

# QTP G28

Quick Terminal Panel Graphic 28 keys

## USER MANUAL



**grifo**<sup>®</sup>

ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6  
40016 San Giorgio di Piano  
(Bologna) ITALY

E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)

<http://www.grifo.it>

<http://www.grifo.com>

Tel. +39 051 892.052 (a. r.)

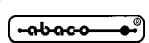
FAX: +39 051 893.661



QTP G28

Edition 5.00

Rel. 05 September 2000

, GPC<sup>®</sup>, grifo<sup>®</sup>, are trade marks of grifo<sup>®</sup>



# QTP G28

Quick Terminal Panel Graphic 28 keys

## USER MANUAL

Intelligent operator interface with frontal size 281x139 mm; frontal depth 32 mm; rear depth 26 mm; surface or flush panel mounting Direct mounting to **Phoenix CombiCard®** industrial instrument cases. Professional keyboard: **28 Keys** and **16 LEDs**; **complete reconfiguration** of all keycodes at User level. **Label slots** for personalization of 5 keys, 5 LEDs and device naming; direct management of **16 LEDs** plus **Blinking** attribute; 11 keys provided with signalation LEDs. **Graphic LCD Display**, back lighted through a cold cathod fluorescent lamp, **240x128** pixels, keyboard regulated contrast and brightness; maximum alphanumeric display: **30** characters per **16** rows. **Graphic primitives** like circumference, rectangle, line, cartesian axis, arrow, characters **zoom** and **direction**, etc. **Buzzer** for **BELL** and Key pressed signalations. Serial **E<sup>2</sup>**, up to 8 Kbytes, to store set up, messages, key codes etc; **RTC** capable to manage leap years, provided with **Lithium** back up battery. Main serial communication line, **galvanically isolated**, in **RS 232**, **RS 422**, **RS 485**, **Current Loop** or **CAN**; auxiliary serial communication line, **galvanically isolated**, in **RS 232**. Local Setup for selecting communication protocols, keyclick abilitation, etc. ;software manageable 3 A **Relay** with normally open and normally closed contacts; Interface to manage a **Badge reader** able to read trace 1, 2 or 3, manual o motorized or Smart Card reader. On board power supply provided with noise supressor filter, also for small external loads; **DC** or **AC** supply voltage, up to **24Vac**

**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>

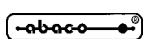
Tel. +39 051 892.052 (a. r.) FAX: +39 051 893.661



QTP G28

Edition 5.00

Rel. 05 September 2000

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

## DOCUMENTATION COPYRIGHT BY grifo®, ALL RIGHTS RESERVED

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

### IMPORTANT

Although all the information contained herein have been carefully verified, **grifo®** assumes no responsibility for errors that might appear in this document, or for damage to things or persons resulting from technical errors, omission and improper use of this manual and of the related software and hardware.

**grifo®** 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:

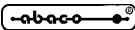


Attention: Generic danger



Attention: High voltage

### Trade Marks

, **GPC®**, **grifo®** : are trade marks of **grifo®**.

Other Product and Company names listed, are trade marks of their respective companies.

# GENERAL INDEX

<b>INTRODUCTION</b> .....	1
<b>FIRMWARE RELEASE</b> .....	1
<b>GENERAL INFORMATION</b> .....	2
<b>DISPLAY</b> .....	3
<b>KEYBOARD</b> .....	3
<b>INDICATOR LEDS</b> .....	3
<b>BUZZER</b> .....	3
<b>SERIAL COMMUNICATION</b> .....	4
<b>EEPROM</b> .....	4
<b>USER RELAY</b> .....	4
<b>REAL TIME CLOCK</b> .....	4
<b>BADGE READER INTERFACE</b> .....	5
<b>CAN INTERFACE</b> .....	5
<b>ON BOARD POWER SUPPLY</b> .....	5
<b>TECHNICAL SPECIFICATIONS</b> .....	6
<b>GENERAL FEATURES</b> .....	6
<b>PHYSICAL FEATURES</b> .....	6
<b>OVERALL DIMENSION</b> .....	7
<b>ELECTRIC FEATURES</b> .....	8
<b>INSTALLATION</b> .....	10
<b>CN1 - POWER SUPPLY CONNECTOR</b> .....	10
<b>CN2 - EXTERNAL LOADS POWER SUPPLY CONNECTOR</b> .....	11
<b>CN3A - PRIMARY SERIAL LINE CONNECTOR</b> .....	12
<b>CN3B - AUXILIARY SERIAL LINE CONNECTOR</b> .....	18
<b>CN4 - USER RELAY CONNECTOR</b> .....	20
<b>CN6 - CAN INTERFACE CONNECTOR</b> .....	22
<b>JUMPERS</b> .....	24
<b>SOFTWARE DESCRIPTION</b> .....	26
<b>LOCAL SET UP</b> .....	26
<b>CAN INTERFACE</b> .....	27
<b>VISUALIZATION MODE</b> .....	27
<b>COMMUNICATION MODE</b> .....	28
<b>MASTER SLAVE COMMUNICATION MODE</b> .....	28
<b>KEYBOARD ACQUISITION</b> .....	29
<b>KEYS MAP</b> .....	30
<b>DEFAULT KEYS CODES</b> .....	31
<b>KEYBOARD REGULATION OF DISPLAY BRIGHTNESS</b> .....	32
<b>CHARACTERS VISUALIZATION ON THE DISPLAY</b> .....	32
<b>RECEIVE BUFFER</b> .....	32

COMMANDS FOR CHARACTERS ERASURE .....	33
BACKSPACE .....	33
CLEAR PAGE .....	33
CLEAR LINE .....	33
CLEAR END OF LINE.....	33
CLEAR END OF PAGE .....	33
COMMANDS FOR CURSOR POSITIONING .....	34
CURSOR LEFT .....	34
CURSOR RIGHT .....	34
CURSOR DOWN .....	34
CURSOR UP .....	34
HOME .....	34
CARRIAGE RETURN .....	35
CARRIAGE RETURN+LINE FEED .....	35
ALPHANUMERIC CURSOR PLACEMENT .....	35
RETURN ALPHANUMERIC CURSOR POSITION .....	35
COMMANDS FOR EEPROM .....	36
REQUEST FOR EEPROM WRITING POSSIBILITY .....	36
WRITING OF LIFE BYTE .....	36
READING OF LIFE BYTE .....	36
WRITE BLOCK OF DATA ON EEPROM.....	37
READ BLOCK OF DATA FROM EEPROM .....	37
COMMANDS FOR LEDS MANAGEMENT .....	38
LED ACTIVATION .....	38
LEDS ACTIVATION WITH MASK .....	38
LEDS MAP .....	39
COMMANDS FOR KEYBOARD MANAGEMENT .....	40
KEY RECONFIGURATION .....	40
KEYCLICK ON WITH MEMORIZATION .....	40
KEYCLICK OFF WITH MEMORIZATION .....	40
KEYCLICK ON WITHOUT MEMORIZATION .....	41
KEYCLICK OFF WITHOUT MEMORIZATION .....	41
COMMANDS FOR VARIOUS FUNCTIONS .....	42
READING OF VERSION NUMBER .....	42
RELAY ACTIVATION .....	42
RELAY DEACTIVATION .....	42
BEEP .....	42
SETTING OF ALPHANUMERIC VISUALIZATION MODE .....	42
SETTING OF GRAPHIC VISUALIZATION MODE.....	43
SCREEN VISUALIZATION .....	43
STRING VISUALIZATION .....	43
COMMANDS FOR AUXILIARY SERIAL LINE MANAGEMENT .....	44
ENABLE COPY OF CHR ON AUXILIARY SERIAL LINE .....	44

ENABLE TRANSPARENT COPY OF CHR ON AUXILIARY SERIAL LINE ..... 44  
 DISABLE COPY OF CHR ON AUXILIARY SERIAL LINE ..... 44  
 READ STATUS OF AUXILIARY SERIAL LINE HANDSHAKE ..... 44

COMMANDS FOR CURSOR ATTRIBUTES MANAGEMENT ..... 45  
 CURSOR OFF ..... 45  
 STATIC CURSOR ON ..... 45  
 BLINKING CURSOR ON ..... 45

COMMANDS FOR REAL TIME CLOCK MANAGEMENT ..... 46  
 CLOCK SET UP ..... 46  
 CLOCK READING ..... 46  
 TIME VISUALIZATION ..... 46  
 DATE VISUALIZATION ..... 47

GRAPHIC COMMANDS ..... 48  
 REVERSE ATTRIBUTE SETTING ..... 48  
 ENABLE SELECTED ATTRIBUTE ..... 48  
 DISABLE SELECTED ATTRIBUTE ..... 48  
 CHARACTERS WRITE DIRECTION ..... 48  
 CHARACTERS ZOOM SETTING ..... 49  
 CHARACTERS ELONGATION SETTING ..... 49  
 GRAPHIC CURSOR PLACEMENT ..... 49  
 RETURN GRAPHIC CURSOR POSITION ..... 50  
 ONE PIXEL VISUALIZATION ..... 50  
 LINE DRAWING ..... 50  
 RECTANGLE DRAWING ..... 50  
 FILLED RECTANGLE DRAWING ..... 51  
 GRADUATED AXES AND GRID DRAWING ..... 51  
 RIGHT ORIENTED FILLED ARROW DRAWING ..... 52  
 RIGHT ORIENTED ARROW DRAWING ..... 52  
 BOTTOM ORIENTED FILLED ARROW DRAWING ..... 53  
 BOTTOM ORIENTED ARROW DRAWING ..... 53  
 LEFT ORIENTED FILLED ARROW DRAWING ..... 54  
 LEFT ORIENTED ARROW DRAWING ..... 54  
 TOP ORIENTED FILLED ARROW DRAWING ..... 55  
 TOP ORIENTED ARROW DRAWING ..... 55  
 CIRCUMFERENCE AND ARCS DRAWING ..... 56  
 FILLED CIRUMFERENCE WITH 3 PIXELS RADIUS DRAWING ..... 56  
 CIRUMFERENCE WITH 3 PIXELS RADIUS DRAWING ..... 57

CAN INTERFACE COMMANDS ..... 58  
 CAN UART WRITING ..... 58  
 CAN UART READING ..... 58  
 CAN UART MESSAGE WRITING ..... 58  
 CAN UART MESSAGE READING ..... 58

APPENDIX A: COMMANDS CODES SUMMARY TABLES ..... A-1



grifo®

ITALIAN TECHNOLOGY

**APPENDIX B: DISPLAY CHARACTERS TABLE ..... B-1**

**APPENDIX C: LABELS INSERTION ..... C-1**

**APPENDIX D: ALPHABETICAL INDEX ..... D-1**





# FIGURE INDEX

FIGURE 1: TERMINAL DIMENSION .....	7
FIGURE 2: FRONTAL VIEW .....	9
FIGURE 3: CN1 - POWER SUPPLY CONNECTOR .....	10
FIGURE 4: CN2 - EXTERNAL LOADS POWER SUPPLY CONNECTOR .....	11
FIGURE 5: CN3A - PRIMARY SERIAL LINE CONNECTOR .....	12
FIGURE 6: RS 232 PRIMARY SERIAL CONNECTION EXAMPLE .....	13
FIGURE 7: RS 422 POINT TO POINT CONNECTION EXAMPLE .....	14
FIGURE 8: RS 485 POINT TO POINT CONNECTION EXAMPLE .....	14
FIGURE 9: RS 485 NETWORK CONNECTION EXAMPLE .....	15
FIGURE 10: 4 WIRES ACTIVE CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE .....	16
FIGURE 11: 4 WIRES PASSIVE CURRENT LOOP NETWORK CONNECTION EXAMPLE .....	16
FIGURE 12: 4 WIRES PASSIVE CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE .....	17
FIGURE 13: 2 WIRES PASSIVE CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE .....	17
FIGURE 14: CN3B - AUXILIARY SERIAL LINE CONNECTOR .....	18
FIGURE 15: RS 232 AUXILIARY SERIAL CONNECTION EXAMPLE .....	19
FIGURE 16: USER RELAY CONNECTOR .....	20
FIGURE 17: PHOTO .....	21
FIGURE 18: CN6 - CAN INTERFACE CONNECTOR .....	22
FIGURE 19: CAN INTERFACE CONNECTION EXAMPLE .....	23
FIGURE 20: JUMPERS TABLE .....	24
FIGURE 21: JUMPERS LOCATION .....	25
FIGURE 22: LOCAL SET UP SCREEN .....	26
FIGURE 23: KEYS MAP .....	30
FIGURE 24: DEFAULT KEYS CODE .....	31
FIGURE 25: LEDs MAP .....	39
FIGURE 26: RIGHT ORIENTED FILLED ARROW .....	52
FIGURE 27: RIGHT ORIENTED ARROW .....	52
FIGURE 28: BOTTOM ORIENTED FILLED ARROW .....	53
FIGURE 29: BOTTOM ORIENTED ARROW .....	53
FIGURE 30: LEFT ORIENTED FILLED ARROW .....	54
FIGURE 31: LEFT ORIENTED ARROW .....	54
FIGURE 32: TOP ORIENTED FILLED ARROW .....	55
FIGURE 33: TOP ORIENTED ARROW .....	55
FIGURE 34: ARCS NUMERATION AND LOCATION .....	56
FIGURE A1: COMMANDS CODES TABLE (1 OF 4) .....	A-1
FIGURE A2: COMMANDS CODES TABLE (2 OF 4) .....	A-2
FIGURE A3: COMMANDS CODES TABLE (3 OF 4) .....	A-3
FIGURE A4: COMMANDS CODES TABLE (4 OF 4) .....	A-4
FIGURE B1: ALPHANUMERIC MODE CHARACTERS TABLE .....	B-1
FIGURE C1: LABEL INSERTION ON THE TERMINAL .....	C-1



## 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 environment, 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 respect 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 RELEASE

This handbook make reference to firmware release **2.0** 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 between the notations. Inside the device, the firmware release number is written on a label stucked on a component or it can be obtained by a proper command sent through the serial line.

## GENERAL INFORMATION

**QTP G28** is a complete **IP-65** operator panel, specifically designed for industrial use and for direct mounting on automatic machinery. It is, in every respect, video terminals suitable to be the direct interface between operator and machinery in any of the control or command operations which could be necessary during running or diagnostic of the same. **QTP G28** features a **Graphic LCD Display**, back lighted by a **cold cathod fluorescent lamp**, in the format **240x128** pxels corresponding to, in alphanumeric mode, to as much as 30 characters per 16 rows. **QTP G28** also features a 28 keys keyboard, key codes are completely reprogrammable by the User, a label slot can be used to insert a name for 5 of these keys; 11 keys are provided with signalation LEDs that can be managed by specific commands sent to the terminal through the serial line.

A second label slot allows to attribute a writing to 5 other LEDs, a third slot allows to insert a name or the User's own logo directly on the terminal front side. **QTP G28** can be installed inside **Phoenix** industrial housings **CombiCard®** serie, obtaining an extremely compact block, featuring protections that can be even **IP 65**. The basic version of **QTP G28** can be expanded utilizing the various options available, as: serial **E<sup>2</sup>** up to **8 KBytes**; **Real Time Clock** with Back Up through Lithium battery; **Relays**; **UART CAN**; interface to manage a magnetic strip **Badge reader**, manual or motorized, or Smart Card reader; several communication protocols; etc.

**QTP G28** is able to execute an entire range of display commands, including Clear Screen, LEDs blinking, Position cursor, EEPROM reading or writing, graduated axis drawing, lines and circles drawing, characters zoom, horizontal and vertical characters direction, etc., compatible with the **ADDS View Point** video terminal standard. Maximum specifications for **QTP G28** are:

- Overall dimensions: frontal size 281x139 mm; frontal depth 32 mm; rear depth 26 mm
- Surface or flush panel mounting
- Mounting direct to **Phoenix Combi Card®** industrial instrument cases
- Professional keyboard: **28 Keys** and **16 LEDs**
- **Complete reconfiguration** of all keycodes at User level
- **Label slots** for personalization of 5 keys, 5 LEDs and device name
- Direct management of **16 LEDs** plus **Blinking** attribute
- 11 keys provided with signalation LEDs
- **Graphic LCD Display**, back lighted through a cold cathod fluorescent lamp, **240x128** pixels, keyboard regulated contrast and brightness
- Maximum alphanumeric display: **30** characters per **16** rows
- **Graphic primitives** like circumference, rectangle, line, cartesian axis, arrow, characters **zoom** and **direction**, etc.
- **Buzzer** for **BELL** and Key pressed signalations
- serial **E<sup>2</sup>**, up to 8 Kbytes, to store set-up, messages, key codes etc.
- **RTC** capable to manage leap years, provided with **Lithium** back up battery
- Main serial communication line, **galvanically isolated**, in **RS 232**, **RS 422**, **RS 485**, **Current Loop** or **CAN**
- Auxiliary serial communication line, **galvanically isolated**, in **RS 232**
- Local Setup for selecting communication protocols, keyclick abilitation, etc.
- Software manageable 3 A **Relay** with normally open and normally closed contacts
- Interface to manage a **Badge reader** able to read trace 1, 2 or 3, manual o motorized or Smart Card reader
- On board power supply provided with noise supressor filter, also for small external loads
- **DC** or **AC** supply voltage, up to **24Vac**

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

## DISPLAY

**QTP G28** has a **graphic LCD display**, back lighted by a **cold cathod fluorescent lamp**, with **240x128** pixels that are equivalent to 240 characters at most (**30** column x **16**rows) in alphanumeric mode. The contrast and brightness can be easily adjusted through the keyboard; so the best viewing condition are always obtained, even with a variable external lighting.

Another very important feature of **QTP G28's** display is its wide viewing angle that ensure a sufficient visible contrast ratio from each frontal position. For detailed information on used display please refer to "TECHNICAL SPECIFICATIONS" chapter.

## KEYBOARD

The **QTP G28** has a membrane keyboard with 28 keys located on the right side of the display. All the keys are metallic dome type so they provide a tactile sensation of the key pressed and they withstand the knocks and bumps of industrial life. Eleven of the 28 keys are provided of indicator LEDs and five have user's reconfigurable label on a removable strip; the remaining 23 keys have a standard labels (see figure 2) that satisfy the normal man-machine interface requirements. These keys are equipped with **autorepeat** and they are totally software reconfigurable or on the other hand the code sent in RS 232 when a key is pressed can be changed or disabled. It is also possible to switch on/off the **keyclick** function, i.e the buzzer short activation each time a key is pressed.

## INDICATOR LEDES

**QTP G28** has **16** indicator **LEDES** for different visual signals, that can be enabled, disabled or enabled with **blinking** attribute, through comfortable serial commands. The last functionality is totally autonomous and it doesn't need any intervention on user side. The **QTP G28** has 11 LEDs linked to keys and the remaining 5 LEDs linked to simple user's configurable labels.

The main purpose of LEDs is to show a visual indication about the card's status, making so easier debug and verify operations even from a long distance from the panel, where the display is not readable. For this reason the LEDs are in three different colours (red, yellow and green) to divide different operative situations as, for example, an allarm, an alert, a ready to work, etc.

## BUZZER

The **QTP G28** have a circuitry ables to emit a steady sound based on a capacitive buzzer. The said circuitry can be activated via software through a specific comand for generating a sound beep or it can be linked to a key pressure just to get the **keyclick** function or it can alert on possible alarm, failure, etc.

## SERIAL COMMUNICATION

The **QTP G28** has two separate hardware serial communication lines: the **primary serial line** is used to exchange data with remote units while the **auxiliary serial line** can be used to hardcopy on a serial printer or other representation devices. In default shape both the serial lines are **RS 232** electric configured but, using a proper indication in the ordering phase, the primary serial line can be configured in:

<b>RS 422</b>	->	<b>.RS422</b> option
<b>RS 485</b>	->	<b>.RS485</b> option
<b>Current Loop</b>	->	<b>.CLOOP</b> option

Thanks to these electric interfaces the **QTP G28** can be linked in a network together with other devices of the same or different type. Independently from the used electric interface, the serial lines are always **galvanically isolated** by the power supply voltage of the terminal panel; this feature excludes each noise and unbalanced problems that normally torment the industrial electric installation. The physical protocol of both the serial lines is completely configurable through a dedicated set up modality that let the user select the values listed in "TECHNICAL SPECIFICATIONS" chapter, by the simple keyboard use.

## EEPROM

**QTP G28** have an **EEPROM** (the capacity vary from 512 Bytes to 2 KBytes) for memorizing setup, keys code, communication protocols, operating mode, name, user data, and so on. Many of the stored data have vital importance so a serial EEPROM has been chosen to obtain the best warranties on validity and maintenance of the saved information, even when power supply is not available. 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 all different configuration must be specified from the user, at the moment of the order, by using the following codes:

<b>1024 Bytes EEPROM</b>	->	<b>.EE-08</b> option
<b>2048 Bytes EEPROM</b>	->	<b>.EE-16</b> option

## USER RELAY

**QTP G28** can have, as option, one relay equipped with 3 Ampere contact. This relay can be switched on/off via software commands directly by the user and it makes available, on the appropriate connector, both the normally open and normally closed contact.

For example this option could be necessary when the automatic door opening must be directly controlled by using the terminal placed in the immediate closeness; in this case it will be sufficient a power actuator driven by the **QTP G28** relay.

The code used to order this option is:

**.RELE'**

## REAL TIME CLOCK

**QTP G28** can have, as option, a Real Time Clock backed by on board **Lithium battery**, that manages hours, minutes, seconds, day, month, year and week day. This device is manageable by the user with appropriate software serial commands which allow to set time and date, to read these data or to display them on the display on a given position. This option more than add a complete time

information autonomously managed by **QTP G28**, makes available a serial real time clock to the external master system. This system should control the elapsed time, activate procedures for time based events, calculate production value in a time period, etc.

The code used to order this option is: **.RTC**

### **BADGE READER INTERFACE**

**QTP G28** as option can have a circuitry that allows to interface the terminal to a single track magnetic card reader which can be **insertion** (manual or motor driven) or **sliding** types for reading the track n.1, 2 or 3. **QTP G28** automatically acquires the card and its decodification while the string is memorized into the local memory; in this way the user receives the message already decodified without any other operations. For further information on this options and its availability, please contact directly **grifo®**.

The code used to order this option is: **.BADGE**

### **CAN INTERFACE**

The **QTP G28** can have, as option, of a complete CAN interface that supports the **BasicCAN** and **PeliCAN 2.0B** standards protocol. With this feature the user can afford and solve many problems as: high speed data transfer, long distance communication, autonomous errors management, multimaster and multislave networks support, etc.

Con questa possibilità si possono affrontare e risolvere problemi di trasferimento dati ad alta velocità, comunicazione su lunghe distanze, gestione autonoma degli errori, supporto di reti multimaster e multislave, ecc.

The code used to order this option is: **.CAN**

### **ON BOARD POWER SUPPLY**

One of the most important peculiarity of **QTP G28** is its own switching power supply that requires an input voltage variable from **8 to 24 Vac** or **12÷34 Vdc**. This section generates two **galvanically isolated** voltages: one for the on board logic and one for the serial communication interfaces. The main **5 Vdc** power supply can be used to power small external loads directly from the same terminal, through a dedicated connector. For further information on power supply section, please refer to "ELECTRIC FEATURES" paragraph.

## TECHNICAL SPECIFICATIONS

### GENERAL FEATURES

**Resources:** IP65 frontal  
16 LEDs, 11 of them associated to keys  
28 software reconfigurable keys, 5 can be labelled on a removable strip by the user  
Graphic LCD Display, back lighted through a cold cathod fluorescent lamp, 240x128 pixels (30x16 characters)  
Buzzer for BELL and keyclick  
2 galvanically isolated serial lines, full duplex, RS 232 buffered, one can be buffered in RS 422, RS 485 or Current Loop (option)  
EEPROM up to 2 KBytes for configuration, keys code, etc.  
Real Time Clock backed with Lithium battery (option)  
Magnetic cards reader for one track badge (option)  
CAN interface (option)  
User relay with changeover contacts (option)

**CPU:** Z180 with 18.432 MHz crystal

**Primary serial line A:** Baud Rate: 1200, 2400, 4800, 9600, 19200, 57600 Baud  
Stop Bit: 1, 2  
Parity: off, odd, even  
Chr lenght: 8 Bits (normal com.), 9 Bits (master slave com.)

**Auxiliary serial line B:** Baud Rate: 1200, 2400, 4800, 9600, 19200, 57600 Baud  
Stop Bit: 1, 2  
Parity: off, odd, even  
Chr lenght: 8 Bits

**Receive buffer dimension:** 3200 Bytes

### PHYSICAL FEATURES

**Dimension:** See figure 1

**Weight:** 1380 grammi

**Mounting:** On panel as front panel and back panel or direct mounting on **Phoenix CombiCard** industrial housings

**Display** **brightness, contrast** Adjustable with keyboard  
**visible area:** 126 x 71 mm  
**pixel size:** 0,5 x 0,5 mm  
**character size:** 8 x 8 pixels in alphanumeric mode  
8 x 8, 8 x 16, 16 x 8, 16 x 16, 24 x 24, 32 x 32, 40 x 40 pixels in graphic mode



<b>Temperature:</b>	0÷50 C°
<b>Humidity:</b>	20% up to 90% (without condense)
<b>Connectors:</b>	CN1: quick release, screw terminal, 3 pins, male CN2: low profile, vertical, 2 pins, male CN3A: plug, 6 pins, female CN3B: plug, 6 pins, female CN4: quick release, screw terminal, 3 pins, male CN6: low profile, vertical, 2 pins, male
<b>Timing:</b>	LEDs blinking 500 msec key autorepeat 100 msec; 500 msec when the key is just pressed key debouncing 20 msec

**OVERALL DIMENSION**

Here is the **QTP G28** dimension (in millimetres) of rear and lateral sides, where connectors are located. This is not a scale drawing.

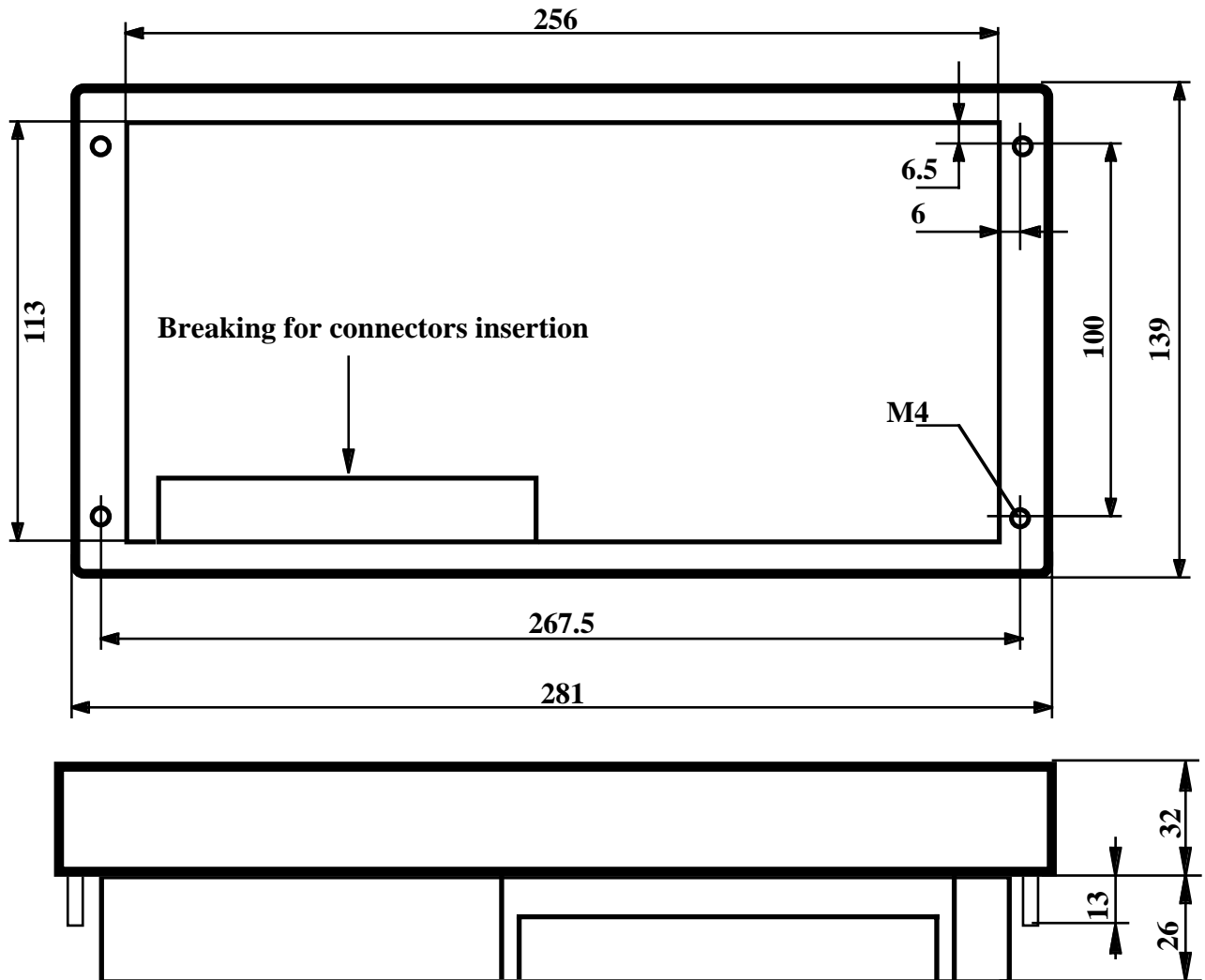


FIGURE 1: TERMINAL DIMENSION

**ELECTRIC FEATURES**

<b>Power supply input volatge:</b>	8÷24 Vac, or 12÷34 Vdc
<b>Power supply power:</b>	7.5 W
<b>Consumption base configuration:</b>	6.41 W
<b>max configuration:</b>	6.97 W
<b>Voltage for external loads:</b>	5 Vdc
<b>Power for external loads:</b>	7.5 W - consumption
<b>Max current on relay contacts:</b>	3 A
<b>Max voltage on relay contacts:</b>	30 V
<b>RS 422, RS 485 termination network:</b>	positive pull up resistor: <b>3.3 K<math>\Omega</math></b> negative pull down resistor: <b>3.3 K<math>\Omega</math></b> line termination resistor <b>120 <math>\Omega</math></b>

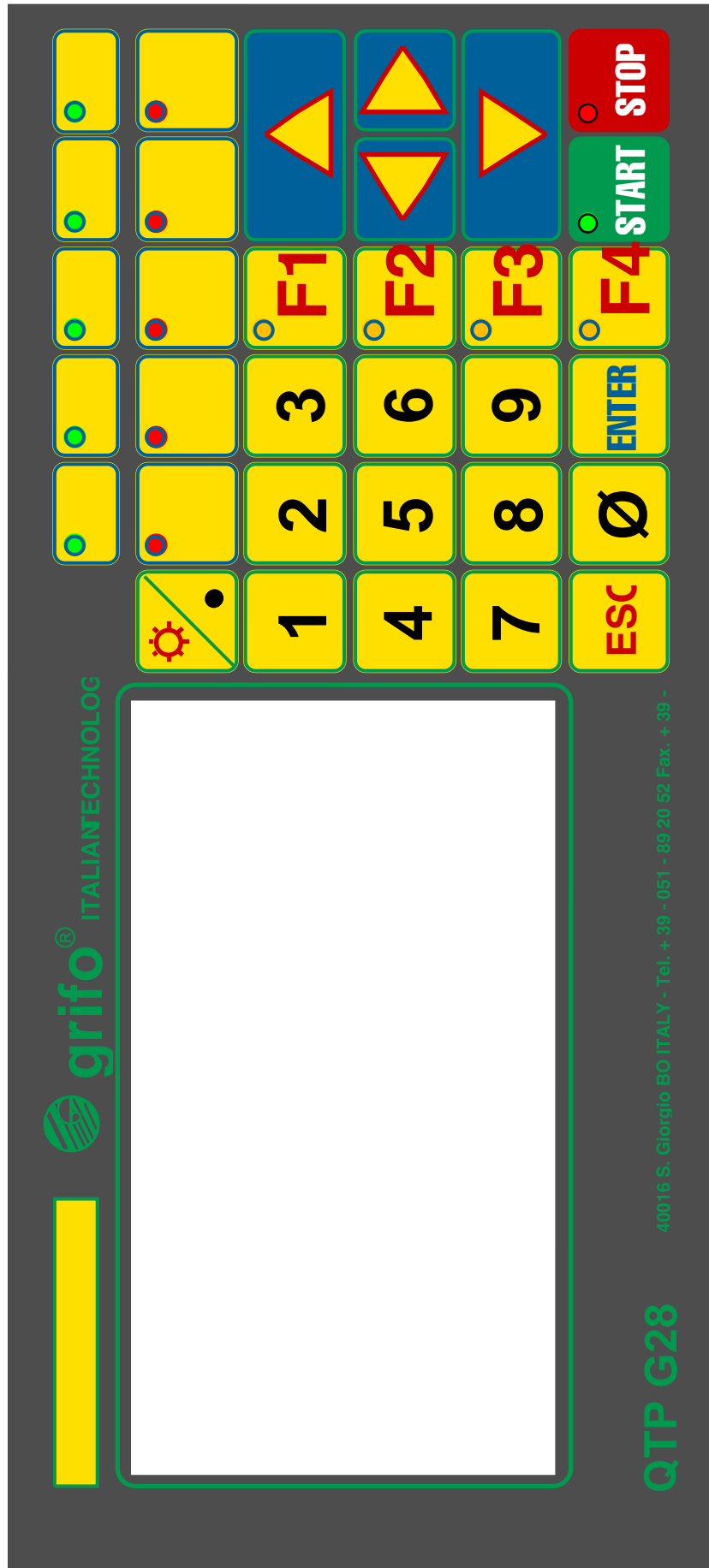


FIGURE 2: FRONTAL VIEW

## INSTALLATION

In this chapter there are the information for a right installation and correct use of the terminal **QTP G28**. In detail there are the locations and functions of each connector and of the user settable jumpers; for the connectors it is described the the pin outs, the meaning of the connected signals and some connection examples, that simplify and speed the installation phase.

### CN1 - POWER SUPPLY CONNECTOR

CN1 is a quick release, screw terminal, male connector with 3 pins.

On CN1 must be connected the single power supply voltage for the terminal and the possible ground connection.

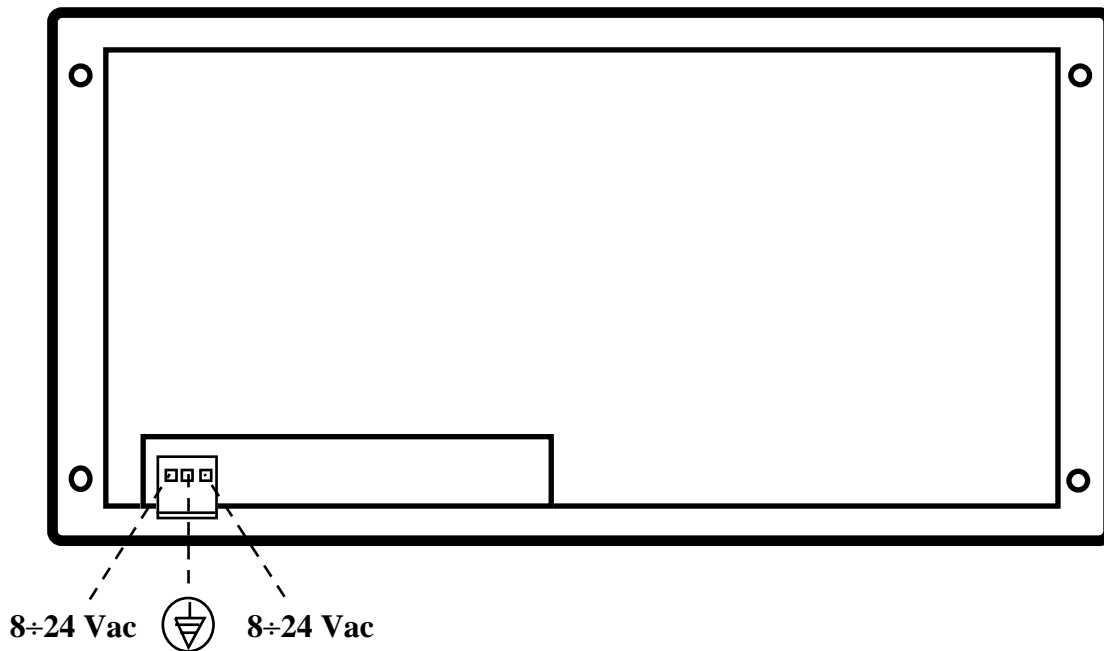


FIGURE 3: CN1 - POWER SUPPLY CONNECTOR

Signals description:

**8÷24 Vac** = I - Power supply lines connected to on board switching section; these signals coincide with an **8÷24 Vac** alternate voltage or equal **12÷34 Vdc** direct voltage.



= - Earth line.

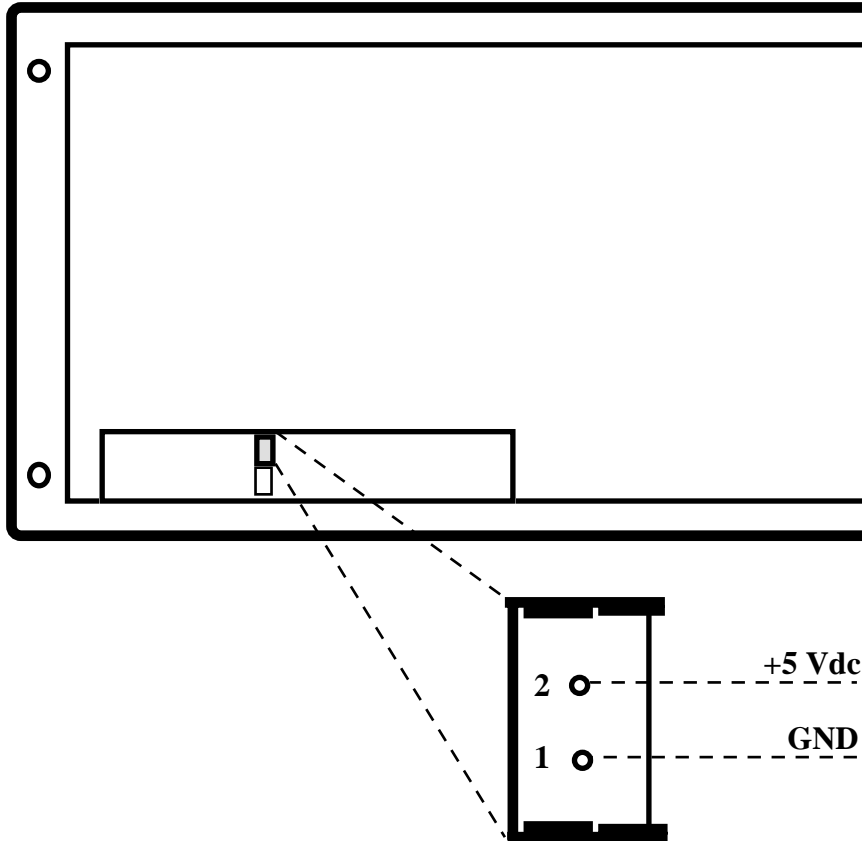
### **N.B.**

On the earth line it is connected the terminal's anti noise filter, so if this protection is necessary the relative pin must be connected to earth signal of mains electric installation; viceversa if this line is not connected, the anti noise filter has no effect.

## CN2 - EXTERNAL LOADS POWER SUPPLY CONNECTOR

CN2 is a 2 pins, male, vertical, low profile connector with 2,54 mm pitch.

Through CN2 the user can supply the on board +5 Vdc voltage to supply external loads up to 1 W maximum consumption.



**FIGURE 4: CN2 - EXTERNAL LOADS POWER SUPPLY CONNECTOR**

Signals description:

**+5 Vdc** = O - Power supply signal of the on board logic at +5 Vdc.  
**GND** = - Ground signal

### **N.B.**

If **QTP G28** is powered with a direct voltage (on CN1 connector), the ground signal of this power supply is different from the GND ground signal available on CN2.

The power supply signals of the on board logic (main) are galvanically isolated from the power supply voltage of the serial communication interfaces (secondary). For this reason the GND signal available on CN2 must not be connected to GND\_C signal available on CN3A and CN3B connectors.

**CN3A - PRIMARY SERIAL LINE CONNECTOR**

CN3A is a 6 pins, female, plug connector.

Through CN3A the user serially communicates with the terminal by using one of the standard electric standards RS 232, RS 422, RS 485 or Current Loop (active or passive). Placing of the signals has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the electric protocols follow the CCITT normative.

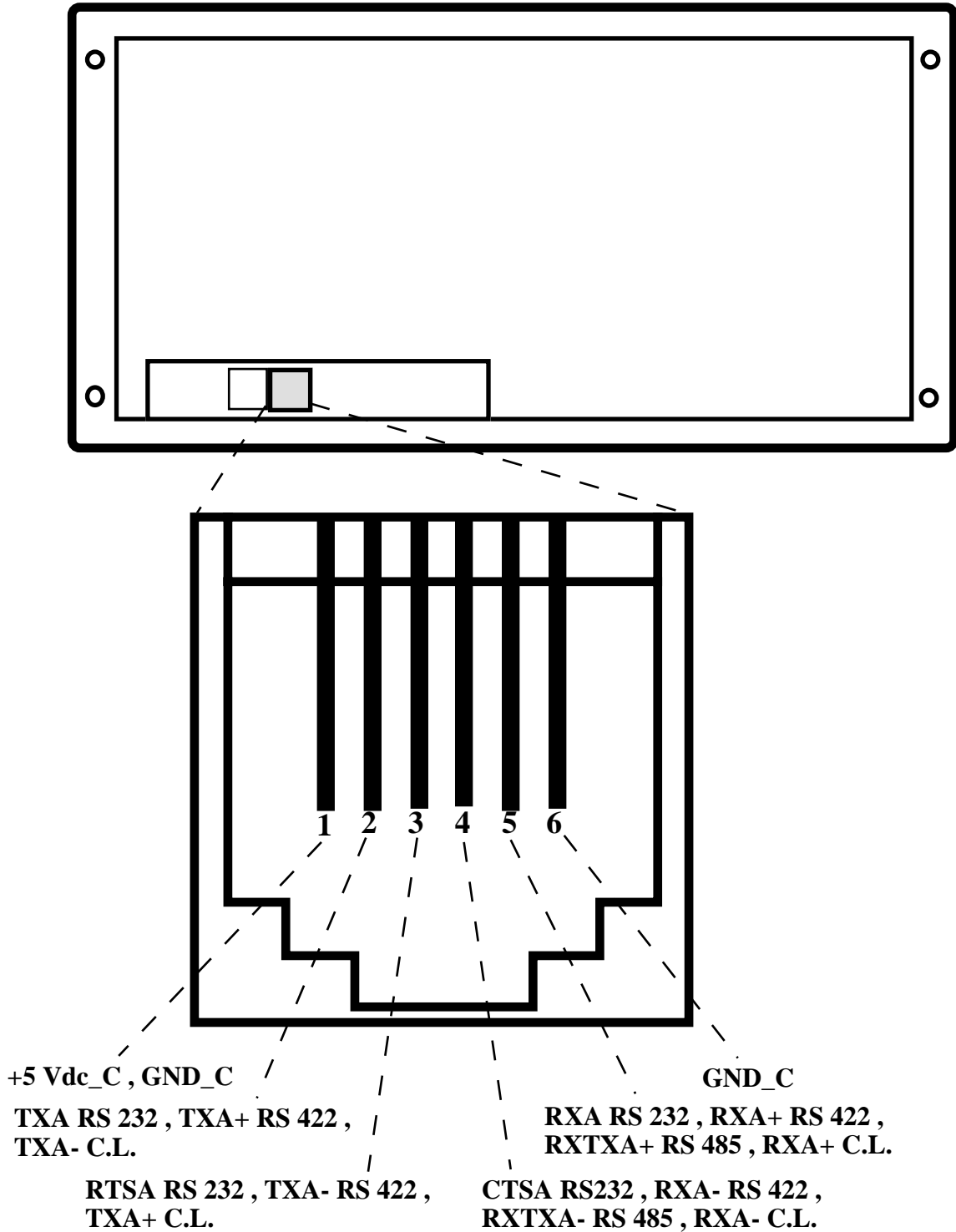


FIGURE 5: CN3A - PRIMARY SERIAL LINE CONNECTOR

Signals description:

- RXA RS 232** = I - Serial line A=primary RS 232 Receive Data signal.
- TXA RS 232** = O - Serial line A=primary RS 232 Transmit Data signal.
- CTSA RS 232** = I - Serial line A=primary RS 232 Clear To Send signal (see paragraph "JUMPERS").
- RTSA RS 232** = O - Serial line A=primary RS 232 Request To Send signal.
- RXA- RS 422** = I - Receive Data Negative: Serial line A=primary negative signal for RS 422 serial differential receive.
- RXA+ RS 422** = I - Receive Data Positive: Serial line A=primary positive signal for RS 422 serial differential receive.
- TXA- RS 422** = O - Transmit Data Negative: Serial line A=primary negative signal for RS 422 serial differential transmit.
- TXA+ RS 422** = O - Transmit Data Positive: Serial line A=primary positive signal for RS 422 serial differential transmit.
- RXTXA- RS 485** =I/O- Receive Transmit Data Negative: Serial line A=primary negative signal for RS 485 serial differential receive and transmit.
- RXTXA+ RS 485** =I/O- Receive Transmit Data Positive: Serial line A=primary positive signal for RS 485 serial differential receive and transmit.
- RXA- C.L.** = I - Receive Data Negative: Serial line A=primary negative signal for Current Loop serial bipolar receive.
- RXA+ C.L.** = I - Receive Data Positive: Serial line A=primary positive signal for Current Loop serial bipolar receive.
- TXA- C.L.** = O - Transmit Data Negative: Serial line A=primary negative signal for Current Loop serial bipolar transmit.
- TXA+ C.L.** = O - Transmit Data Positive: Serial line A=primary positive signal for Current Loop serial bipolar transmit.
- +5 Vdc\_C** = O - +5 Vdc power supply of the communication interfaces.
- GND\_C** = - Ground signal of the communication interfaces.

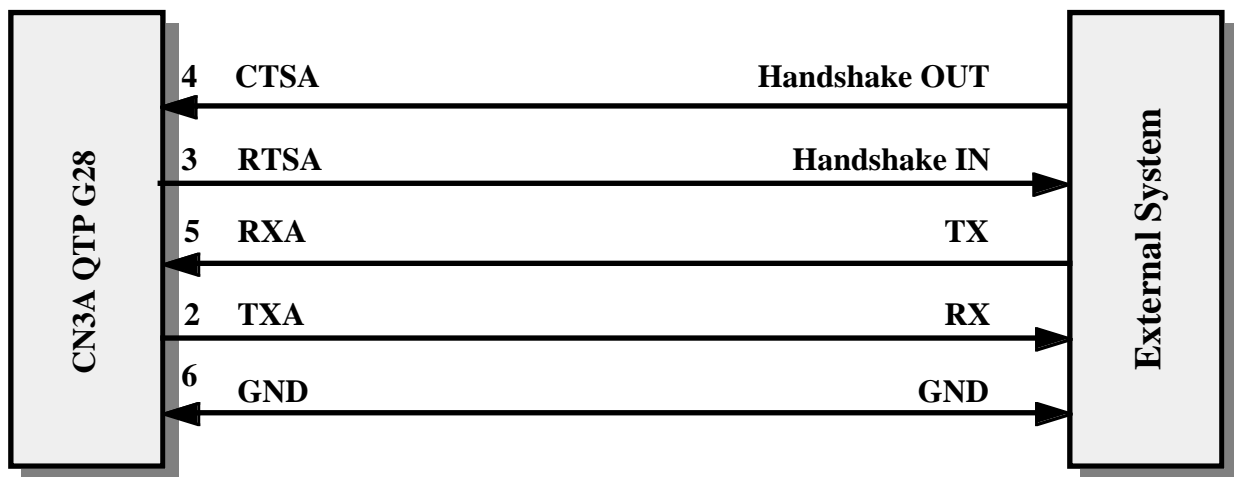


FIGURE 6: RS 232 PRIMARY SERIAL CONNECTION EXAMPLE

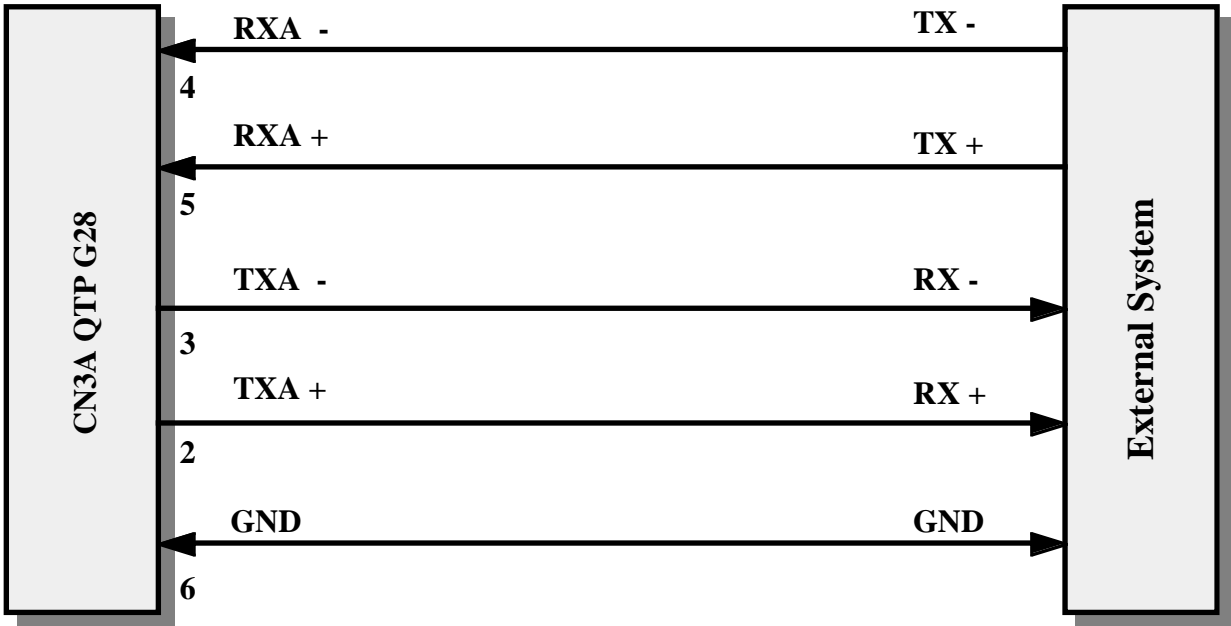


FIGURE 7: RS 422 POINT TO POINT CONNECTION EXAMPLE

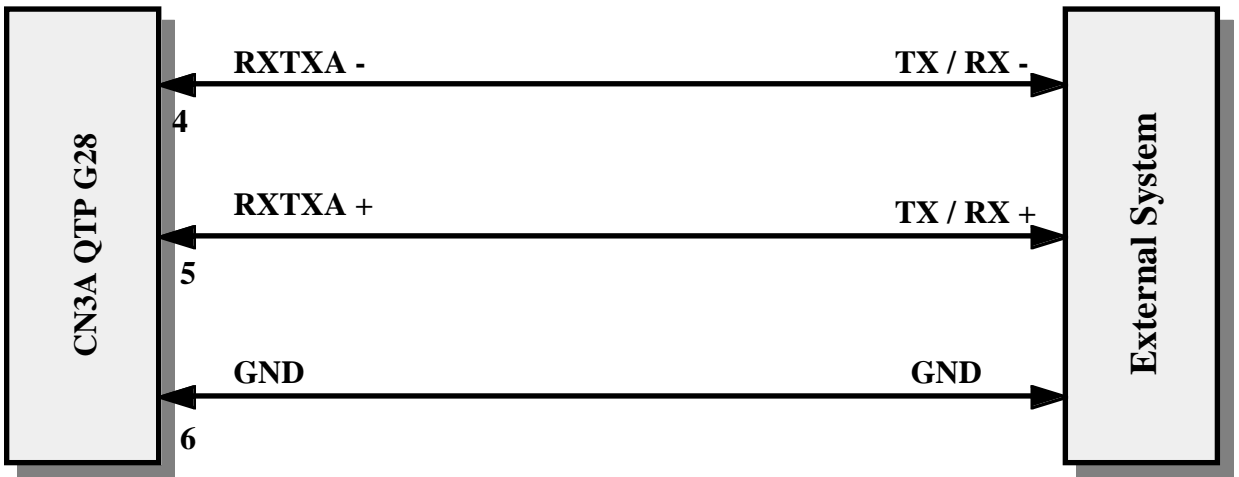


FIGURE 8: RS 485 POINT TO POINT CONNECTION EXAMPLE



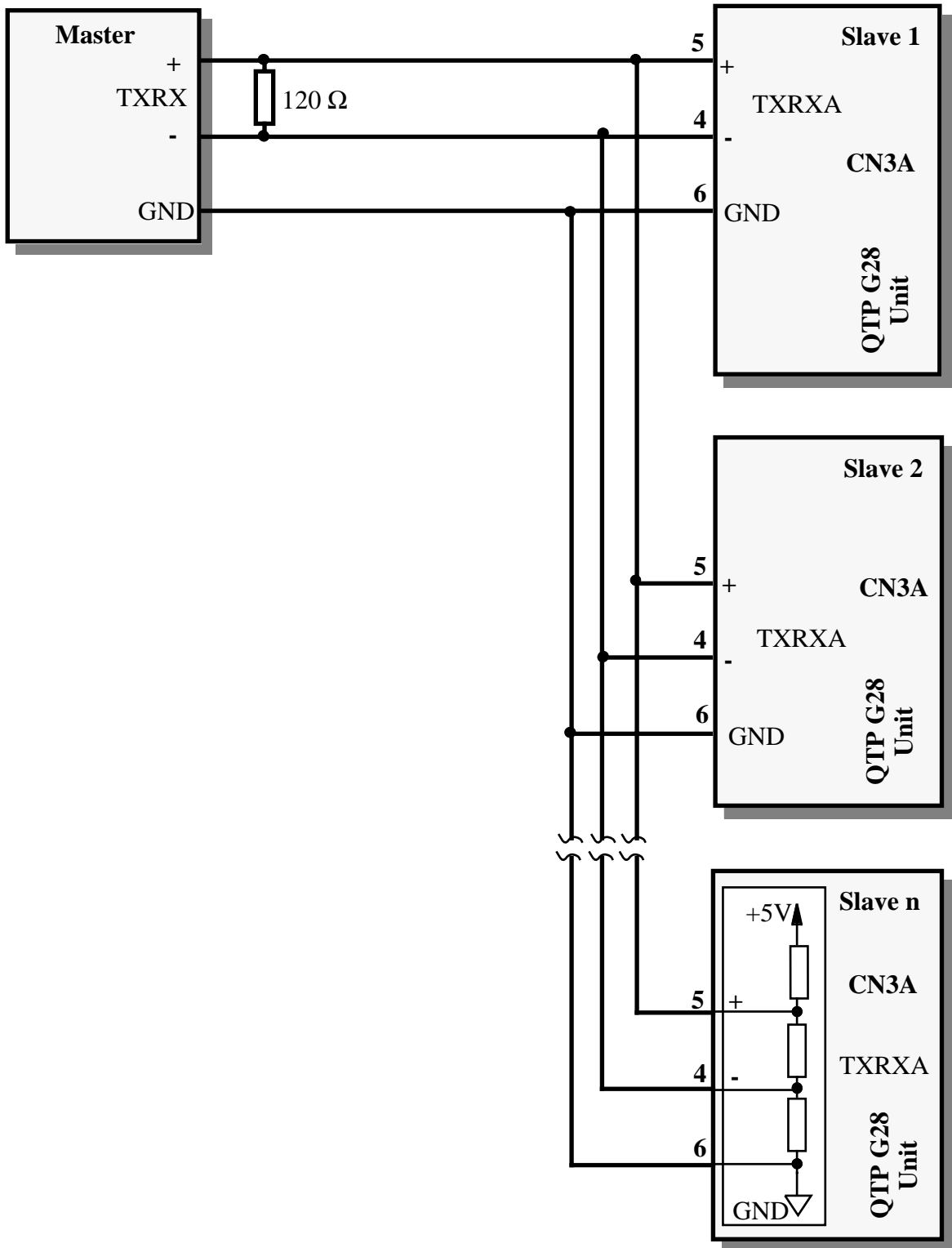


FIGURE 9: RS 485 NETWORK CONNECTION EXAMPLE

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 extremities, respectively near the master unit and the slave unit at the greatest distance from the master.

Forcing and terminating circuitry is installed on QTP G28 board; it can be enabled or disabled through specific jumpers, as explained later, in paragraph "JUMPERS".

For further information please refer to TEXAS INSTRUMENTS data book, "RS 422 and RS 485 Interface Circuits", the introduction about RS 422-485.

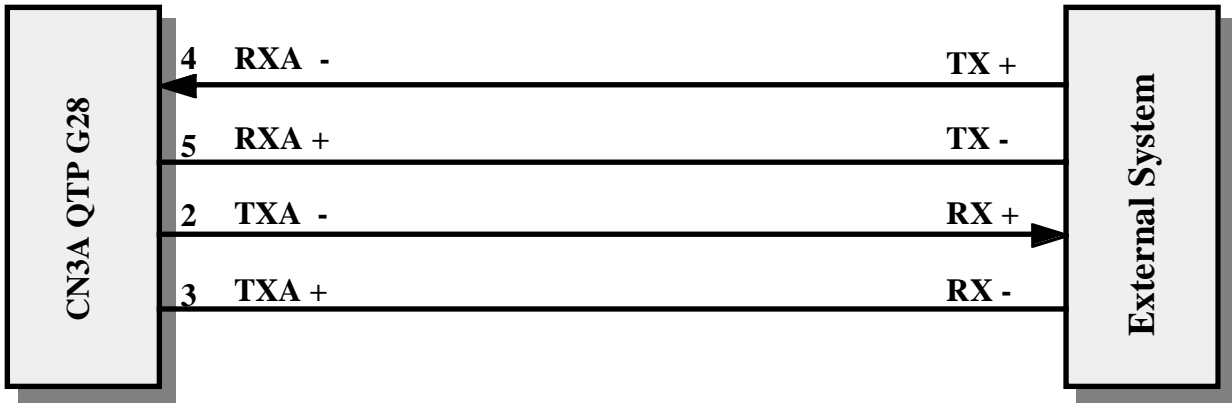


FIGURE 10: 4 WIRES ACTIVE CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

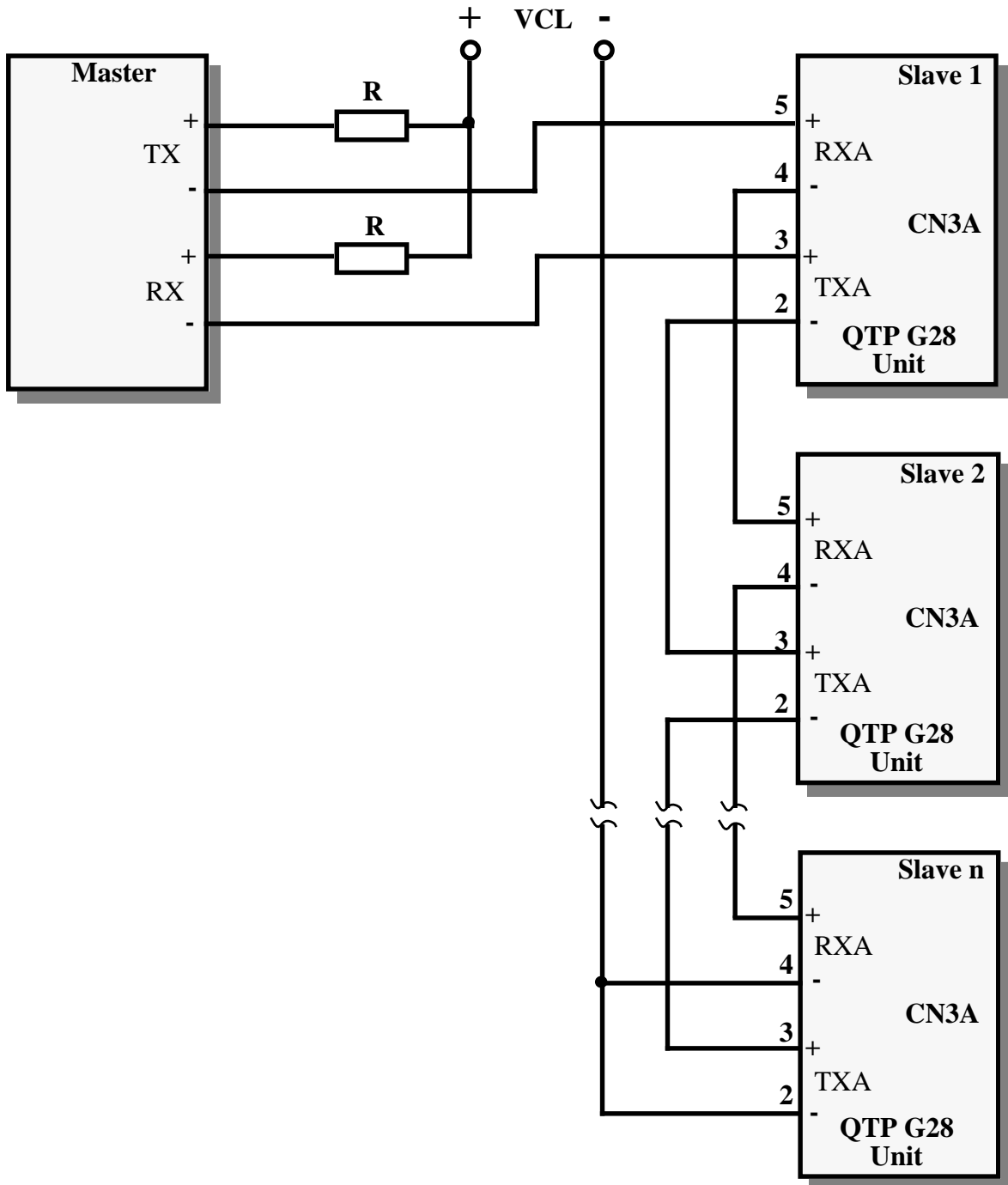


FIGURE 11: 4 WIRES PASSIVE CURRENT LOOP NETWORK CONNECTION EXAMPLE

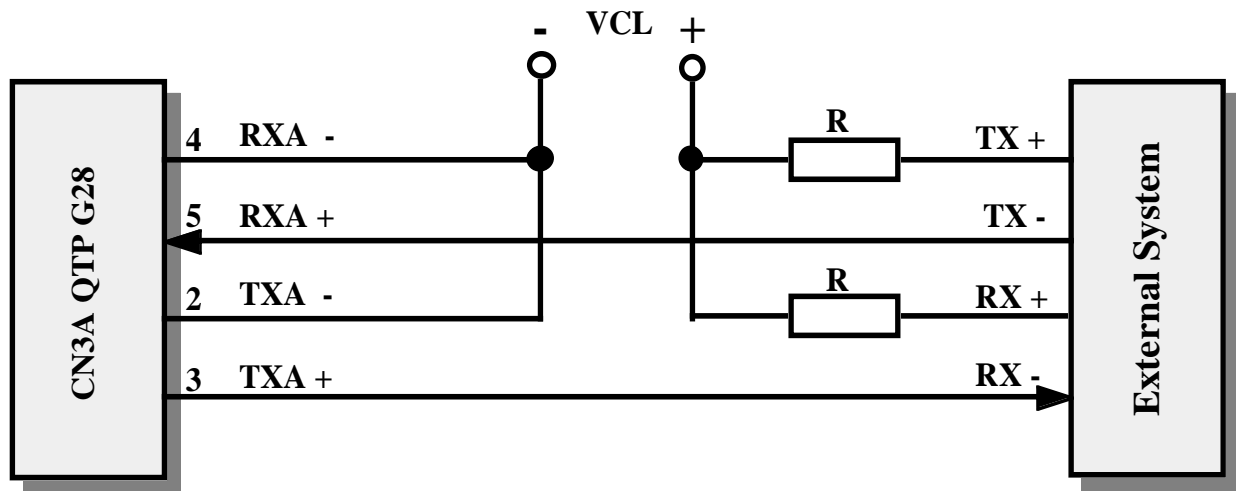


FIGURE 12: 4 WIRES PASSIVE CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

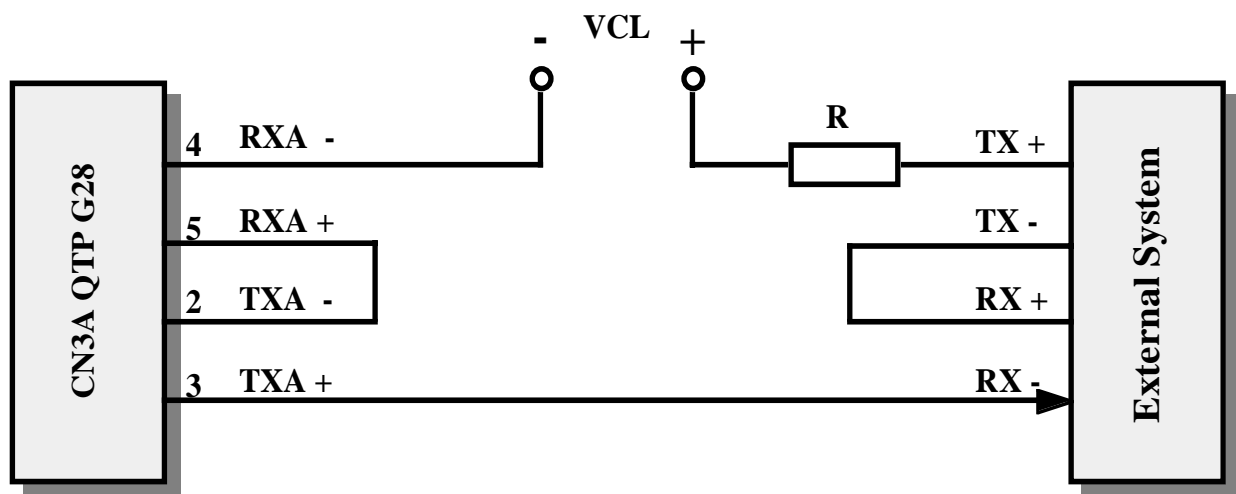


FIGURE 13: 2 WIRES PASSIVE CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

There are two possible passive Current Loop connections: 2 wires and 4 wires. These connections are shown in figures 11÷13 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 considering 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 shortcircuit each transmitter can dissipate at most **125 mW**;
- in case of shortcircuit each receiver can dissipate at most **90 mW**.

The **QTP G28** is provided of an indeed current generator that make active the Current Loop interface for point to point connection (see figure 10). This feature allows a serial communication even on a long and noised connection with a very low price in fact it is not necessary to add an external current generator. The on board power supply for the active Current Loop interface is galvanically isolated and it is dimensioned for two devices only, so it can be used exclusively in 4 wires point to point connection.

The operations required to configure le Current Loop serial line are described in the "JUMPERS" paragraph. For further info please refer to HEWLETT-PACKARD data book, (**HCPL 4100** and **4200** optocoupler devices).

**CN3B - AUXILIARY SERIAL LINE CONNECTOR**

CN3B is a 6 pins, female, plug connector.

On CN3B is available an RS 232 buffered serial line that is managed by the terminal for retransmission functions towards another external serial system. The typical applications for this auxiliary serial line are the management of a serial printer, both in hardcopy or transparent mode. Placing of the signals has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the electric protocols follow the CCITT normative.

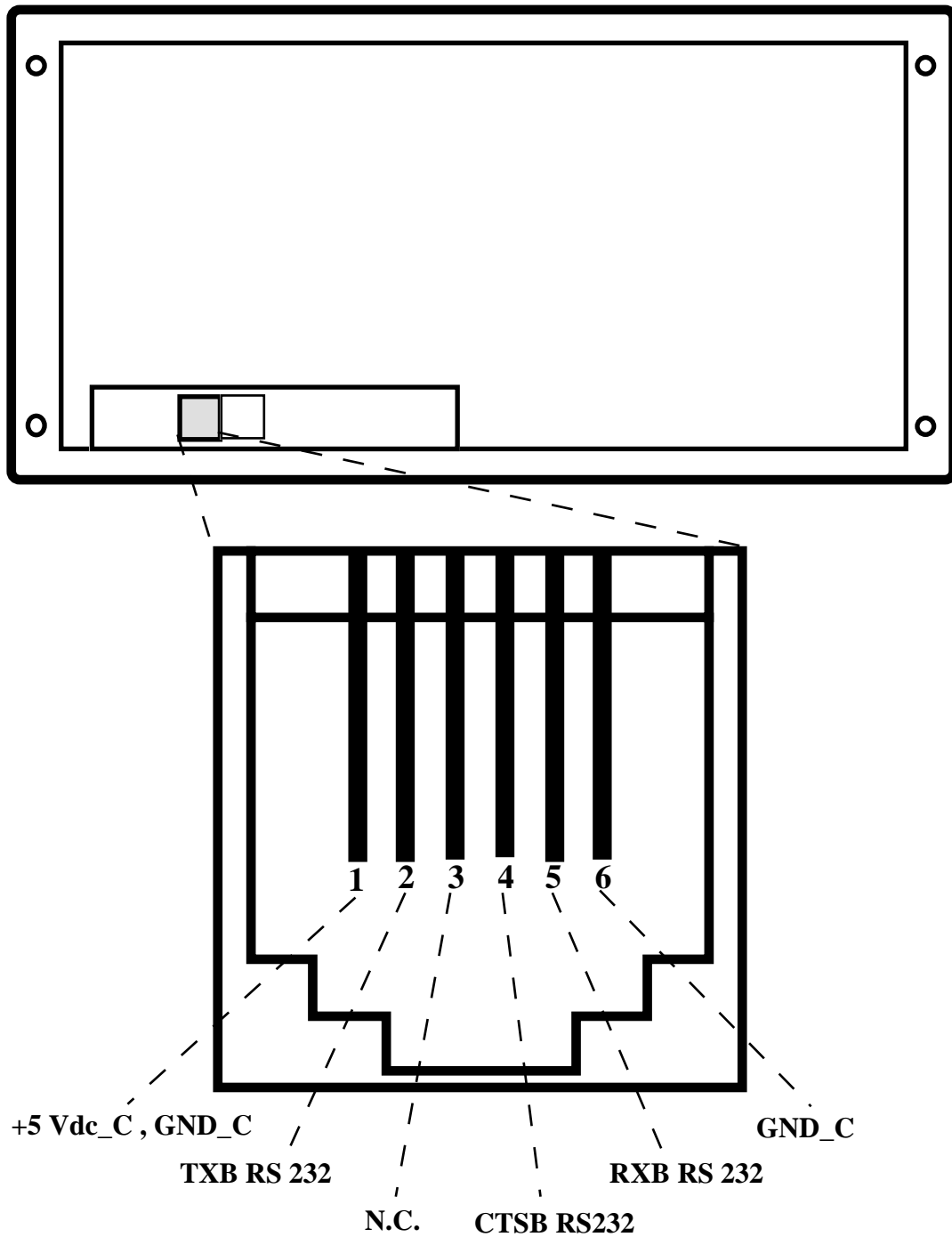


FIGURE 14: CN3B - AUXILIARY SERIAL LINE CONNECTOR

Signal description:

- RXB RS 232** = I - Serial line B=auxiliary RS 232 Receive Data signal.
- TXB RS 232** = O - Serial line B=auxiliary RS 232 Transmit Data signal.
- CTSB RS 232** = I - Serial line B=auxiliary RS 232 Clear To Send signal.
- +5 Vdc\_C** = O - +5 Vdc power supply of the communication interfaces.
- GND\_C** = - Ground signal of the communication interfaces.
- N.C.** = - Not connected.

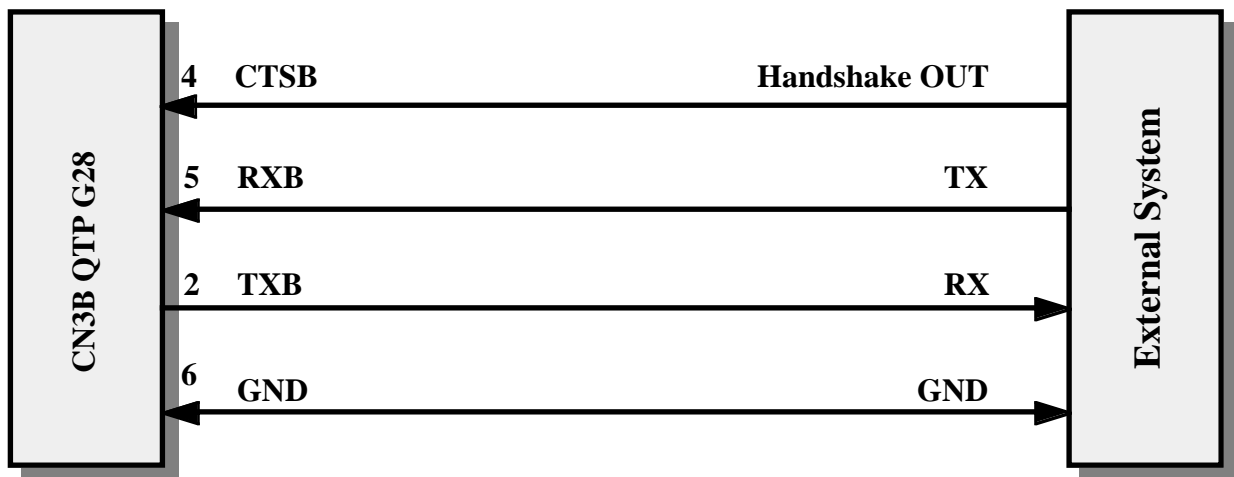


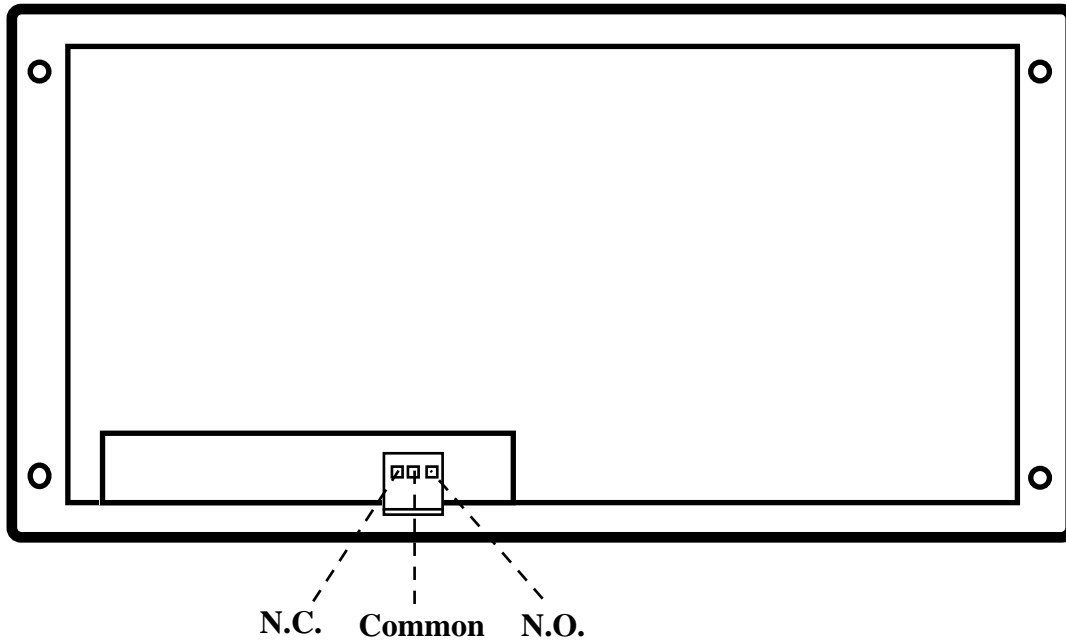
FIGURE 15: RS 232 AUXILIARY SERIAL CONNECTION EXAMPLE

**CN4 - USER RELAY CONNECTOR**

CN4 is a quick release, screw terminal, male connector with 3 pins.

On the connector there are the contacts normally opened, normally closed and common of the relay as shown in the drawings here under. Please remind that the described relay supports a maximum load of **3 Ampere** on a **30 Vdc** voltage.

The connector CN4 is included in the **.RELE'** option and it is available only when the option is ordered.



**FIGURE 16: USER RELAY CONNECTOR**

Signals description:

- Common** = Relay common contact
- N.C.** = Relay contact normally closed
- N.O.** = Relay contact normally opened

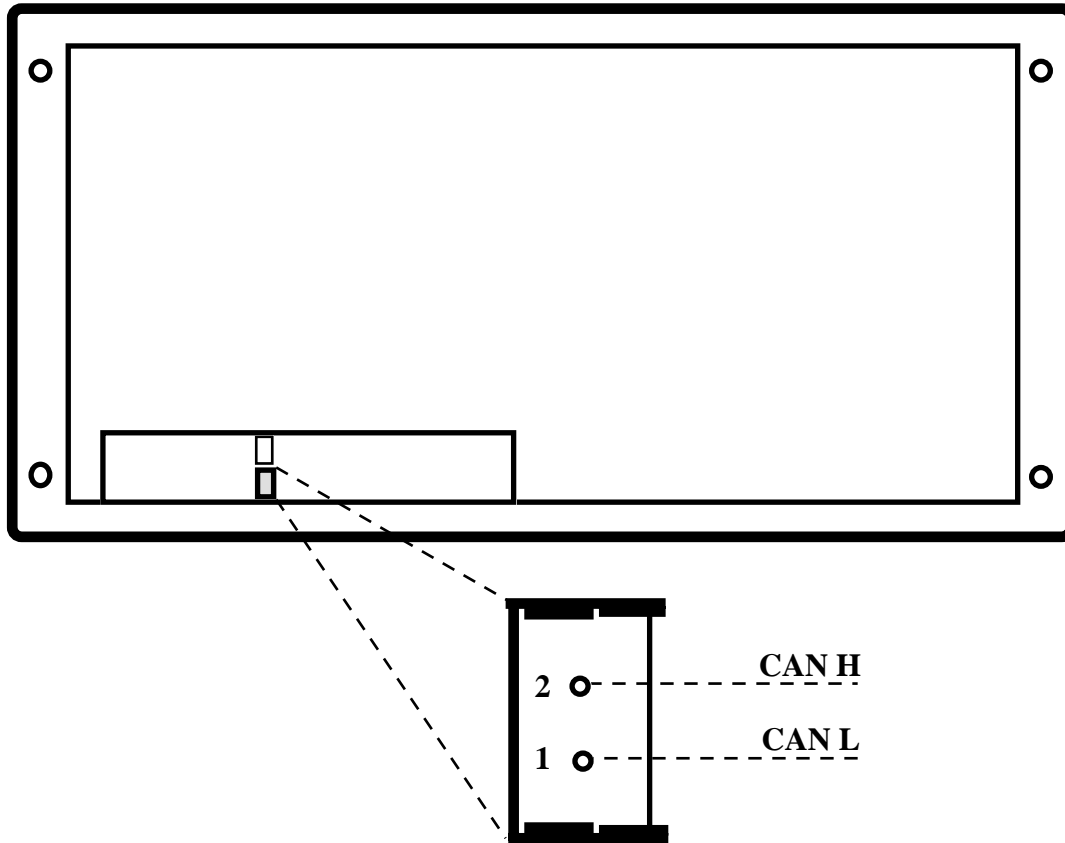


FIGURE 17: PHOTO

**CN6 - CAN INTERFACE CONNECTOR**

CN6 is a 2 pins, male, vertical, low profile connector with 2,54 mm pitch.

Through CN6 must be connected the CAN serial communication line by following the standard rules defined by the same protocol.



**FIGURE 18: CN6 - CAN INTERFACE CONNECTOR**

Signals description:

**CANH** = I/O - Differential line high for CAN interface.

**CANL** = I/O - Differential line low for CAN interface.



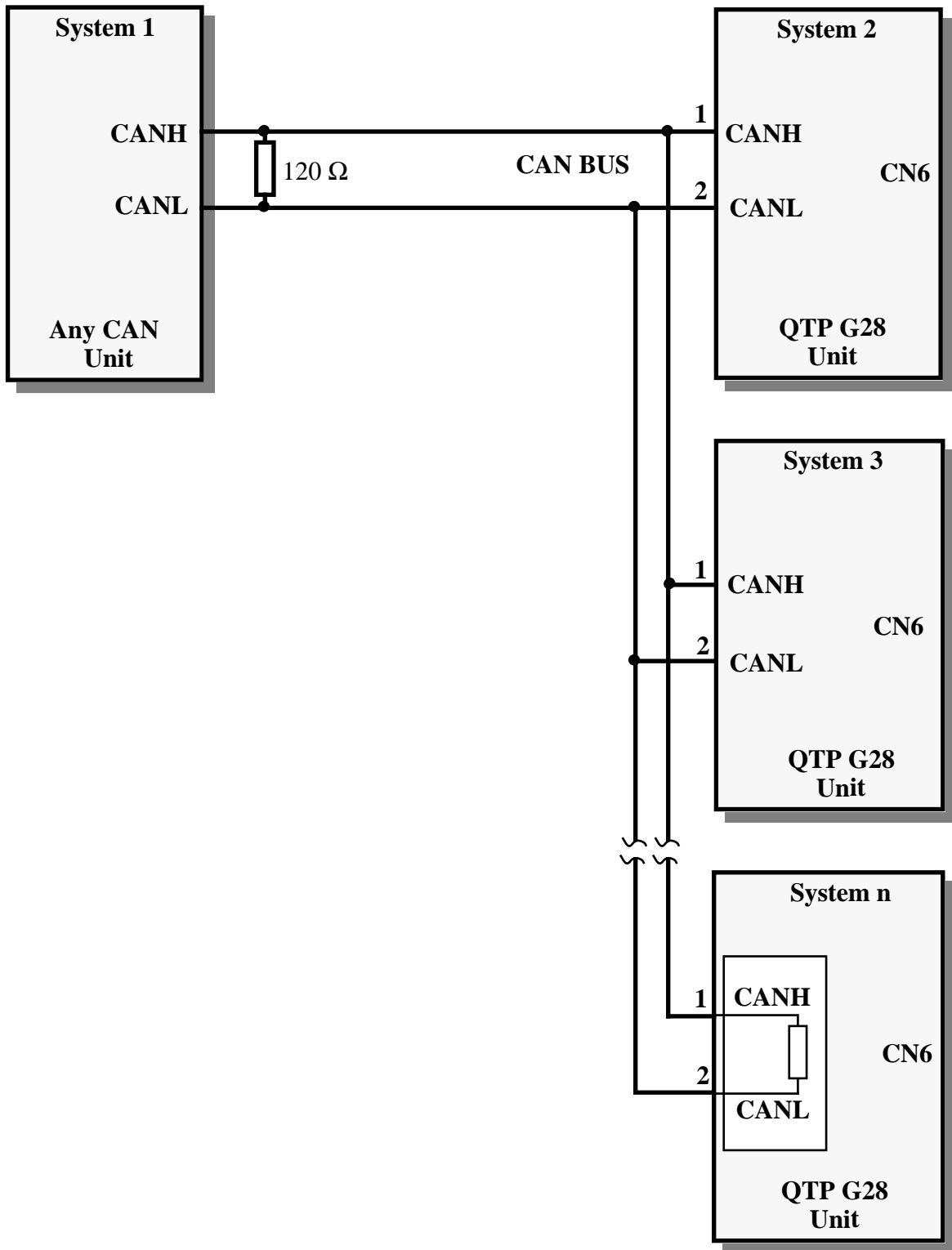


FIGURE 19: CAN INTERFACE CONNECTION EXAMPLE

Please remind that a CAN network must have two termination resistors (120 Ω) placed at its extremities, respectively near the master unit and the slave unit at the greatest distance from the master. The proper termination of the CAN line ensures the right functionality of the communication, in fact the CAN line interface avoids transients and it is immune to radio and electromagnetic noises only when the field connections are correctly performed.

On QTP G28 board the terminating circuitry is already installed: it can be connected or not through specific jumper, as explained later, in paragraph "JUMPERS".

## JUMPERS

On **QTP G28** there are some jumpers (both solder and strip type) for card configuration. In the following figures there is the list, the position and the functionalities of the jumpers modifiable by the user; please remind that the jumpers not described in the following chapter absolutely can't be modified by the user, to avoid terminal malfunction punishment.

JUMPERS	CONNECTION	USE	DEF.
J1 J2	not connected connected	They don't connect the forcing and terminating circuit to the RS 422, RS 485 serial line. They connect the forcing and terminating circuit to the RS 422, RS 485 serial line.	*
J3 J4	position 1-2 position 1-1	They configure the Current Loop receiver as active interface. They configure the Current Loop receiver as passive interface.	*
J5 J6	position 1-2 position 1-1	They configure the Current Loop transmitter as active interface. They configure the Current Loop transmitter as passive interface.	*
J8	not connected connected	The hardware handshake CTSA RS232 of primary serial line is driven by homonymous signal on CN3A. It forces active the hardware handshake CTSA RS232 of primary serial line.	*
J9	not connected connected	It doesn't connect the 120 $\Omega$ terminating resistor to CAN line. It connects the 120 $\Omega$ terminating resistor to CAN line.	*

**FIGURE 20: JUMPERS TABLE**

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.

To recognize these valid connections, please refer to the board printed diagram (serigraph) or to figures 21 of this manual, where the pins numeration is listed. Obviously the **QTP G28** jumpers can be reached only by opening the same terminal; this open operation separates the front panel from the rear carter and it is simply executed by removing the 4 black screws on the four corner of the frontal panel.

The connection in position **1-1** described for J3, J4 and J5, J6 jumpers, coincides with a single 2 pins strip header that connects the pin 1 of the jumpers (i.e. the strip header must be connected among pin 1 of J3 and pin 1 of J4); while the position 1-2 coincides with two separate 2 pins header that connect pins 1 and 2 of the jumpers (i.e. one streap header on J3 and another streap header on J4).

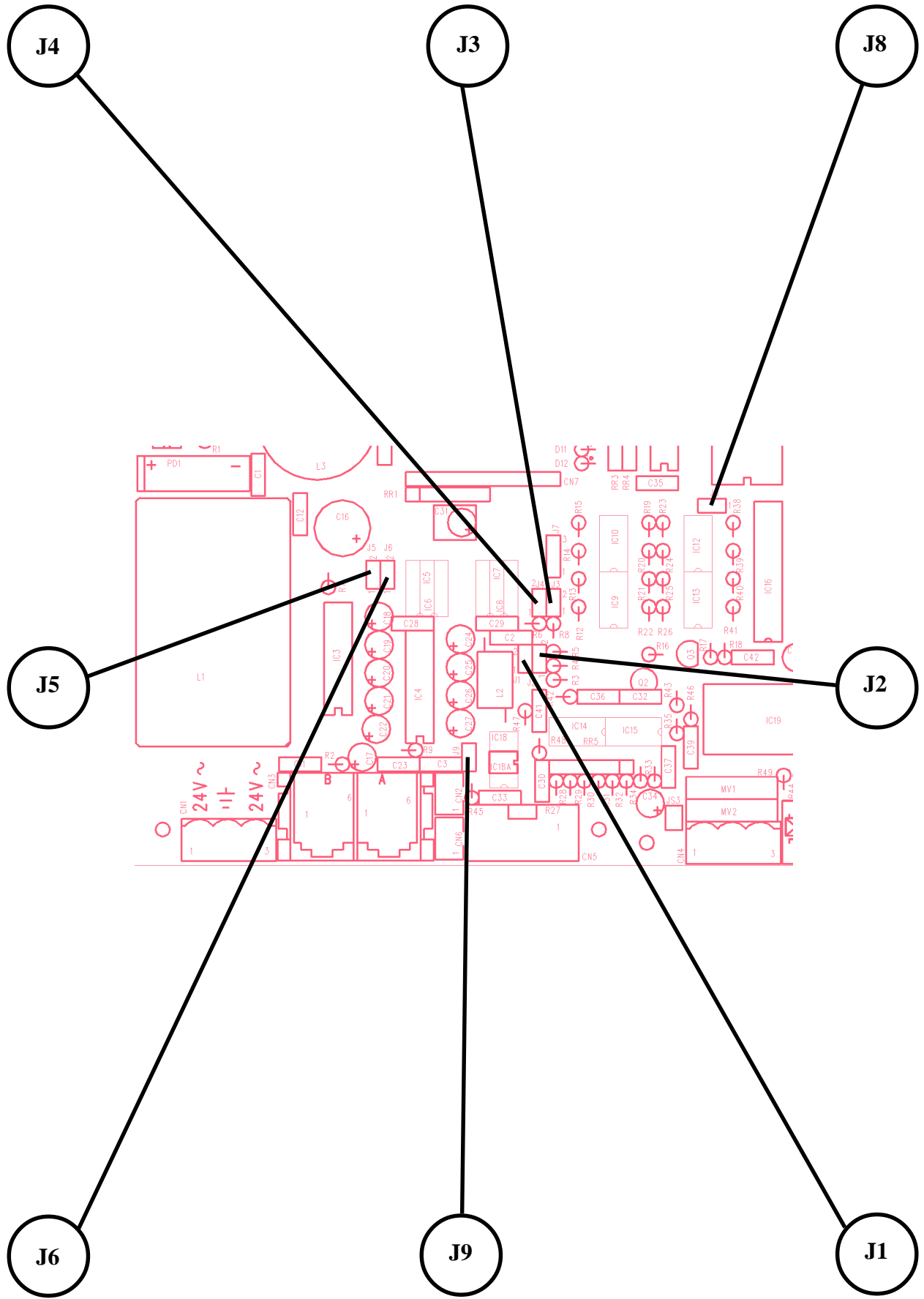


FIGURE 21: JUMPERS LOCATION

## SOFTWARE DESCRIPTION

**QTP G28** terminal is a complete video terminal and for this reason anything received through serial line, if it is not a command, is shown on the display while codes of any pressed keys of the onboard keyboard, are transmitted to the control master unit. On board of this terminal panel is also implemented a local set up program which allows to set the communication protocols by using the **QTP**'s keyboard and display. This manual contains, in addition to the description of the different functions, a complete list of the command sequences and the recognized combination used to benefit of the main features of **QTP G28**. For each code or codes sequence, there is a double description: 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 program and the **QTP G28** keyboard and display, it is possible to directly set the communication protocol parameter, the keyclick function, the representation and communication modality, etc.

To enter the set up mode the user must power on the **QTP G28** and simultaneously he must press the keys number 30 and 2 (see figure 23) for at least two seconds. At this point the display shows the following screen:

#### - LOCAL SETUP V. x.y -

```

COMUNICATION MODE :_ Normal (M. Slave)
NAME : 255 (0÷255)
KEYCLICK : ON (OFF)
DISPLAY MODE : Graphic (Alphanum)
BAUD RATE PRIMARY : 19200 (1200 2400 4800 9600 57600)
STOP BITS PRIMARY : 1 (2)
PARITY PRIMARY : OFF (Odd Even) *
BAUD RATE AUX. : 1200 (2400 4800 9600 19200 57600)
STOP BITS AUX. : 1 (2)
PARITY AUX. : OFF (Odd Even)
CAN INTERFACE : Disable (Low level)
ARROWS UP/DOWN = Change line
ARROWS LEFT/RIGHT=Change value
STOP = Exit
    
```

FIGURE 22: LOCAL SET UP SCREEN

Where the x.y notation is replaced by the firmware version number of the terminal, the indications reported on the right side of each entry coincide with the default settings and the indications inside round brackets list the other possible settings for each item.

\* Remind that when M. Slave communication is selected it is not possible to use the parity on the primary serial line, so when this communication mode is enabled the setting of "PARITY PRIMARY" is always OFF and it is not modifiable.

At the beginning the cursor is placed on the first entry and to select the other entries it is sufficient to move the cursor through the UP, DOWN arrow keys (figure 23, numbers 12 and 9) while to change the current setting the LEFT, RIGHT arrow keys (figure 23, numbers 10, 11) can be used. At the end the STOP key (figure 23, number 27) can be pressed to confirm and save all the current settings and to exit from set up program.

For a detailed description of the local set up entries and of the available values, please refer to proper following paragraphs.

### **N.B.**

The local set up mode can be entered only during power on sequence if keys number 30 and 2 are simultaneously pressed for at least two seconds. During normal work of **QTP G28** the pressure of these two keys will have no effects.

## **CAN INTERFACE**

The CAN interface, available on **QTP G28**, is managed by the firmware in two distinct modes:

**Disable** the CAN interface is disabled and it can't be used;

**Low level** the CAN interface is available for the external master unit in a transparent manner. In this condition the unit that communicates with **QTP G28**, through the primary serial line, is capable to use the CAN interface as if it is an own peripheral and therefore it can communicate on CAN networks in each modality. By using the available CAN commands the user can initialize the interface, transmit and receive messages, define communication masks, select the transfer rate, manage the communication protocol, etc.

The possibility to use the CAN interface as primary communication line will be added in the future firmware versions, as well as other management modalities specifically required by the user.

The selection of the CAN interface mode is managed only inside the local set up program, as described in the homonymous paragraph.

## **VISUALIZATION MODE**

The **QTP G28** provides two different representation mode on the display:

**Alphanum** the visualization on display is exclusively alphanumeric with a maximum of 30 per 16 characters of 8x8 pixels. The commands supported in this mode are all those that don't provide graphic representation.

**Graphic** the visualization on display is either alphanumeric or graphic and, as described in "PHYSICAL FEATURES" paragraph, is characterized by zoom and elongation attributes. All the commands are supported in this mode.

The selection of the display mode is managed both inside the local set up program, as described in the homonymous paragraph, and through two dedicated serial commands.

## COMMUNICATION MODE

The **QTP G28** provides two different communication mode on the primary serial line:

- Normal** the communication uses 8 bits per character and user programmable stop bits, parity and baud rates, through local set up. This communication mode is suitable for point to point connections in RS 232, RS 422 and Current Loop.
- M. Slave** the communication uses 9 bits per character, no parity and user programmable stop bits and baud rates, through local set up. This communication mode is suitable either for point to point connections (with all the electric protocol) or network connections (with RS 422, RS 485 and Current Loop protocol). For further information on master slave mode, please refer to next paragraph.

The selection of the communication mode is managed only inside the local set up program, as described in the homonymous paragraph, while the electric protocol must be specified in the order phase.

## **MASTER SLAVE COMMUNICATION MODE**

The Master Slave mode uses the 9 bits communication technique.

In addition to the 8 data bit also a 9<sup>th</sup> 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 9<sup>th</sup> 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 G28** is used, the identifying code must be that one set by the local set up programm on the NAME entry.

When this byte is sent (with 9<sup>th</sup> bit set to 1) the **QTP G28** 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 G28** 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 G28** will consider the data string ended and it will begin the answering phase. The Time Out value for each baud rate is below described:

<b>Baud Rate</b>	<b>Time Out</b>
57600 Baud	286 µsec
19200 Baud	859 µsec
9600 Baud	1719 µsec
4800 Baud	3437 µsec
2400 Baud	6875 µsec
1200 Baud	13750 µsec

When the Time Out is over, the **QTP G28** begins the answering phase which 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.

For example if a string containing the command to read the version is transmitted, the pressed key code will be sent back as answer to the current received command, while the required version number (previously asked for) will be sent as answer of the next received command.

When the last character of the command string has been transmitted, the master must wait a time of:

$$\text{"character transmission time"} + \text{Time Out}$$

before reaching the first character of the answering string returned by the **QTP G28**.

For example if we are working at 57.6 KBaud, when the transmission of the last char has been completed, it is necessary to wait for about 477 (character transmission time at 57.6 KBaud=191 + Time Out=286)  $\mu$ sec, before the reception of the first answering byte of **QTP G28** will be completed.

#### 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 transmission, according to this scheme:

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

<i>If 9<sup>th</sup> bit must be 1</i>	->	<i>Set parity to ODD</i>
<i>If 9<sup>th</sup> bit must be 0</i>	->	<i>Set parity to EVEN</i>

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

<i>If 9<sup>th</sup> bit must be 1</i>	->	<i>Set parity to EVEN</i>
<i>If 9<sup>th</sup> bit must be 0</i>	->	<i>Set parity to ODD</i>

### KEYBOARD ACQUISITION

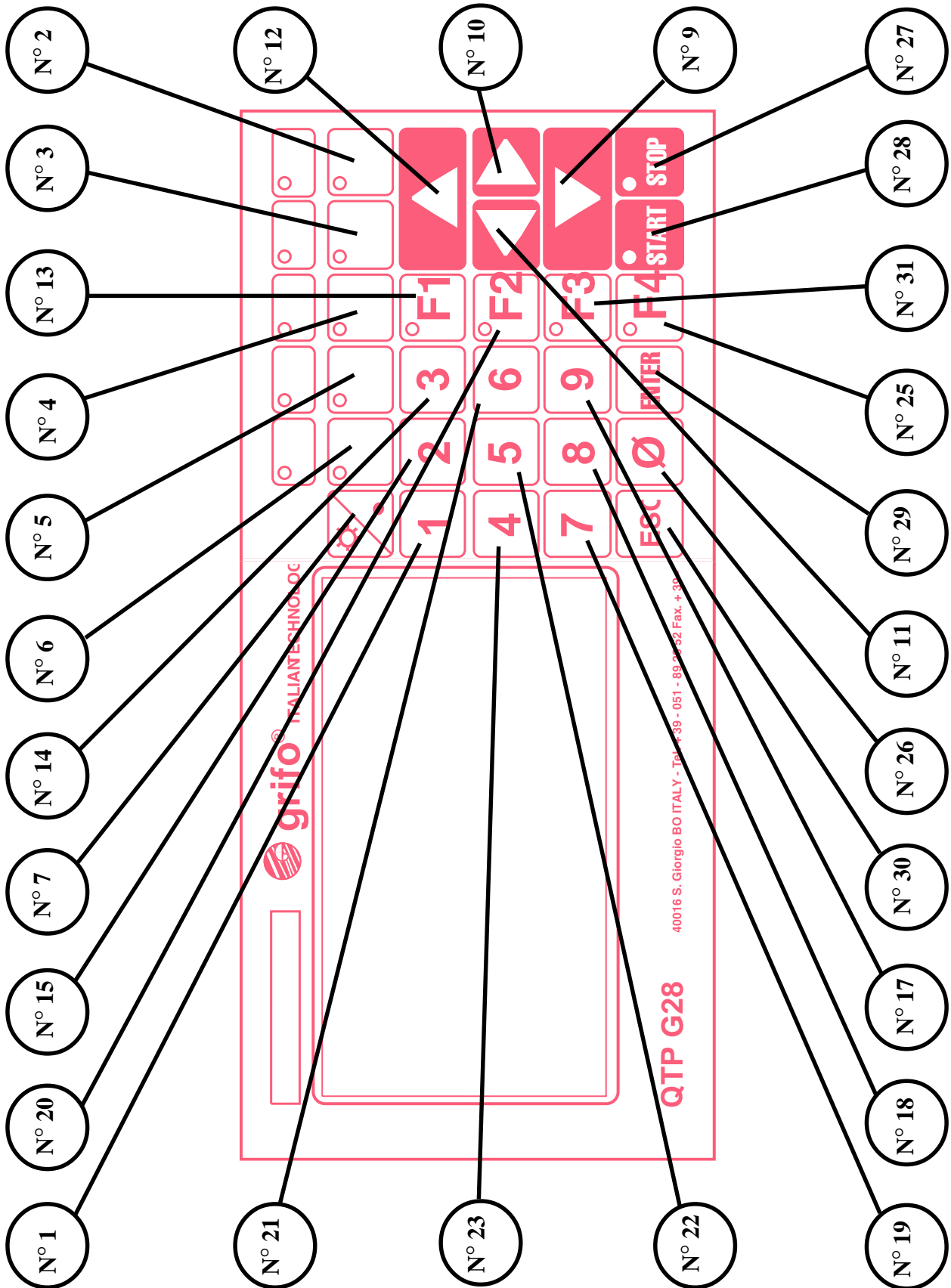
When **QTP G28** recognizes a key pressure, it transmits the pertinent code in serial mode and this is immediate if normal communication mode is used. When Master Slave communication is used, the said code is returned only upon specific request of the Master unit with the format illustrated in the previous paragraph.

The **autorepeat** function of the pressed key is also implemented so when **QTP G28** recognizes the pressure on a specific key for a time period over **500 msec**. it will repeat the serial transmission of its code every **100 msec**. and it lasts until the key is released.

Figure 23 shows the keys numbers of **QTP G28**, instead the table of figure 24 shows the codes serially sent by the terminal when a key is pressed. As per the command sequences these ones are expressed under decimal form, hexadecimal and mnemonic formats by means of the specific characters of the ASCII table.

**KEYS MAP**

Below is reported a figure that shows the numbers of all the QTP G28 keys.



**FIGURE 23: KEYS MAP**



## DEFAULT KEYS CODES

Here are the tables which show the codes that **QTP G28** send on primary serial line when a key is pressed; the listed codes are those under default conditions i.e when no keys have been yet reconfigured with proper command.

KEY N°	CODE	HEX CODE	MNEMONIC
1	49	31	1
2	70	46	F
3	69	45	E
4	68	44	D
5	67	43	C
6	66	42	B
7	46	2E	.
9	10	0A	LF
10	12	0C	FF
11	8	08	BS
12	11	0B	VT
13	58	3A	:
14	51	33	3
15	50	32	2
17	57	39	9
18	56	38	8
19	55	37	7
20	59	3B	;
21	54	36	6
22	53	35	5
23	52	34	4
25	63	3F	?
26	48	30	0
27	61	3D	=
28	60	3C	<
29	13	0D	CR
30	27	1B	ESC
31	62	3E	>

FIGURE 24: DEFAULT KEYS CODE

## KEYBOARD REGULATION OF DISPLAY BRIGHTNESS

One of the most important feature of **QTP G28** is the possibility to change the display brightness directly from keyboard, with no inflection on standard functionality of the terminal. In detail if you refer to figure 23, the brightness is **increased** by simultaneously pressing the keys N° 7 and N° 12 and the brightness is **decreased** by simultaneously pressing the keys N° 7 e N° 9. During the brightness regulation the keyclick is audible, if activated, but the codes of the pressed keys are not transmitted. To remind this possibility the used keys have a clear symbols, in fact the keys number 7, 9 and 12 respectively have the brightness, down arrow and up arrow symbols. The brightness indirectly acts also on display contrast so its variation always ensures the best visibility conditions, especially when the environmental lighting changes.

## CHARACTERS VISUALIZATION ON THE DISPLAY

When **QTP G28** is setted in alphanumeric mode, it shows on the display all the characters having a code included in the range **32÷255 (20÷FF Hex)**; if it is sent a code not included in this range and this latter is not a command, the code is ignored. The characters in the range: **32÷125 (20÷7D Hex)** correspond to those ones of the standard ASCII table, while characters associated to **126÷255 (7E÷FF Hex)** are those described in "APPENDIX A" table.

When **QTP G28** is setted in graphic mode, it shows on the display all the characters having a code included in the range **32÷126 (20÷7E Hex)**; if it is sent a code not included in this range and this latter is not a command, the code is ignored. The characters in the range: **32÷126 (20÷7E Hex)** correspond to those ones of the standard ASCII table.

In both the visualization mode the character is visualized at the current cursor position and this latter is moved on the the next position; if the cursor is placed in the last position of the display (low, right corner), it will be moved to Home position (up,left corner).

## RECEIVE BUFFER

**QTP G28** 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 (**3200** bytes long) and after processed at the end of the currently executed operation.

Naturally when command sequences that requires a long execution time (graphic commands, EEPROM management commands, etc.) are continuously received, the buffer will become full and it overflows. For this reason the **QTP G28** advise of the upcoming overflow by disabling the hardware handshake of the primary serial line (**RTSA RS232** setted at -12 Vdc).

The master unit must properly manage this signal, by arresting the transmission until the **QTP G28** has emptied the receive buffer and it is still ready to receive other data.

Please note that when **RS 422**, **RS 485** or **Current Loop** communication is selected this feature is not usable, in fact the /RTS signal is no more available. Thus 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.

## 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 POSITIONING

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, it 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.

## ALPHANUMERIC 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 codes are the row and column values of the position, plus a constant offset of 32 (20 Hex).

If, for example, the user wants to place the cursor at Home position (line 0, column 0), the following byte sequence must be sent:

**27 89 32 32.**

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

## RETURN ALPHANUMERIC CURSOR POSITION

*Code:* 27 90  
*Hex code:* 1B 5A  
*Mnemonic:* ESC Z

When QTP G28 receives this command, it returns sequentially the two values "r" and "c" that coincide to the row and column where the cursor is currently positioned. The returned values are absolute and without offset, thus if the cursor is in Home position the command will send back the values:

**0 0**

## COMMANDS FOR EEPROM

In the following paragraphs are described all the commands that manage the data saved on **QTP G28** on board EEPROM.

### REQUEST FOR EEPROM WRITING POSSIBILITY

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

This command checks if the **QTP G28** 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 G28** receives this command, it answers with the following codes:

6	(06 Hex)	(ACK)	->	<b>QTP G28</b> ready
21	(15 Hex)	(NACK)	->	<b>QTP G28</b> not ready

If the **QTP G28** sends back the NACK code, it is not yet possible to memorize a new data on EEPROM.

### WRITING OF LIFE BYTE

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

This command sets the card "Life 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 desired value, it allows for example, to verify that **QTP G28** 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 LIFE BYTE

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

The **QTP G28** sends back on the serial line the value of its "Life Byte".

For example, this command can be useful to verify the presence or the correct running of the terminal.

## WRITE BLOCK OF DATA ON EEPROM

**Code:**            27 6 *addL addH nb byte1 ... byte n*  
**Hex code:**       1B 6 *addL addH nb byte1 ... byte n*  
**Mnemonic:**      ESC ACK ASCII(*addL*) ASCII(*addH*) ASCII(*nb*) ASCII(*byte1*) ...  
                       ... ASCII(*byte n*)

After the two initial codes, other n bytes must be transmitted with the following meaning:

<i>addL</i>	(96÷255)	:	Low byte of start write address.
<i>addH</i>	(0÷1 in default configuration) (0÷3 with .EE08 option) (0÷7 with .EE16 option)	:	High byte of start write address.
<i>nb</i>	(0÷255)	:	Number of bytes to write.
<i>byte1</i>	(0÷255)	:	Byte 1 to save on EEPROM.
:			
:			
<i>byte n</i>	(0÷255)	:	Byte n to save on EEPROM.

The first 96 bytes of on board serial EEPROM (0÷95), are not available for the user in fact they are used for **QTP G28** set up parameters.

**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.

## READ BLOCK OF DATA FROM EEPROM

**Code:**            27 7 *addL addH nb*  
**Hex code:**       1B 7 *addL addH nb*  
**Mnemonic:**      ESC BEL ASCII(*addL*) ASCII(*addH*) ASCII(*nb*)

After the two initial codes, other 3 bytes must be transmitted with the following meaning:

<i>addL</i>	(96÷255)	:	Low byte of start write address.
<i>addH</i>	(0÷1 in default configuration) (0÷3 with .EE08 option) (0÷7 with .EE16 option)	:	High byte of start write address.
<i>nb</i>	(0÷255)	:	Number of bytes to write.

and **QTP G28** returns the "nb" bytes acquired from the specified address. The first 96 bytes of on board serial EEPROM (0÷95), are not available for the user in fact they are used for **QTP G28** set up parameters.

## COMMANDS FOR LEDS MANAGEMENT

Below are described all the **QTP G28** commands that can be used to manage the visualization LEDs.

### LED ACTIVATION

**Code:**            27 50 n.LED Attr.  
**Hex code:**       1B 32 n.LED Attr.  
**Mnemonic:**      ESC 2 ASCII(n.LED) ASCII(Attr.)

The LED shown in "n.LED" is setted with the attribute specified in "Attr.". The LEDs numbers are included in **0÷15** range as shwon in figure 25.

The available attributes are as follows:

0	(00 Hex)	->	Not enabled LED
255	(FF Hex)	->	Enabled LED
85	(55Hex)	->	Blinking LED

For example if you wish to enable LED n.5 with blinking attribute, the following sequence has to be sent:

**27 50 5 85**

If the parameters LED number or attribute are not valid, the command is ignored.

### LEDS ACTIVATION WITH MASK

**Code:**            27 52 mask1 mask2 mask3  
**Hex code:**       1B 34 mask1 mask2 mask3  
**Menomonic:**     ESC 4 ASCII(mask1) ASCII(mask2) ASCII(mask3)

All the LEDs available on **QTP G28** are contemporarily managed as indicated in "mask1", "mask2" and "mask3" with the following correspondence:

mask1 (bit 0 ...7)	->	LED 0 ... LED 7
mask2 (bit 0 ...7)	->	LED 8 ... LED 15
mask3	->	(No function, manteined for compatibility)

If a bit is placed at 0 logic state, the correpondent LED is turned off (disabled), viceversa it will be turned on (enabled) if the correspondent bit is at 1 state.

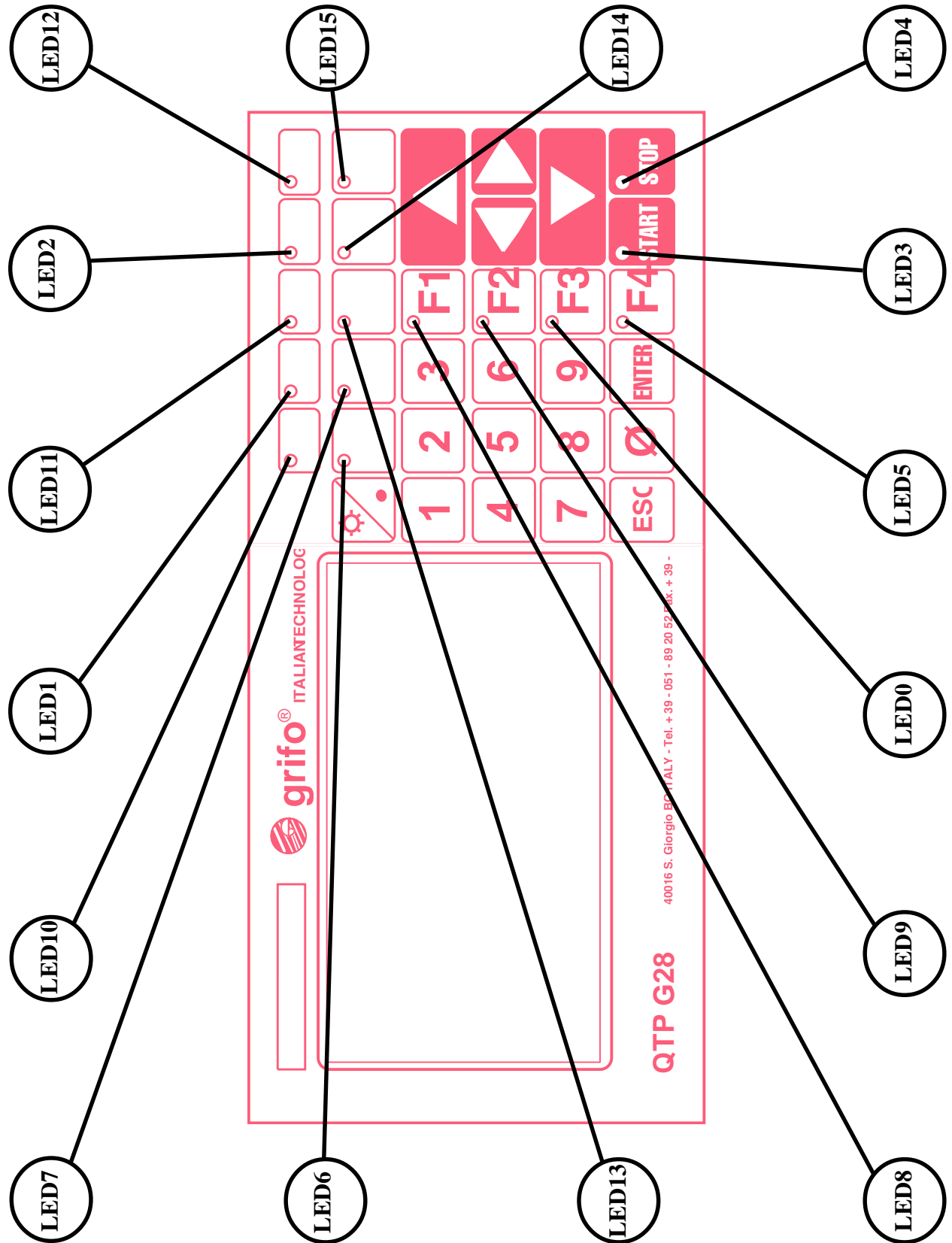
If there are some LEDs having the blinking attribute, this latter will be disabled.

**NOTE:** The "mask3" must be always sent even if it has no meaning, for a correct management of all the terminal's LEDs.



**LEDS MAP**

The following figure shows the number of **QTP G28** LEDs and their location on the front panel.



**FIGURE 25: LEDS MAP**

## COMMANDS FOR KEYBOARD MANAGEMENT

Below are described the commands that can be used to manage the **QTP G28** keyboard.

### KEY RECONFIGURATION

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

When the selected key is reconfigured, each time it is pressed, the card will serially send the new specified code. The number of the key to be reconfigured (*n.key*) is described in figure 23 and it must be included in a range of **0÷31 (0÷1F Hex)**; if this is not done the command is ignored. The code value can vary in **0÷254 (0÷FE Hex)** range, as the **255 value (FF Hex)** indicates that the key is disabled so when it is pressed the **QTP** will not send any codes on the primary serial line.

**NOTE:** The said command needs a data writing on the on board EEPROM so before executing it, it is better to check if the card is ready for a new writing on such device, otherwise the command will be ignored.

### KEYCLICK ON WITH MEMORIZATION

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

The **keyclick** function is switched on so there is a sound feedback when a key is pressed. This setting is saved inside the on board EEPROM, thus it is maintained even if the power voltage fails.

**NOTE:** The said command involves a data writing on the on board EEPROM so before executing it, it is better to check if the card is ready for a new writing on such device, otherwise the command will be ignored.

### KEYCLICK OFF WITH MEMORIZATION

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

The **keyclick** function is disabled so there is not sound feedback when a key is pressed. This setting is saved inside the on board EEPROM, thus it is maintained even if the power voltage fails.

**NOTE:** The said command involves a data writing on the on board EEPROM so before executing it, it is better to check if the card is ready for a new writing on such device, otherwise the command will be 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.

## 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.

## COMMANDS FOR VARIOUS FUNCTIONS

Below are described the commands that can be used to manage different **QTP G28** features.

### READING OF VERSION NUMBER

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

On the primary serial line is returned a string of 3 chars containing the program managing version that is resident and executed by **QTP G28**.

### RELAY ACTIVATION

*Code:* 27 56  
*Hex code:* 1B 38  
*Mnemonic:* ESC 8

The relay on the card is enabled; in detail the normally opened contact is closed (**Common** contact is connected to **N.O.** contact) and the normally closed contact is opened.

This command has result only if the **.RELE'** option is mounted on **QTP G28**.

### RELAY DEACTIVATION

*Code:* 27 57  
*Hex code:* 1B 39  
*Mnemonic:* ESC 9

The relay on the card is disabled; in detail the normally opened contact is opened and the normally closed contact is closed (**Common** contact is connected to **N.C.** contact).

This command has result only if the **.RELE'** option is mounted on **QTP G28**.

### BEEP

*Code:* 7  
*Hex code:* 7  
*Mnemonic:* BEL

The buzzer is enabled for a time of 0.1 second.

### SETTING OF ALPHANUMERIC VISUALIZATION MODE

*Code:* 27 208  
*Hex code:* 1B D0  
*Mnemonic:* ESC ASCII(208)

After this command is received, the **QTP G28** display is cleaned, the cursor is placed in HOME position and the display's alphanumeric visualization mode is selected.

Please remind that this visualization mode doesn't support the graphic commands.

## SETTING OF GRAPHIC VISUALIZATION MODE

**Code:** 27 209  
**Hex code:** 1B D1  
**Mnemonic:** ESC ASCII(209)

After this command is received, the **QTP G28** display is cleaned, the cursor is placed in HOME position, a ZOOM=0 is setted, elongation is disabled and the display's graphic visualization mode is selected.

Please remind that this visualization mode supports all the graphic commands.

## SCREEN VISUALIZATION

**Code:** 27 122 *screenH screenL*  
**Hex code:** 1B 7A *screenH screenL*  
**Mnemonic:** ESC z ASCII(*screenH*) ASCII(*screenL*)

After the two initial codes, other 2 bytes must be transmitted with the following meaning:

*screenL* (1÷130) : Low byte of the screen to visualize.  
*screenH* (0) : High byte of the screen to visualize.

The "screenH" byte is provided for future expansion of the terminal but the user must always send it with a 0 value. If graphic mode is enabled, before using this command please check that ZOOM is 0 and no elongation is selected: if this conditions are not met the surplus characters are over displayed. This command can be used only when the **QTP G28** on board EPROM contains the file "SCREEN.DAT". This file is created by **RKD\_EDIT.EXE** program, supplied on the distribution disk, that allows the production of alphanumeric screens (pages of text that contains enough characters to cover exactly the display). These screens can be recalled and displayed through the command reducing the size of the user program and the amount of data to transmit. The file SCREEN.DAT must be burned in the EPROM starting from 10000H address, using an external EPROM programmer.

For further informations about how to use the **RKD\_EDIT.EXE** program, please refer to its own on line help.

## STRING VISUALIZATION

**Code:** 27 8 *chr0 ... chr n*  
**Hex code:** 1B 8 *chr0 ... chr n*  
**Mnemonic:** ESC BS ASCII(*chr0*) ... ASCII(*chr n*)

After the two initial codes, other n bytes must be transmitted with the following meaning:

*chr0* (0÷255) : First character of the string.  
 :  
 :  
*chr n* (13) : Last character of the string: it must be a CR.

**NOTE** The string is displayed on the display only when the last byte (CR=0DH) is received.

## COMMANDS FOR AUXILIARY SERIAL LINE MANAGEMENT

The following paragraph describe the commands for the **QTP G28** auxiliary serial line management.

### **ENABLE COPY OF CHR ON AUXILIARY SERIAL LINE**

*Code:*            27 3  
*Hex code:*       1B 3  
*Mnemonic:*       ESC ETX

After this command is received, all the characters received on primary serial line are normally processed and executed and before they are transmitted on the auxiliary serial line.

### **ENABLE TRANSPARENT COPY OF CHR ON AUXILIARY SERIAL LINE**

*Code:*            27 2  
*Hex code:*       1B 2  
*Mnemonic:*       ESC STX

After this command is received, all the characters received on primary serial line are transmitted on the auxiliary serial line in a transparent mode: the terminal doesn't process any received character.

### **DISABLE COPY OF CHR ON AUXILIARY SERIAL LINE**

*Code:*            27 4  
*Hex code:*       1B 4  
*Mnemonic:*       ESC EOT

After this command is received, the auxiliary serial line is no more used.

### **READ STATUS OF AUXILIARY SERIAL LINE HANDSHAKE**

*Code:*            27 5  
*Hex code:*       1B 5  
*Mnemonic:*       ESC ENQ

When **QTP G28** executes this command it returns on the primary serial line one byte that contain the handshake status:

Byte = 0   --> Handshake **CTSB RS232** enabled.  
Byte = 1   --> Handshake **CTSB RS232** disabled.

The most frequent use of the auxiliary serial line is the management of a serial printer and through this command it is possible to acquire the printer status and, for example, recognize if it can receive data to print. The hardware handshake **CTSB RS232** on CN3B should be connected to proper output handshake signal of the printer, that informs about its status.

## 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.

### **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 char.

### **BLINKING CURSOR ON**

*Code:*            27 77  
*Hex code:*       1B 4D  
*Mnemonic:*      ESC M

The cursor is activated so it is visible. Now it is a blinking line placed under the char.

## COMMANDS FOR REAL TIME CLOCK MANAGEMENT

The following paragraph describe the commands for the on board Real Time Clock. Please remind that these commands are correctly executed only if **.RTC** option is mounted on **QTP G28**.

### CLOCK SET UP

**Code:**            27 33 70 *hou min sec day mon yea wee*  
**Hex code:**        1B 21 46 *hou min sec day mon yea wee*  
**Mnemonic:**        ESC ! F ASCII(*hou*) ASCII(*min*) ASCII(*sec*) ASCII(*day*) ASCII(*mon*)  
                       ASCII(*yea*) ASCII(*wee*)

The on board Real Time Clock is set with the data contained in the parameters that follow the command. The meanings and the allowed ranges of these parameters is below described:

<i>hou</i>	(0÷23)	->	<i>Hour</i>
<i>min</i>	(0÷59)	->	<i>Minute</i>
<i>sec</i>	(0÷59)	->	<i>Second</i>
<i>day</i>	(1÷31)	->	<i>Day of the mont</i>
<i>mon</i>	(1÷12)	->	<i>Month</i>
<i>yea</i>	(0÷99)	->	<i>Year</i>
<i>wee</i>	(0÷6)	->	<i>Day of the week: 0 = Sunday</i>
			:           :
			6 = Saturday

If one of these ones has a value not included in the allowed range, the relative RTC location is not programmed.

### CLOCK READING

**Code:**            27 33 102  
**Hex code:**        1B 21 66  
**Mnemonic:**        ESC ! f

Seven bytes, named **hou**, **min**, **sec**, **day**, **mon**, **yea**, **wee**, are sent back via primary serial line with the date, time and day of the week which are acquired from the on board Real Time Clock. The meaning of these returned parameters is the same listed in "CLOCK SET UP" paragraph.

### TIME VISUALIZATION

**Code:**            27 33 116 *r c frm*  
**Hex code:**        1B 21 74 *r c frm*  
**Mnemonic:**        ESC ! t ASCII(*r*) ASCII(*c*) ASCII(*frm*)

The current time acquired from the on board Real Time Clock is displayed starting from the position indicated by the "r" and "c" bytes.

If alphanumeric visualization is selected these parameters express the row and column values of the display at which an offset of **32 (20 Hex)** must be added, so theyr allowed range are **32÷47** e **32÷61**. If graphic visualization is selected these parameters on the contrary express the absolute row and column values in pixels, so theyr allowed range are **0÷127** e **0÷239**.



Always in graphic mode the time is displayed with the zoom, elongation, and write direction currently selected when the command is received. If the row and column values are not included in the specified ranges, this command will be ignored.

The "**frm**" parameter is used to specify the visualization format, in detail:

- Bit 0** -> 1 The time is visualized and automatically managed in "**r**", "**c**" position.
- 0 The visualization of the time is interrupted.
- Bit 1** -> 1 The time is visualized in AM/PM format: **HH:MM:SSm** with: **HH**=hours, **MM**=minutes, **SS**=seconds, **m=a** (AM) or **p** (PM).
- 0 The time is visualized in 24H format: **HH:MM:SS** with: **HH**=hours, **MM**=minutes, **SS**=seconds.
- Bit 2÷7** -> 0 Reserved for future expansion. They must be set to **0** value.

For example, if you wish to visualize the time starting from Home position, in 24H format, with graphic mode enabled, it will be necessary to send the following sequence:

**27 33 116 0 0 1**

## DATE VISUALIZATION

**Code:** *27 33 100 r c frm*

**Hex code:** *1B 21 64 r c frm*

**Mnemonic:** *ESC ! d ASCII(r) ASCII(c) ASCII(frm)*

The current date acquired from the on board Real Time Clock is displayed starting from the position indicated by the "**r**" and "**c**" bytes.

If alphanumeric visualization is selected these parameters express the row and column values of the display at which an offset of **32 (20 Hex)** must be added, so theyr allowed range are **32÷47** e **32÷61**.

If graphic visualization is selected these parameters on the contrary express the absolute row and column values in pixels, so theyr allowed range are **0÷127** e **0÷239**.

Always in graphic mode the date is displayed with the zoom, elongation, and write direction currently selected when the command is received. If the row and column values are not included in the specified ranges, this command will be ignored.

The "**frm**" parameter is used to specify the visualization format, in detail:

- Bit 0** -> 1 The date is visualized and automatically managed in "**r**", "**c**" position.
- 0 The visualization of the date is interrupted.
- Bit 1** -> 1 The format that visualizes the date is as follows: **MM-DD-YY**, with **MM**=month, **DD**=day, **YY**=year.
- 0 The format that visualizes the date is as follows: **DD-MM-YY**, with **DD**=day, **MM**=month, **YY**=year.
- Bit 2÷7** -> 0 Reserved for future expansion. They must be set to "0" value.

Please note that the weerk day is not managed.

For example, if you wish to visualize the date starting from Home position, in MM-DD-YY format, with graphic mode selected, it will be necessary to send the following sequence:

**27 33 100 0 0 3**

## GRAPHIC COMMANDS

Here follows the description of the several graphic mode commands, available on **QTP G28**. Please remind that all these commands can be used only when the display mode is setted as graphic through the local set up program or the proper command, previously described.

### **REVERSE ATTRIBUTE SETTING**

*Code:*            **27 48 80**  
*Hex code:*       **1B 30 50**  
*Mnemonic:*       **ESC 0 P**

This command is useful when the user wishes to delete a graphic primitive or parts of the display, in fact it is sufficient to activate the REVERSE attribute and redraw the picture to be deleted, or more easily to emphas some text.

After this command has been sent the reverse attribute can be enabled or disabled with the following two commands.

### **ENABLE SELECTED ATTRIBUTE**

*Code:*            **14**  
*Hex code:*       **E**  
*Mnemonic:*       **SO**

Please remind that before to send this command, at least one selection attribute command must be transmitted, otherwise it is ignored.

### **DISABLE SELECTED ATTRIBUTE**

*Code:*            **15**  
*Hex code:*       **F**  
*Mnemonic:*       **SI**

Please remind that before to send this command, at least one selection attribute command must be transmitted, otherwise it is ignored.

### **CHARACTERS WRITE DIRECTION**

*Code:*            **27 210 dir**  
*Hex code:*       **1B D2 dir**  
*Mnemonic:*       **ESC ASCII(210) ASCII(dir)**

This command sets the charactes write direction; in detail according to the value of the parameter the direction will be:

<b>dir:</b>	<b>0</b>	->	<i>Horizontal</i> characters writing ( <b>default</b> )
	<b>1</b>	->	<i>Vertical</i> characters writing

Vertical writing moves from top to bottom and writes the characters rotated of **90 degrees clockwise**.

## CHARACTERS ZOOM SETTING

**Code:** 27 201 zoom  
**Hex code:** 1B C9 zoom  
**Mnemonic:** ESC ASCII(201) ASCII(zoom)

This command sets the zoom level for the characters; in detail the new character size will be as reported in the following list:

<b>zoom:</b>	<b>0</b>	->	<b>8x8</b> pixels characters (default)
	<b>1</b>	->	<b>16x16</b> pixels characters
	<b>2</b>	->	<b>24x24</b> pixels characters
	<b>3</b>	->	<b>32x32</b> pixels characters
	<b>4</b>	->	<b>40x40</b> pixels characters

**NOTE** After the zoom level setting, the cursor is placed in the Home position.

## CHARACTERS ELONGATION SETTING

**Code:** 27 207 elong  
**Hex code:** 1B CF elong  
**Mnemonic:** ESC ASCII(207) ASCII(elong)

This command sets the elongation level for characters visualization; in detail the new elongation level will set the following characters dimensions:

<b>elong:</b>	<b>0</b>	->	<i>Normal:</i> <b>8x8</b> pixels characters ( <b>default</b> )
	<b>1</b>	->	<i>Vertical:</i> <b>8x16</b> pixels characters
	<b>2</b>	->	<i>Orizzontal:</i> <b>16x8</b> pixels characters

**NOTE** This command can be used only if the current zoom value is setted to **0**, viceversa some visualization problems will appear on the display.

## GRAPHIC CURSOR PLACEMENT

**Code:** 27 206 y x 0  
**Hex code:** 1B CE y x 0  
**Mnemonic:** ESC ASCII(206) ASCII(y) ASCII(x) NUL

This command moves the cursor to the point pointed by **x** and **y** coordinate; this position is absolute, so it does not depend on the values of the other parameters and it is different from the alphanumeric cursor placement

The characters received after this command will be displayed from the indicated point, going to the right and to the bottom.

The values of **y** and **x** must be respectively in the range **0÷127** and **0÷239**, or on the other hand the display dimension.

**NOTE** The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## RETURN GRAPHIC CURSOR POSITION

*Code:* 27 211  
*Hex code:* 1B D3  
*Mnemonic:* ESC ASCII(211)

When **QTP G28** receives this command, it returns sequentially three values: **x**, **y** and **0** that coincide to the coordinates in pixels where the cursor is currently positioned.

The values of **y** and **x** are included respectively in the ranges **0÷127** and **0÷239**, or on the other hand the display dimension, they are absolute and without offset; thus if the cursor is in Home position the command will send back the values:

**0 0 0**

**NOTE** The code **0** (NUL) returned at the end of the answer sequence is present for compatibility to future versions and must be always received.

## ONE PIXEL VISUALIZATION

*Code:* 27 229 y x 0  
*Hex code:* 1B E5 y x 0  
*Mnemonic:* ESC ASCII(229) ASCII(y) ASCII(x) NUL

This command visualizes the pixel at the coordinates **x** and **y**.

The values of **y** and **x** must be respectively in the range **0÷127** and **0÷239**, or on the other hand the display dimension.

**NOTE** The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## LINE DRAWING

*Code:* 27 203 y1 x1 0 y2 x2 0  
*Hex code:* 1B CB y1 x1 0 y2 x2 0  
*Mnemonic:* ESC ASCII(203) ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2)  
NUL

This command draws a line starting from the point at coordinates **x1**, **y1** and ending at the point at coordinates **x2**, **y2**. The values of **y1**, **y2** and **x1**, **x2** must be respectively in the range **0÷127** and **0÷239**, or on the other hand the display dimension.

**NOTE** The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## RECTANGLE DRAWING

*Code:* 27 202 y1 x1 00 y2 x2 0  
*Hex code:* 1B CA y1 x1 00 y2 x2 0  
*Mnemonic:* ESC ASCII(202) ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2)  
NUL

This command draws a rectangle whose top left corner is located at **x1, y1** and whose bottom right corner is located at **x2, y2**. The values of **y1, y2** and **x1, x2** must be respectively in the range **0÷127** and **0÷239**, or on the other hand the display dimension.

**NOTE** The code **0 (NUL)** put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## FILLED RECTANGLE DRAWING

*Code:*            **27 204 y1 x1 0 y2 x2 0**  
*Hex code:*       **1B CC y1 x1 0 y2 x2 0**  
*Mnemonic:*      **ESC ASCII(204) ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2)**  
                      **NUL**

This command draws a filled rectangle whose top left corner is located at **x1, y1** and whose bottom right corner is located at **x2, y2**. The values of **y1, y2** and **x1, x2** must be respectively in the range **0÷127** and **0÷239**, or on the other hand the display dimension. Please remark that this command, opportunely matched with the reverse attribute, allows to delete rectangular parts of the display, leaving unchanged the rest.

**NOTE** The code **0 (NUL)** put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## GRADUATED AXES AND GRID DRAWING

*Code:*            **27 228 y x 0 ly lx 0 gr dt**  
*Hex code:*       **1B E4 y x 0 ly lx 0 gr dt**  
*Mnemonic:*      **ESC ASCII(228) ASCII(y) ASCII(x) NUL ASCII(ly) ASCII(lx) NUL**  
                      **ASCII(gr) ASCII(dt)**

This command allows to quickly create a cartesian axes with the following features:

- x** : Vertical coordinate of the cartesian axes origin; must be in the range **3÷235**.
- y** : Horizontal coordinate of the cartesian axes origin; must be in the range **4÷124**.
- lx** : Length of X axis; it must be in the range **1÷max**.
- ly** : Length of Y axis; it must be in the range **1÷max**.
- gr** : **0** = Grid hidden; **1** = Grid visible.
- dt** : Distance between two graduations; must be in the rangee **3÷255 (3÷FF Hex)**.

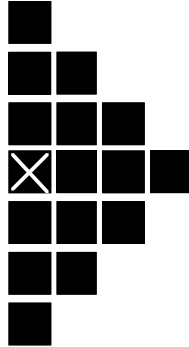
The value of **max** parameter depends on the size of the picture to draw; as it must be always possible to show it completely in the display area.

**NOTE** The code **0 (NUL)** put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## RIGHT ORIENTED FILLED ARROW DRAWING

**Code:**            27 230 y x 0  
**Hex code:**       1B E6 y x 0  
**Mnemonic:**      ESC ASCII(230) ASCII(y) ASCII(x) NUL

This command draws a filled arrow oriented to the right, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.



**FIGURE 26: RIGHT ORIENTED FILLED ARROW**

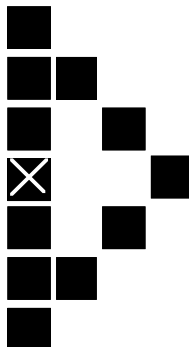
The values of the parameters **y** and **x** must be respectively in the range **3÷124** and **0÷236**, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE**      The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## RIGHT ORIENTED ARROW DRAWING

**Code:**            27 231 y x 0  
**Hex code:**       1B E7 y x 0  
**Mnemonic:**      ESC ASCII(231) ASCII(y) ASCII(x) NUL

This command draws an empty arrow oriented to the right, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.



**FIGURE 27: RIGHT ORIENTED ARROW**

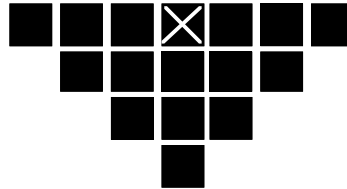
The values of the parameters **y** and **x** must be respectively in the range **3÷124** and **0÷236**, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE**      The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## BOTTOM ORIENTED FILLED ARROW DRAWING

**Code:** 27 232 y x 0  
**Hex code:** 1B E8 y x 0  
**Mnemonic:** ESC ASCII(232) ASCII(y) ASCII(x) NUL

This command draws a filled arrow oriented to the bottom, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.



**FIGURE 28: BOTTOM ORIENTED FILLED ARROW**

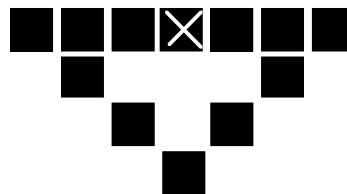
The values of the parameters **y** and **x** must be respectively in the range **0÷124** and **3÷236**, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE** The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## BOTTOM ORIENTED ARROW DRAWING

**Code:** 27 233 y x 0  
**Hex code:** 1B E9 y x 0  
**Mnemonic:** ESC ASCII(233) ASCII(y) ASCII(x) NUL

This command draws an empty arrow oriented to the bottom, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.



**FIGURE 29: BOTTOM ORIENTED ARROW**

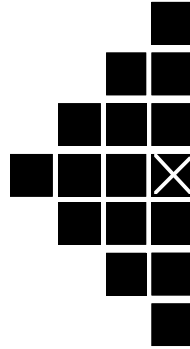
The values of the parameters **y** and **x** must be respectively in the range **0÷124** and **3÷236**, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE** The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## LEFT ORIENTED FILLED ARROW DRAWING

**Code:**            27 234 y x 0  
**Hex code:**       1B EA y x 0  
**Mnemonic:**      ESC ASCII(234) ASCII(y) ASCII(x) NUL

This command draws a filled arrow oriented to the left, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.



**FIGURE 30: LEFT ORIENTED FILLED ARROW**

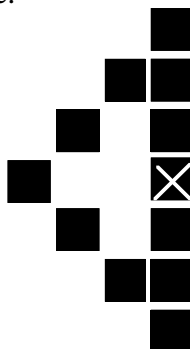
The values of the parameters **y** and **x** must be respectively in the range **3÷124** and **3÷239**, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE**     The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## LEFT ORIENTED ARROW DRAWING

**Code:**            27 235 y x 0  
**Hex code:**       1B EB y x 0  
**Mnemonic:**      ESC ASCII(235) ASCII(y) ASCII(x) NUL

This command draws an empty arrow oriented to the left, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.



**FIGURE 31: LEFT ORIENTED ARROW**

The values of the parameters **y** and **x** must be respectively in the range **3÷124** and **0÷239**, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE**     The code **0** (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.



## TOP ORIENTED FILLED ARROW DRAWING

**Code:** 27 236 y x 0  
**Hex code:** 1B EC y x 0  
**Mnemonic:** ESC ASCII(236) ASCII(y) ASCII(x) NUL

This command draws a filled arrow oriented to the top, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.

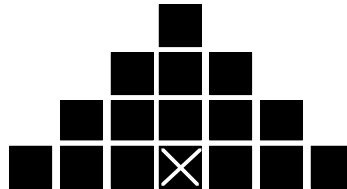


FIGURE 32: TOP ORIENTED FILLED ARROW

The values of the parameters **y** and **x** must be respectively in the range 3÷127 and 3÷236, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE** The code 0 (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## TOP ORIENTED ARROW DRAWING

**Code:** 27 237 y x 0  
**Hex code:** 1B ED y x 0  
**Mnemonic:** ESC ASCII(237) ASCII(y) ASCII(x) NUL

This command draws an empty arrow oriented to the top, with a centre "X" pointed by coordinates **x** and **y**, as shown in the following figure.

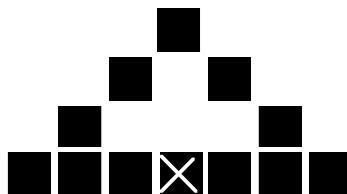


FIGURE 33: TOP ORIENTED ARROW

The values of the parameters **y** and **x** must be respectively in the range 3÷127 and 3÷236, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE** The code 0 (NUL) put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

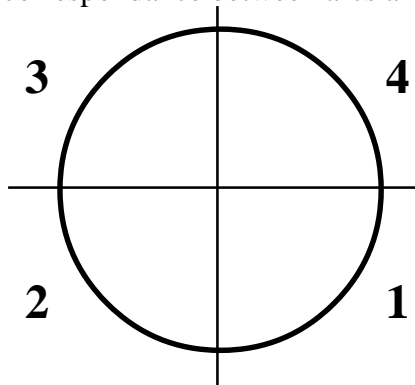
## CIRCUMFERENCE AND ARCS DRAWING

**Code:** `27 205 y x 0 r arc.in. n.arc.`  
**Hex code:** `1B CD y x 0 r arc.in. n.arc.`  
**Mnemonic:** `ESC ASCII(205) ASCII(y) ASCII(x) NUL ASCII(r) ASCII(arc.in.)`  
`ASCII(n.arc.)`

This command draws a circumference, or an arc of circumference, having radius equal to **r** and center at coordinates **x**, **y**. The values of these parameters must range from **0** to a maximum that depends on the size of the picture to draw; as it must be always possible to show it completely in the display area.

The remaining codes in the sequence allow the user to draw half circumference or quarters of circumference, they indicate the arc number from which to start drawing (**arc.in.**) and the number of arcs to draw (**n.arc.**); the values of these parameters must be in the range **1÷4**.

The following figure shows the correspondance between arcs and their numbers.



**FIGURE 34: ARCS NUMERATION AND LOCATION**

If, for example, the user wants to display a "C" shaped half circumference having radius of 20 pixels and center located at (120, 60), the following sequence must be transmitted:

**27 205 60 120 0 20 2 3**

**NOTE** The code **0 (NUL)** put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## FILLED CIRCUMFERENCE WITH 3 PIXELS RADIUS DRAWING

**Code:** `27 238 y x 0`  
**Hex code:** `1B EE y x 0`  
**Mnemonic:** `ESC ASCII(238) ASCII(y) ASCII(x) NUL`

This command draws a filled circumference, having radius equal to **3** and center at coordinates **x**, **y**. The values of the parameters **y** and **x** must be respectively in the range **2÷125** and **2÷237**, or on the other hand the positions that allow the complete figure drawing, on the display area.

Please remark that this command, opportunely matched with the following one, allows to create easily on/off visual indications, useful, for example, in a synoptic display.

**NOTE** The code **0 (NUL)** put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## CIRUMFERENCE WITH 3 PIXELS RADIUS DRAWING

*Codice:*            27 239 y x 0

*Codice Hex:*     1B EF y x 0

*Mnemonic:*     ESC ASCII(239) ASCII(y) ASCII(x) NUL

This command draws an empty circumference, having radius equal to **3** and center at coordinates **x**, **y**. The values of the parameters **y** and **x** must be respectively in the range **2÷125** and **2÷237**, or on the other hand the positions that allow the complete figure drawing, on the display area.

**NOTE**     The code **0 (NUL)** put at the end of the command sequence is present for compatibility to future versions and must be always sent for a correct execution of this command.

## CAN INTERFACE COMMANDS

Below are described the commands that manage the CAN serial interface available on **QTP G28** terminal. Most of these commands directly interact with **SJA 1000** UART, produced by **PHILIPS**, that drives the CAN interface at high level; thus if it is required a description of internal registers and functional modes of this UART, please refer to proper documentation of the manufacturing company. Please remind that the following commands are executed only when the **.CAN** option is mounted on **QTP G28** and the Low level mode is selected with the local set up program.

### CAN UART WRITING

*Code:*            27 128 add dat  
*Hex code:*       1B 80 add dat  
*Mnemonic:*      ESC ASCII(128) ASCII(add) ASCII(dat)

This command writes the value "dat" (0÷255) in the SJA 1000 UART register allocated at the address "add" (0÷31). The main purposes of this command are the CAN interface initialization, configuration, transmission, error reset, etc.

### CAN UART READING

*Code:*            27 129 add  
*Hex code:*       1B 81 add  
*Mnemonic:*      ESC ASCII(129) ASCII(add)

This command reads the value "dat" (0÷255) from the SJA 1000 UART register allocated at the address "add" (0÷31) and send it back on the primary serial line. The main purposes of the command are the CAN interface status check, error control, reception, etc.

### CAN UART MESSAGE WRITING

*Code:*            27 130 byte1 ... byte10  
*Hex code:*       1B 82 byte1 ... byte10  
*Mnemonic:*      ESC ASCII(130) ASCII(byte1) ... ASCII(byte10)

This command waits up to 15 msec. the possibility of the UART to send a message and if it is possible writes the message "byte1÷byte10" (0÷255) into the 10 registers of the UART buffer transmission. Moreover the status register **SR** value (0÷255) is returned on the primary serial line. This command is used to transmit data to other units of the CAN network.

### CAN UART MESSAGE READING

*Code:*            27 131  
*Hex code:*       1B 83  
*Mnemonic:*      ESC ASCII(131)

This command returns the status register **SR** value (0÷255), checks the reception of a message from the SJA 1000 UART and if available it sends back the message "byte1÷byte10" (0÷255) acquired from the 10 registers of the UART buffer reception. This command is used to receive data from other units of the CAN network.

## APPENDIX A: COMMANDS CODES SUMMARY TABLES

The tables of this appendix list a summary of all the command sequences recognized by QTP G28:

<i>COMMAND</i>	<i>CODE</i>	<i>HEX CODE</i>	<i>MNEMONIC</i>
<b>Characters visualization</b>	32÷255	20÷FF	SP÷ASCII(255)
<b>BACKSPACE</b>	08	08	BS
<b>CLEAR PAGE</b>	12	0C	FF
<b>CLEAR LINE</b>	25	19	EM
<b>CLEAR END OF LINE</b>	27 75	1B 4B	ESC K
<b>CLEAR END OF PAGE</b>	27 107	1B 6B	ESC k
<b>CURSOR LEFT</b>	21	15	NACK
<b>CURSOR RIGHT</b>	06	06	ACK
<b>CURSOR DOWN</b>	10	0A	LF
<b>CURSOR UP</b>	26	1A	SUB
<b>HOME</b>	01	01	SOH
<b>CARRIAGE RETURN</b>	13	0D	CR
<b>Carriage Return + Line Feed</b>	29	1D	GS
<b>Alphanumeric cursor placement</b>	27 89 r c	1B 59 r c	ESC Y ASCII(r) ASCII(c)
<b>Return alphanum. cursor position</b>	27 90	1B 5A	ESC Z
<b>Request forEEPROM writing possibility</b>	27 51	1B 33	ESC 3
<b>Writing of life byte</b>	27 33 78 byte	1B 21 4E byte	ESC ! N ASCII(byte)
<b>Reading of life byte</b>	27 33 110	1B 21 6E	ESC ! n
<b>Write block of data on EEPROM</b>	27 6 addL addH nb byte1 ...byten	1B 6 addL addH nb byte1 ... byten	ESC ACK ASCII(addL) ASCII(addH) ASCII(nb) ASCII(byte1) ... ASCII(byten)
<b>Read block of data from EEPROM</b>	27 7 addL addH nb	1B 7 addL addH nb	ESC BEL ASCII(addL) ASCII(addH) ASCII(nb)
<b>LED activation</b>	27 50 n.LED Attr.	1B 32 n.LED Attr.	ESC 2 ASCII(n.LED) ASCII(Attr.)
<b>LEDs activation with mask</b>	27 52 mask1 mask2 mask3	1B 34 mask1 mask2 mask3	ESC 4 ASCII(mask1) ASCII(mask2) ASCII(mask3)
<b>Key reconfiguration</b>	27 55 n.key code	1B 37 n.key code	ESC 7 ASCII(n.key) ASCII(code)
<b>Keyclick on with memorization</b>	27 33 53	1B 21 35	ESC ! 5
<b>Keyclick off with memorization</b>	27 33 54	1B 21 36	ESC ! 6
<b>Keyclick on without memorization</b>	27 53	1B 35	ESC 5
<b>Keyclick off without memorization</b>	27 54	1B 36	ESC 6

FIGURE A1: COMMANDS CODES TABLE (1 OF 4)

<i>COMMAND</i>	<i>CODE</i>	<i>HEX CODE</i>	<i>MNEMONIC</i>
Reading of version number	27 86	1B 56	ESC V
RELAY activation	27 56	1B 38	ESC 8
RELAY DEactivation	27 57	1B 39	ESC 9
Beep	07	07	BEL
Setting of alphanumeric visualization mode	27 208	1B D0	ESC ASCII(208)
Setting of graphic visualization mode	27 209	1B D1	ESC ASCII(209)
Screen visualization	27 122 screenH screenL	1B 7A screenH screenL	ESC z ASCII(screenH) ASCII(screenL)
String visualization	27 8 chr0 ...chrn	1B 8 chr0 ...chrn	ESC BS ASCII(chr0) ...ASCII(chrn)
Enable copy of chr on auxiliary serial line	27 3	1B 3	ESC ETX
Enable transparent copy of chr on auxiliary serial line	27 2	1B 2	ESC STX
Disable copy of chr on auxiliary serial line	27 4	1B 4	ESC EOT
Read status of auxiliary serial line handshake	27 5	1B 5	ESC ENQ
Cursor off	27 80	1B 50	ESC P
Static cursor on	27 79	1B 4F	ESC O
Blinking cursor on	27 77	1B 4D	ESC M
Clock set up	27 33 70 hou min sec day mon yea wee	1B 21 46 hou min sec day mon yea wee	ESC ! F ASCII(hou) ASCII(min) ASCII(sec) ASCII(day) ASCII(mon) ASCII(yea) ASCII(wee)
Clock reading	27 33 102	1B 21 66	ESC ! f
Time visualization	27 33 116 r c frm	1B 21 74 r c frm	ESC ! t ASCII(r) ASCII(c) ASCII(frm)
Date visualization	27 33 100 r c frm	1B 21 64 r c frm	ESC ! d ASCII(r) ASCII(c) ASCII(frm)

FIGURE A2: COMMANDS CODES TABLE (2 OF 4)

<i>COMMAND</i>	<i>CODE</i>	<i>HEX CODE</i>	<i>MNEMONIC</i>
<b>Reverse attribute setting</b>	27 48 80	1B 30 50	ESC O P
<b>Enable selected attribute</b>	14	E	SO
<b>Disable selected attribute</b>	15	F	SI
<b>Characters write direction</b>	27 210 dir	1B D2 dir	ESC ASCII(210) ASCII(dir)
<b>Characters zoom setting</b>	27 201 zoom	1B C9 zoom	ESC ASCII(201) ASCII(zoom)
<b>Characters elongation setting</b>	27 207 elong	1B CF elong	ESC ASCII(207) ASCII(elong)
<b>Graphic cursor placement</b>	27 206 y x 0		ESC ASCII(206) ASCII(y) ASCII(x) NUL
<b>Return graphic cursor position</b>	27 211	1B D3	ESC ASCII(211)
<b>One pixel visualization</b>	27 229 y x 0	1B E5 y x 0	ESC ASCII(229) ASCII(y) ASCII(x) NUL
<b>Line drawing</b>	27 203 y1 x1 0 y2 x2 0	1B CB y1 x1 0 y2 x2 0	ESC ASCII(203) ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2) NUL
<b>Rectangle drawing</b>	27 202 y1 x1 0 y2 x2 0	1B CA y1 x1 0 y2 x2 0	ESC ASCII(202) ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2) NUL
<b>Filled rectangle drawing</b>	27 204 y1 x1 0 y2 x2 0	1B CC y1 x1 0 y2 x2 0	ESC EOT ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2) NUL
<b>Graduated axes and grid drawing</b>	27 228 y x 0 ly lx 0 gr dt	1B E4 y x 0 ly lx 0 gr dt	ESC ASCII(228) ASCII(y) ASCII(x) NUL ASCII(ly) ASCII(lx) NUL ASCII(gr) ASCII(dt)
<b>Right oriented filled arrow drawing</b>	27 230 y x 0	1B E6 y x 0	ESC ASCII(230) ASCII(y) ASCII(x) NUL
<b>Right oriented arrow drawing</b>	27 231 y x 0	1B E7 y x 0	ESC ASCII(231) ASCII(y) ASCII(x) NUL
<b>Bottom oriented filled arrow drawing</b>	27 232 y x 0	1B E8 y x 0	ESC ASCII(232) ASCII(y) ASCII(x) NUL
<b>Bottom oriented arrow drawing</b>	27 233 y x 0	1B E9 y x 0	ESC ASCII(233) ASCII(y) ASCII(x) NUL
<b>Leftt oriented filled arrow drawing</b>	27 234 y x 0	1B EA y x 0	ESC ASCII(234) ASCII(y) ASCII(x) NUL
<b>Leftt oriented arrow drawing</b>	27 235 y x 0	1B EB y x 0	ESC ASCII(235) ASCII(y) ASCII(x) NUL
<b>Top oriented filled arrow drawing</b>	27 236 y x 0	1B EC y x 0	ESC ASCII(236) ASCII(y) ASCII(x) NUL
<b>Top oriented arrow drawing</b>	27 237 y x 0	1B ED y x 0	ESC ASCII(237) ASCII(y) ASCII(x) NUL

FIGURE A3: COMMANDS CODES TABLE (3 OF 4)

<i>COMMAND</i>	<i>CODE</i>	<i>HEX CODE</i>	<i>MNEMONIC</i>
<b>Circumference and arcs drawing</b>	27 205 y x 0 r arc.in. n.arc.	1B CD y x 0 r arc.in. n.arc.	ESC ASCII(205) ASCII(y) ASCII(x) NUL ASCII(r) ASCII(arc.in.) ASCII(n.arc.)
<b>Filled circumference with 3 pixels radius drawing</b>	27 238 y x 0	1B EE y x 0	ESC ASCII(238) ASCII(y) ASCII(x) NUL
<b>Circumference with 3 pixels radius drawing</b>	27 239 y x 0	1B EF y x 0	ESC ASCII(239) ASCII(y) ASCII(x) NUL
<b>CAN UART writing</b>	27 128 add dat	1B 80 add dat	ESC ASCII(128) ASCII(add) ASCII(dat)
<b>CAN UART reading</b>	27 129 add	1B 81 add	ESC ASCII(129) ASCII(add)
<b>CAN UART message writing</b>	27 130 byte1 ... byte10	1B 82 byte1 ... byte10	ESC ASCII(130) ASCII(byte1) ... ASCII(byte10)
<b>CAN UART message reading</b>	27 131	1B 83	ESC ASCII(131)

**FIGURE A4: COMMANDS CODES TABLE (4 OF 4)**



APPENDIX B: DISPLAY CHARACTERS TABLE

The following table shows the characters set that can be displayed on QTP G28 display when alphanumeric mode is selected, plus the relative codes.

Higher Lower 4 bit 4 bit	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	0	1	2	3	4	5	6	7	8	9	*	#
xxxx0001	A	B	C	D	E	F	G	H	I	J	K	L
xxxx0010	M	N	O	P	Q	R	S	T	U	V	W	X
xxxx0011	Y	Z	[	\	]	^	_	`	{		}	~
xxxx0100	0	1	2	3	4	5	6	7	8	9	*	#
xxxx0101	A	B	C	D	E	F	G	H	I	J	K	L
xxxx0110	M	N	O	P	Q	R	S	T	U	V	W	X
xxxx0111	Y	Z	[	\	]	^	_	`	{		}	~
xxxx1000	0	1	2	3	4	5	6	7	8	9	*	#
xxxx1001	A	B	C	D	E	F	G	H	I	J	K	L
xxxx1010	M	N	O	P	Q	R	S	T	U	V	W	X
xxxx1011	Y	Z	[	\	]	^	_	`	{		}	~
xxxx1100	0	1	2	3	4	5	6	7	8	9	*	#
xxxx1101	A	B	C	D	E	F	G	H	I	J	K	L
xxxx1110	M	N	O	P	Q	R	S	T	U	V	W	X
xxxx1111	Y	Z	[	\	]	^	_	`	{		}	~

FIGURE B1: ALPHANUMERIC MODE CHARACTERS TABLE



## APPENDIX C: LABELS INSERTION

The video terminal **QTP G28** is provided of three personalization labels that can be used to add the user's logo, some keys and LEDs meaning in accord with the developed application. In the following photo and description are described all the necessary information for the first insertion, or the subsequent replacement of these labels.

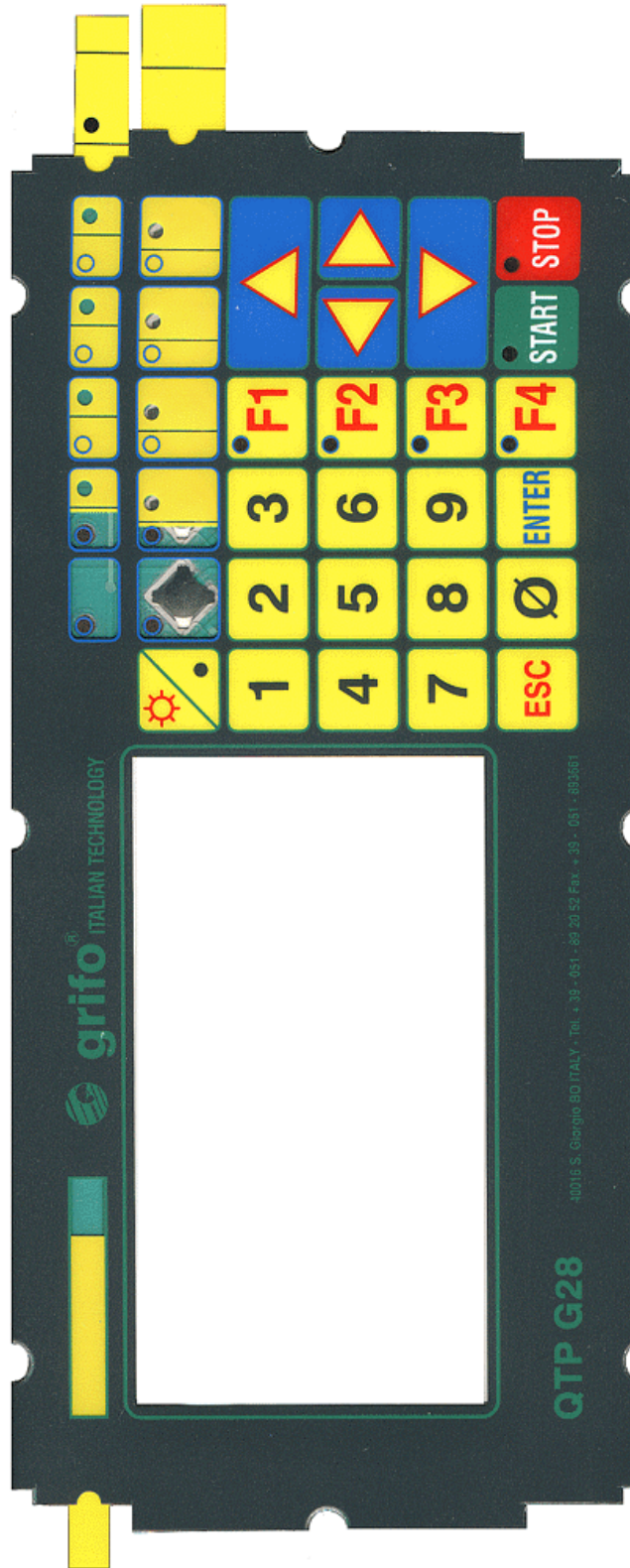


FIGURE C1: LABEL INSERTION ON THE TERMINAL

The user must execute the operation below described, to insert the personalization label on the terminal slots:

- 1) Unscrew the 4 black screws from the front panel.
- 2) Take the rear panel out.
- 3) Unscrew the 8 screws and their grey plastic spacers which are placed in the rear side of the frame.
- 4) Separate the keyboard+printed circuit group from the black plastic frame.
- 5) Now the keyboard is ready for inserting the 3 personalization labels (see previous figure).
- 6) Remount the **QTP G28** following the previous instructions but on the back way.

The personalization labels are supplied with the acquired terminal, inside a proper envelope, tied on the front panel of **QTP G28**. If the user needs other labels kits, he can order them with the **.KCAR-G28** code, or he can realize them by using a proper material that have a proper solidity and a small thickness.

About the effective personalization of the labels, some different method can be used: realize some autoadhesive and transparent labels (through an electronic label maker) and then stick them on the **QTP G28** labels, or alternatively print directly the supplied labels (through plotter, laser printers, transfer card, etc.).

## APPENDIX D: ALPHABETICAL INDEX

**Simbols**9 bits **28****A**ADDS View Point **26**Alphanumeric **27**Alphanumeric **42**ALPHANUMERIC CURSOR PLACEMENT, command **35**Assistance **1**Autorepeat **3, 7, 29**Auxiliary serial line **6, 18, 26, 44****B**Back light **3**Back up **4**BACKSPACE, command **33**Badge **5**BasicCAN **5**Baud Rate **6, 26, 28**BEEP, command **42**Blinking **3, 7**BLINKING CURSOR ON, command **45**BOTTOM ORIENTED ARROW DRAWING, command **53**BOTTOM ORIENTED FILLED ARROW DRAWING, command **53**Brightness **6, 32**Buzzer **3, 6****C**CAN interface **5, 6, 22, 24, 26, 27, 58**CAN INTERFACE COMMANDS **58**CAN UART MESSAGE READING **58**CAN UART MESSAGE WRITING **58**CAN UART READING, command **58**CAN UART WRITING, command **58**CARRIAGE RETURN, command **35**CARRIAGE RETURN+LINE FEED, command **35**CHARACTERS ELONGATION SETTING, command **49**Characters set **B-1**Characters size **6**Characters visualization **32**CHARACTERS WRITE DIRECTION, command **48**CHARACTERS ZOOM SETTING, command **49**Chr lenght **6**

CIRCUMFERENCE AND ARCS DRAWING, command 56  
CIRUMFERENCE WITH 3 PIXELS RADIUS DRAWING, command 57  
CLEAR END OF LINE, command 33  
CLEAR END OF PAGE, command 33  
CLEAR LINE, command 33  
CLEAR PAGE, command 33  
CLOCK READING, command 46  
CLOCK SET UP, command 46  
Commands codes tables A-1  
COMMANDS FOR AUXILIARY SERIAL LINE MANAGEMENT 44  
COMMANDS FOR CHARACTERS ERASURE 33  
COMMANDS FOR CURSOR ATTRIBUTES MANAGEMENT 45  
COMMANDS FOR CURSOR POSITIONING 34  
COMMANDS FOR EEPROM 36  
COMMANDS FOR KEYBOARD MANAGEMENT 40  
COMMANDS FOR LEDS MANAGEMENT 38  
COMMANDS FOR REAL TIME CLOCK MANAGEMENT 46  
COMMANDS FOR VARIOUS FUNCTIONS 42  
Communication 4, 12, 28  
Connectors 7  
    CN1 10  
    CN2 11  
    CN3A 12  
    CN3B 18  
    CN4 20  
    CN6 22  
Consumption 8  
Contrast 6, 32  
CPU 6  
Current Loop 4, 12, 17  
    Active 16, 24  
    Passive 16, 17, 24  
Cursor 34, 35, 45, 49  
CURSOR DOWN, command 34  
CURSOR LEFT, command 34  
CURSOR OFF, command 45  
CURSOR UP, command 34

## D

Date 4, 47  
DATE VISUALIZATION, command 47  
Debouncing 7  
Default configuration 24, 31  
Dimension 6, 7  
DISABLE COPY OF CHR ON AUXILIARY SERIAL LINE, command 44  
DISABLE SELECTED ATTRIBUTE, command 48  
Display 3, 6

**E**

Earth line **10**

EEPROM **4, 6, 36**

ENABLE COPY OF CHR ON AUXILIARY SERIAL LINE, command **44**

ENABLE SELECTED ATTRIBUTE, command **48**

ENABLE TRANSPARENT COPY OF CHR ON AUXILIARY SERIAL, command **44**

EPROM **43**

External loads **8, 11**

**F**

Features

Electric **8**

General **2, 6**

Physical **6**

FILLED CIRCUMFERENCE WITH 3 PIXELS RADIUS DRAWING, command **56**

FILLED RECTANGLE DRAWING, command **51**

Filter **10**

Firmware **1, 26**

Fluorescent lamp **3**

Frontal view **9**

**G**

GRADUATED AXES AND GRID DRAWING, command **51**

Graphic **27**

GRAPHIC CURSOR PLACEMENT, command **49**

GRAPHICS COMMAND **48**

Ground connection **11**

**H**

Handshake **13, 24, 32, 44**

Hardcopy **18**

HOME, command **34**

Humidity **7**

**I**

Installation **10**

IP65 frontal **6**

**J**

Jumpers **24**

**K**

KEY RECONFIGURATION, command **40**

Keyboard **3, 29**

Keyclick **3, 26, 40**  
KEYCLICK OFF WITH MEMORIZATION, command **40**  
KEYCLICK OFF WITHOUT MEMORIZATION, command **41**  
KEYCLICK ON WITH MEMORIZATION, command **40**  
KEYCLICK ON WITHOUT MEMORIZATION, command **41**  
Keys **6**  
Keys code **31, 40**  
Keys map **30**

## L

Labels **3, 6, C-1**  
LED ACTIVATION, command **38**  
LEDs **3, 6**  
LEDS ACTIVATION WITH MASK, command **38**  
LEDs map **39**  
LEFT ORIENTED ARROW DRAWING, command **54**  
LEFT ORIENTED FILLED ARROW DRAWING, command **54**  
Life Byte **36**  
LINE DRAWING, command **50**  
Lithium battery **4, 6**

## M

Magnetic card **5**  
Master Slave **28**  
Mounting **1, 6**

## N

Name **26, 28**  
Network connection **15, 16, 28**

## O

ONE PIXEL VISUALIZATION, command **50**  
Options **4, 5, 20**

## P

Parity **6, 26**  
PeliCAN 2.0B **5**  
Personalization **C-1**  
Photo **21**  
Pixel size **6**  
Point to point connection **14, 16, 17**  
Power **8**  
Power supply **5, 8, 10**  
Primary serial line **6, 12, 26**  
Printer **18, 44**



**R**

READ BLOCK OF DATA FROM EEPROM, command 37  
READ STATUS OF AUXILIARY SERIAL LINE HANDSHAKE, command 44  
READING OF LIFE BYTE, command 36  
READING OF VERSION NUMBER, command 42  
Real Time Clock 4, 6, 46  
Receive buffer 6, 32  
RECTANGLE DRAWING, command 50  
Relay 4, 6, 8, 20  
RELAY ACTIVATION, command 42  
RELAY DEACTIVATION, command 42  
Release 1  
REQUEST FOR EEPROM WRITING POSSIBILITY, command 36  
Resources 6  
RETURN ALPHANUMERIC CURSOR POSITION, command 35  
RETURN GRAPHIC CURSOR POSITION, command 50  
REVERSE ATTRIBUTE SETTING, command 48  
RIGHT ORIENTED ARROW DRAWING, command 52  
RIGHT ORIENTED FILLED ARROW DRAWING, command 52  
RS 232 4, 12, 13, 18  
RS 422 4, 8, 12, 14, 24  
RS 485 4, 8, 12, 14, 15, 24

**S**

Screen 43  
SCREEN VISUALIZATION, command 43  
Serial lines 4, 6  
Set up 4, 26  
SETTING OF ALPHANUMERIC VISUALIZATION MODE, command 42  
SETTING OF GRAPHIC VISUALIZATION MODE, command 43  
SJA 1000 58  
Slave 28  
Software 26  
STATIC CURSOR ON, command 45  
Stop Bit 6, 26  
STRING VISUALIZATION, command 43  
Switching 5

**T**

Technical specifications 6  
Temperature 7  
Termination circuitry 8, 15, 23, 24  
Time 4, 46  
Time Out 28  
TIME VISUALIZATION, command 46  
Timing 7, 28  
TOP ORIENTED ARROW DRAWING, command 55

TOP ORIENTED FILLED ARROW DRAWING, command **55**

## V

VCL **17**

Visible area **6**

Visualization mode **27**

## W

Warranty **1**

Weight **6**

WRITE BLOCK OF DATA ON EEPROM, command **37**

WRITING OF LIFE BYTE, command **36**

## Z

Zoom **49**