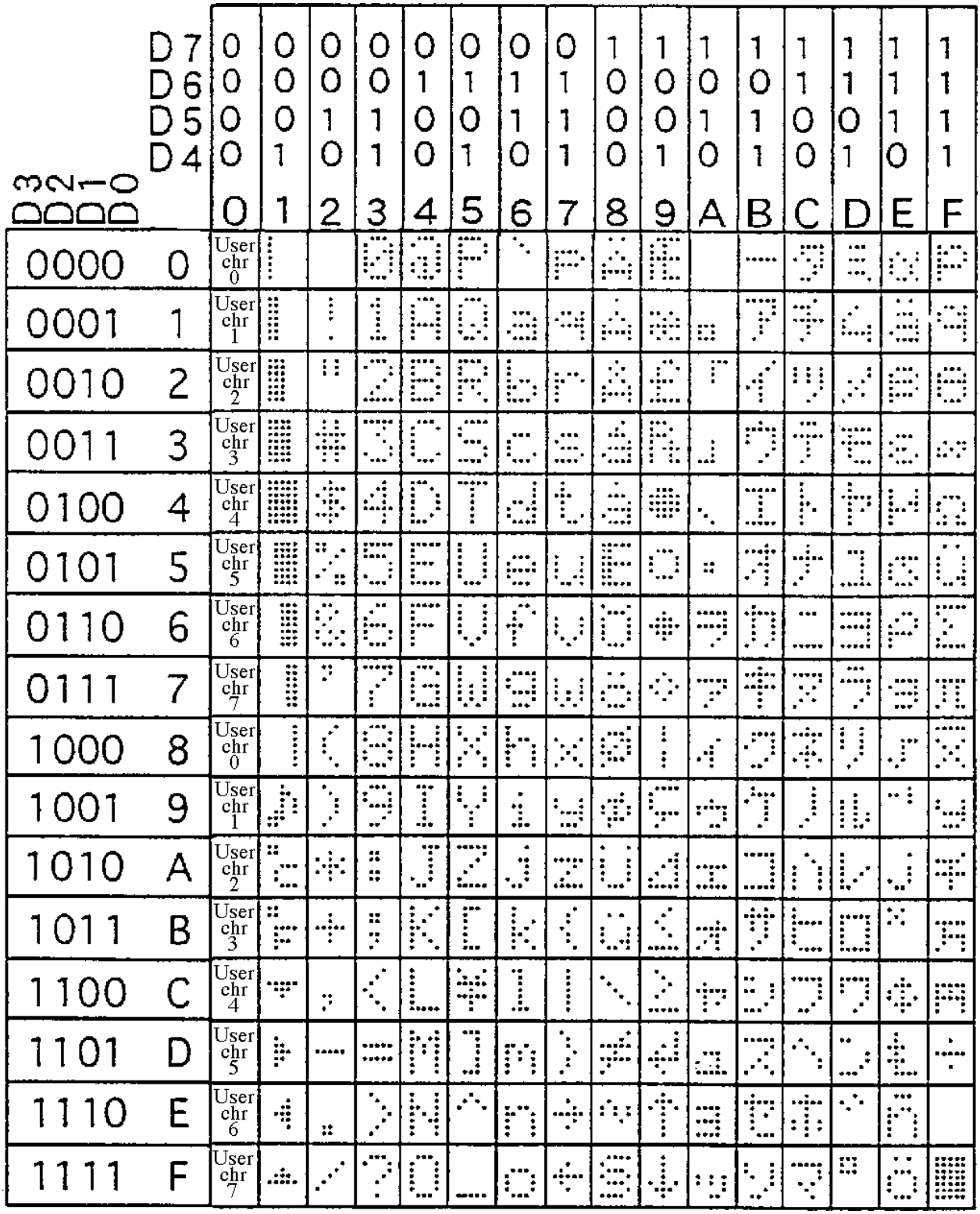


FIGURE B1: QTP 4X6 -F2, F4, F24 CHARACTERS TABLE

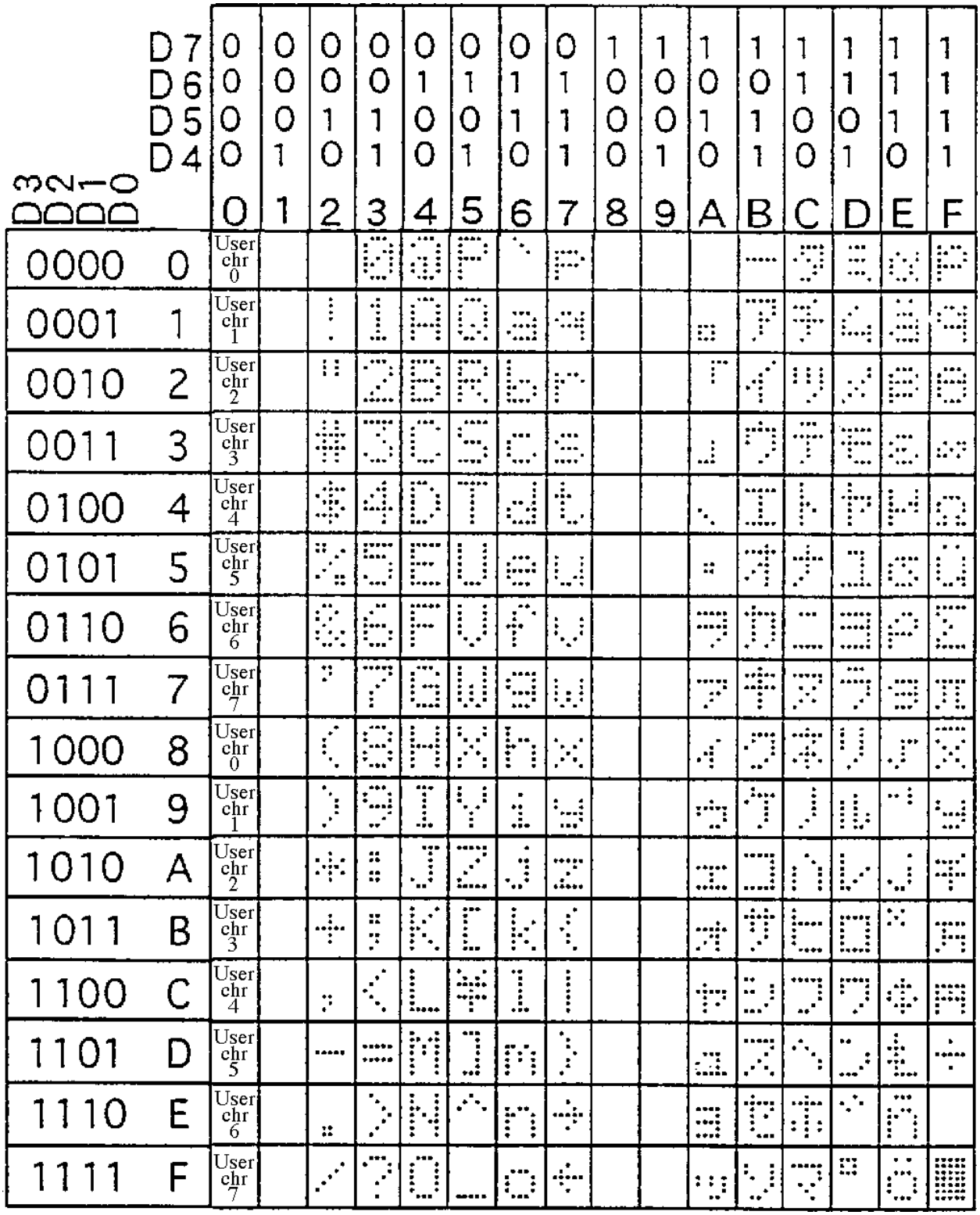


FIGURE B2: QTP 4X6-C4B, C24 CHARACTERS TABLE

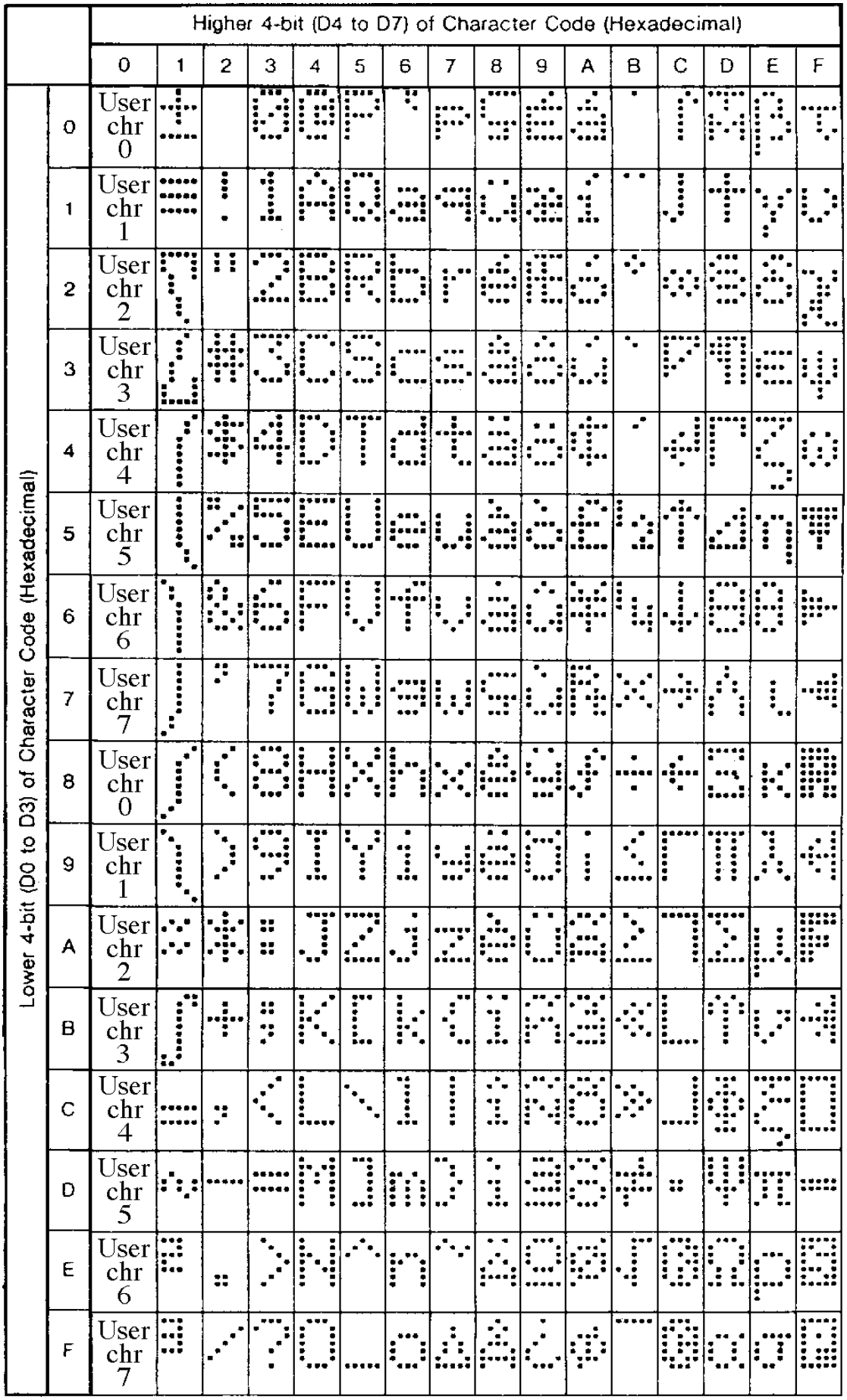


FIGURE B3: QTP 4X6-C2, C4 CHARACTERS TABLE

APPENDIX C: MOUNTING OUTLINE DIMENSIONS

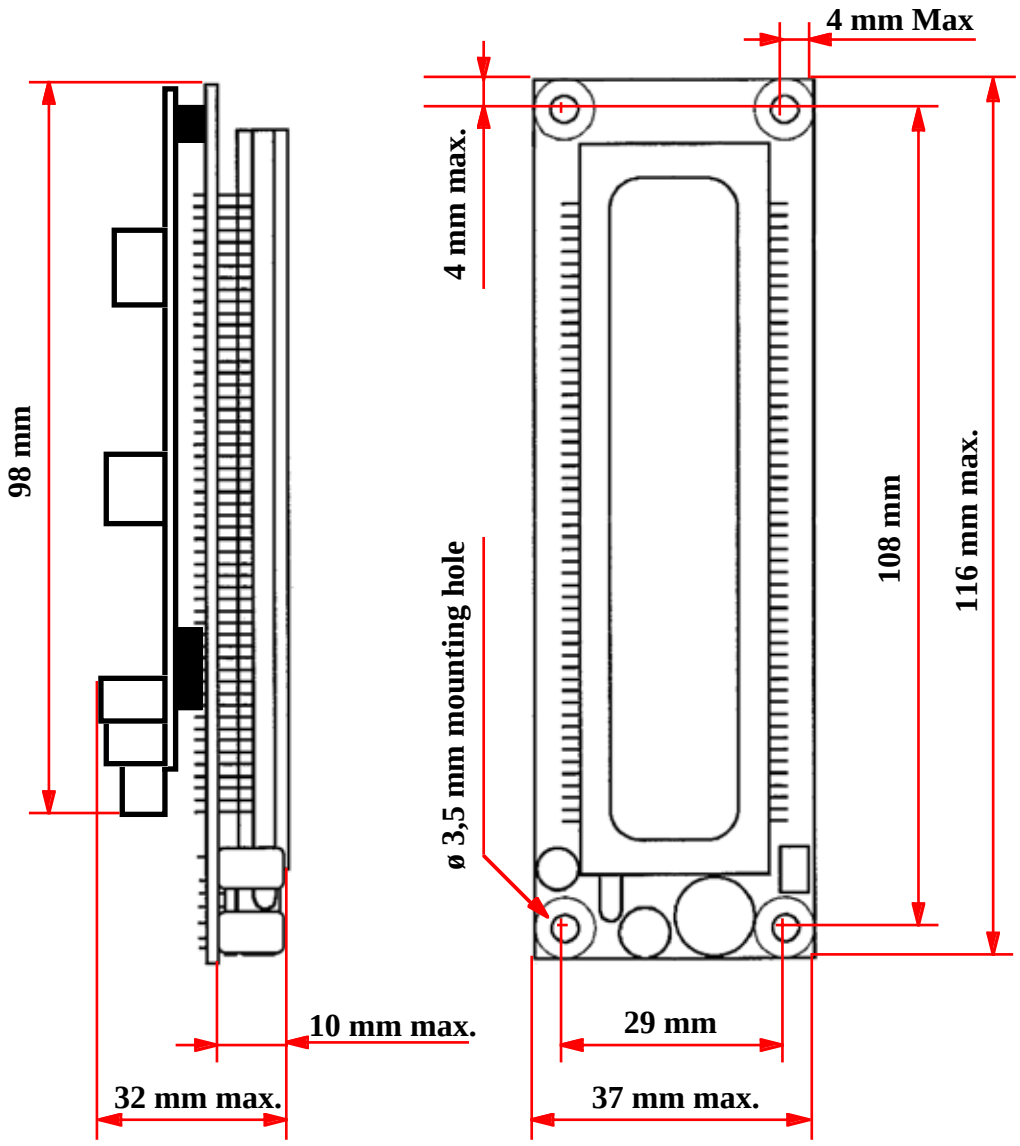


FIGURE C1: QTP 4X6-C2, F2 DIMENSIONS

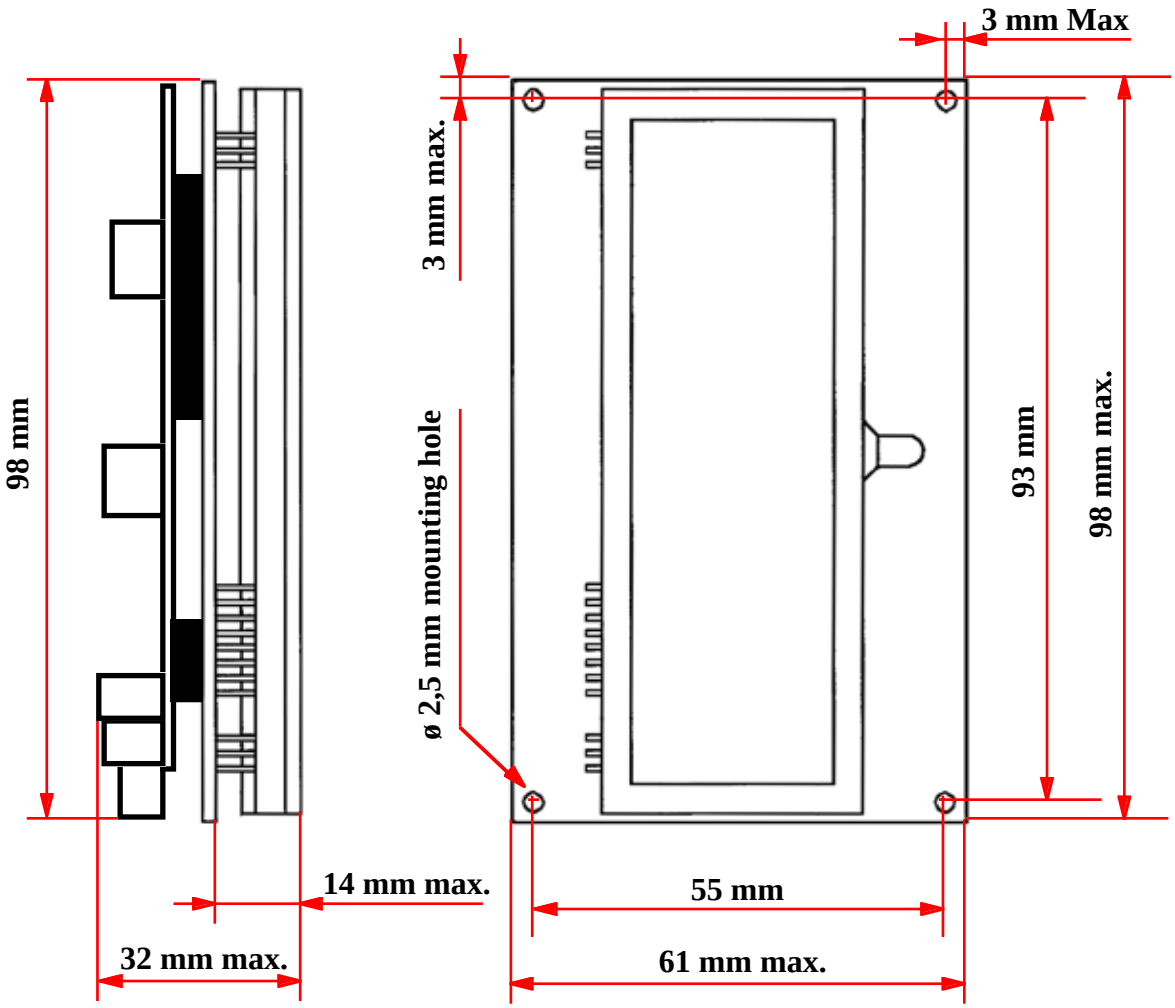


FIGURE C2: QTP 4X6-C4, F4 DIMENSIONS

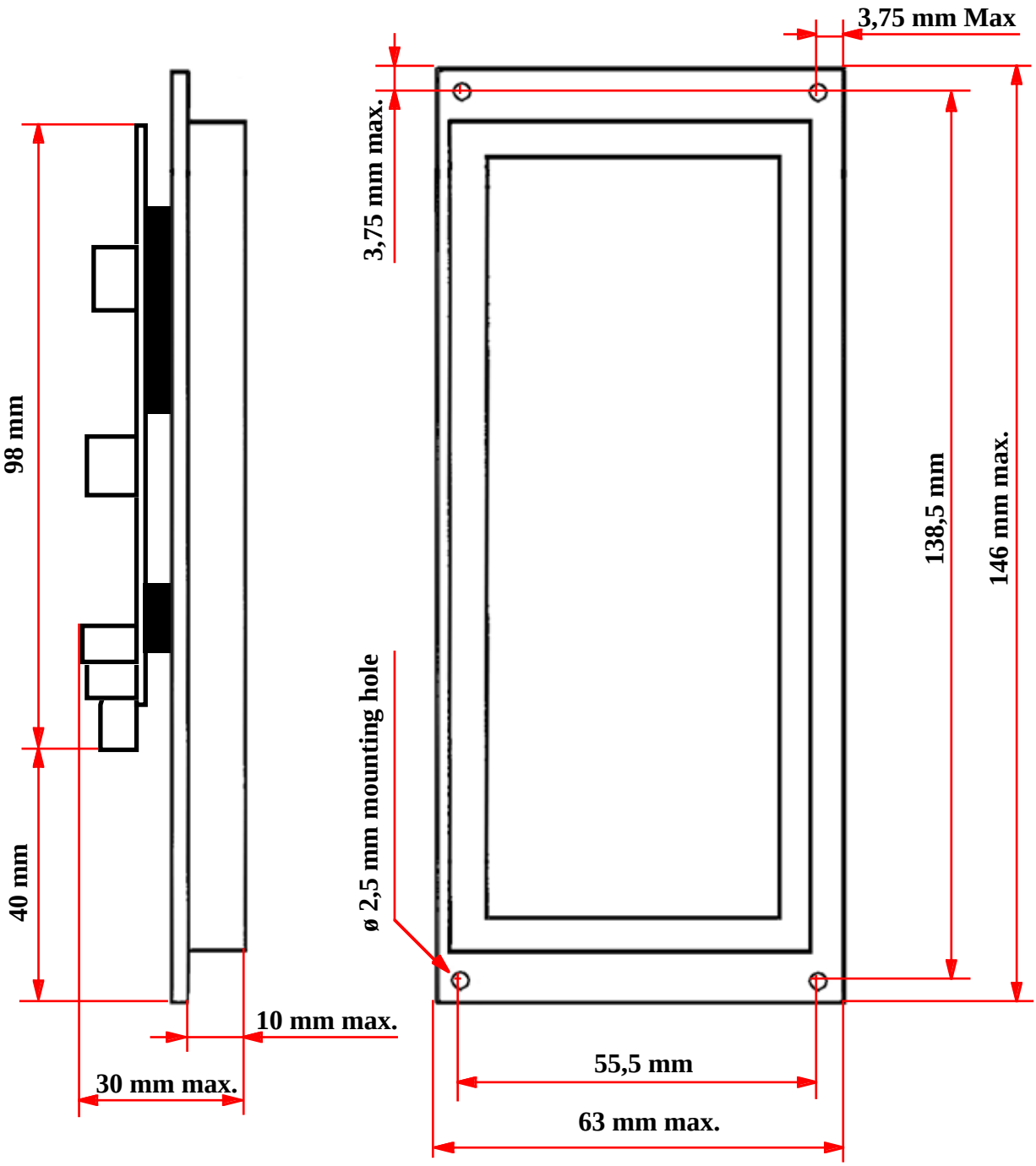


FIGURE C3: QTP 4X6-C4B DIMENSIONS

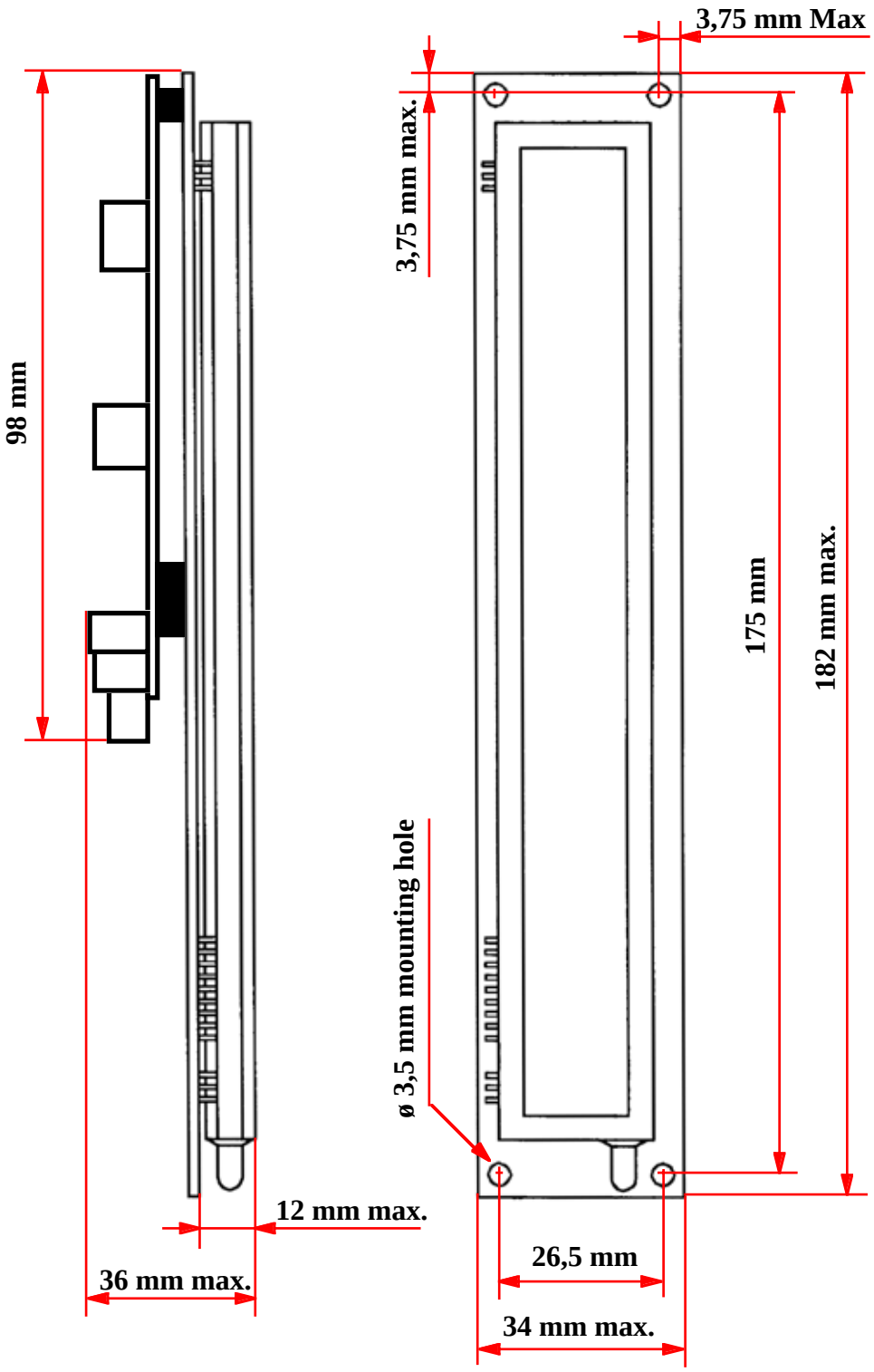


FIGURE C4: QTP 4X6-C24, F24 DIMENSIONS

APPENDIX D: ALPHABETICAL INDEX