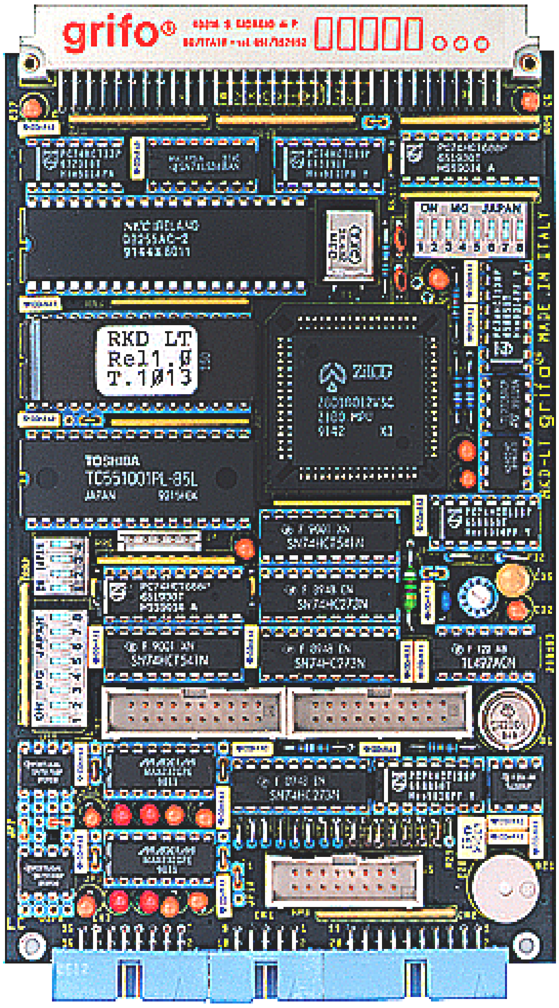
RKD LT

Remote Keyboard Display controller

TECHNICAL MANUAL



grifo®
ITALIAN TECHNOLOGY

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

RKD LT Edition 5.00 *Rel. 21 September 1999*

RKD LT

Remote Keyboard Display controller

TECHNICAL MANUAL

Intelligent video terminal for **ABACO®** inteligent bus; size 100x160 mm single EUROPE format; management of FUTABA 20x2 up to 40x4 characters alphanumeric displays; management of graphic LCD displays mounting HITACHI or TOSHIBA controller, 120x64 up to 240x128 pixels; on board LCD contrast regulation trimmer; management of a 56 keys BG keyboard or generic a 7x8 matrix keyboard, featuring Shift, Control, Caps Lock, Autorepeat and Keyclick functions; management of 8 external LEDs; on board buzzer for acoustic indications and keyclick; RS 232, 422 or Current Loop serial communication line; auxilary RS 232 serial line to echo the received data; Baud Rate from 2400 up to 19200 baud; **ABACO®** BUS addressing space as low as 2 bytes; 8 pins Dip Switch to set the I/O address; up to 2KBytes of E^{2}PROM to store messages; capability to store alphanumeric screens in EPROM for later visalization; one 8 pins Dip Switch and one 4 pins Dip Switch for set-up and configuration; only one + 5Vdc power supply; 130 mA (without display and keyboard); optionally available with **GDOS 80** operating system already installed to let the User develop his/her own management programs

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

RKD LT Edition 5.00 *Rel. 21 September 1999*

DOCUMENTATION COPYRIGHT BY grifo®, ALL RIGHTS RESERVED

IMPORTANT

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

SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

⚠ Attention: Generic danger

⚡ Attention: High voltage

Trade Marks

GENERAL INDEX

GENERAL INDEX

INTRODUCTION

The use of these devices has turned - IN EXCLUSIVE WAY - to specialized personnel.

The purpose of this handbook is to give the necessary information to the cognizant and sure use of the products. They are the result of a continual and systematic elaboration of data and technical tests saved and validated from the manufacturer, related to the inside modes of certainty and quality of the information.

The reported data are destined- IN EXCLUSIVE WAY- to specialized users, that can interact with the devices in safety conditions for the persons, for the machine and for the enviroment, impersonating an elementary diagnostic of breakdowns and of malfunction conditions by performing simple functional verify operations , in the height respect of the actual safety and health norms.

The informations for the installation, the assemblage, the dismantlement, the handling, the adjustment, the reparation and the contingent accessories, devices etc. installation are destined - and then executable - always and in exclusive way from specialized warned and educated personnel, or directly from the TECHNICAL AUTHORIZED ASSISTANCE, in the height respect of the manufacturer recommendations and the actual safety and health norms.

The devices can't be used outside a box. The user must always insert the cards in a container that rispect the actual safety normative. The protection of this container is not threshold to the only atmospheric agents, but specially to mechanic, electric, magnetic, etc. ones.

To be on good terms with the products, is necessary guarantee legibility and conservation of the manual, also for future references. In case of deterioration or more easily for technical updates, consult the AUTHORIZED TECHNICAL ASSISTANCE directly.

To prevent problems during card utilization, it is a good practice to read carefully all the informations of this manual. After this reading, the user can use the general index and the alphabetical index, respectly at the begining and at the end of the manual, to find information in a faster and more easy way.

CARD VERSION AND FIRMWARE

The present handbook is reported:

- Card **RKD LT** :	**181093** and later
- Firmware **TRKDGL**:	**2.1** and later
- Firmware **TRKDAF**:	**1.1** and later

The validity of the bring informations is subordinate to the number of the card release. The user must always verify the correct correspondence among the two denotations. On the card the release number is present in more points both board printed diagram (serigraph) and printed circuit (for example along lthe border on the component side, near trimmer R27); the firmware version is reported in the label on the EPROM.

GENERAL FEATURES

RKD LT (**R**emote **K**eyboard **D**isplay controller) is an intresting peripherial intelligent board, size 100x160 mm standard EUROPE format, designed to solve efficently and economically the problem of user interfacing in the industrial sector.
The board is complete video terminal, capable to manage in total autonomy both the acquisition and the visualization of informations.
Communication between **RKD LT** and its master unit can be serial or parallel, in order to satisfy every need for speed and suitability. These features allow any card provided with a serial interface and any **ABACO®** BUS control card (**GPC®** serie) to interface to the **RKD LT** intelligent controller producing the remarkable advantage not to have to develop specific management software, reducing so code space occupation and development time, and increasing the execution speed.
RKD LT can interface to **FUTABA fluorescent alphanumeric** displays capable of resolutions from **20x2** up to **40x4** characters, or **LCD graphic displays mounting HITACHI or TOSHIBA controller**, capable of resolutions from **120x64** up to **240x128** pixels.
he board can manage directly the **56 keys BG keyboard**, or a generic 7x8 keys matrix keyboard, featuring the **Shift**, **Caps Lock**, **Control, Autorepeat** and **Keyclick** functions, in order to make more confortable and easy the phase of data input.
Up to **8 external LEDs** can be connected to the **RKD LT** board to provide visual signalations such as alarms, machine states, etc.
RKD LT board is capable to solve easily all the difficulties relative to messages visualization, graphic displays, visual and acoustic signaalations, keyboard acquisition, hardcopy, etc.

- Intelligent video terminal for **ABACO® BUS**
- Size 100x160 mm single EUROPE format
- **High speed parallel** communication or **serial** communication
- Management of **FUTABA alphanumeric displays 20x2** up to **40x4** characters
- Management of **graphic LCD displays mounting HITACHI or TOSHIBA controller**, **120x64** up to **240x128** pixels
- On board LCD **contrast** regulation trimmer
- Management of a **56 keys BG keyboard** or generic a 7x8 matrix keyboard, featuring **Shift, Caps Lock, Control, Autorepeat** and **Keyclick** functions
- Management of **8 external LEDs**
- On board **buzzer** for acoustic indications and keyclick
- **RS 232, RS 422** or **Current Loop** serial communication line
- Auxilary **RS 232 serial line to echo** the received data
- **Baud Rate** from **2400** up to **19200 baud**
- **ABACO®** BUS addressing space as low as **2 bytes**
- 8 pins **Dip Switch** to set the I/O address
- Up to **2KBytes** of **EEPROM** to store messages, parameters, etc.
- Capability to **store alphanumeric screens** in **EPROM** for later visalization
- One **8 pins Dip Switch** and one **4 pins Dip Switch** for set-up and configuration
- Only one + **5Vdc** power supply; **130 mA** (without display and keyboard)
- Optionally available with **GDOS 80** operating system already installed to let the User develop his/her own management programs

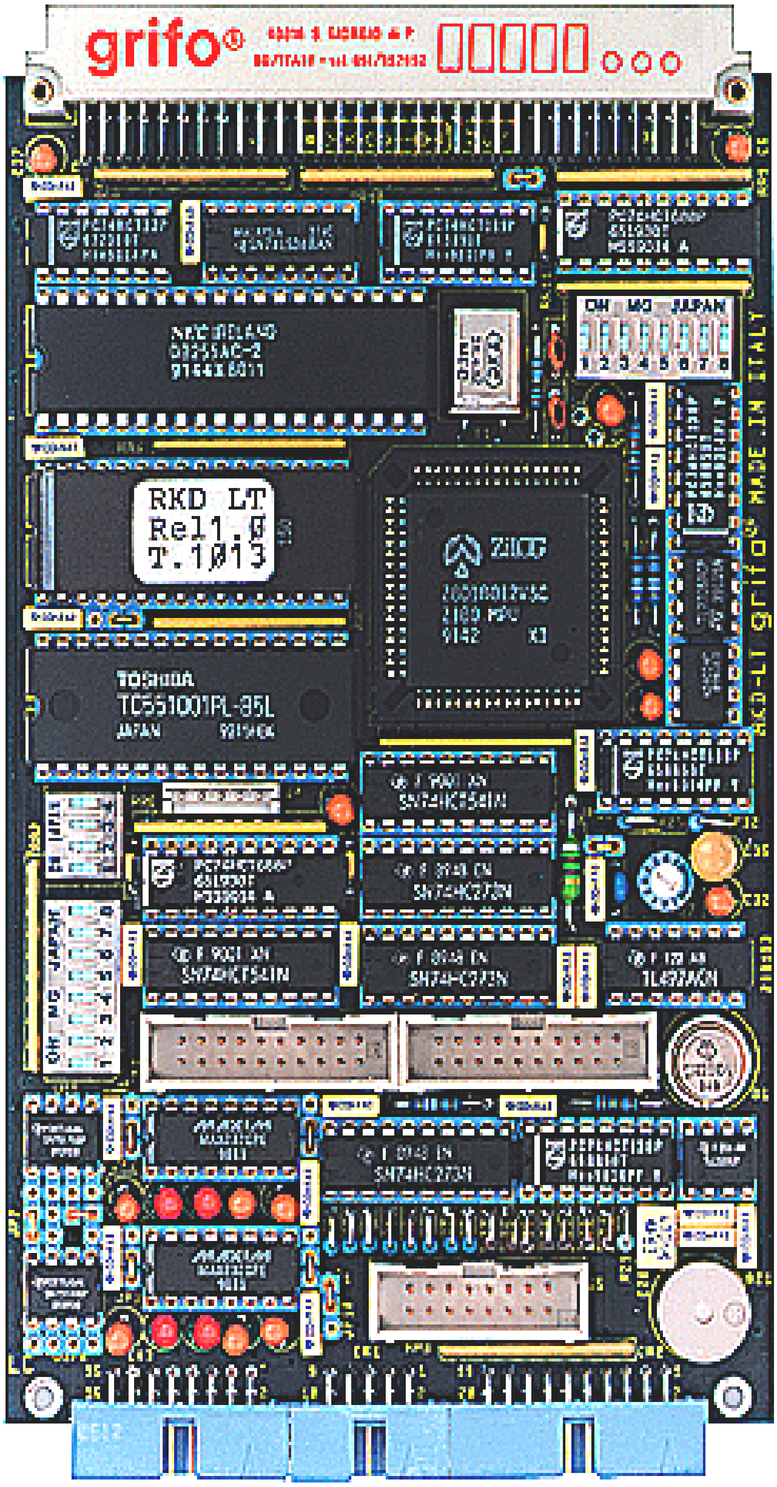


FIGURE 1: CARD PHOTO

COMMUNICATION TO CONTROL UNIT

Communication between **RKD LT** and its master unit can be serial or by the **ABACO®** BUS, a high speed parallel data path, in order to satisfy every need for communication velocity and hardware needs.

ABACO® BUS PARALLEL INTERFACE

RKD LT board is provided with an **ABACO®** BUS paralel interface capable to speed up all the communication operations.
This section manages the communication between the on board CPU and the master control card (**GPC®** serie); in detail it manages the I/O mapping of the board in the addressing space by simply acting on a 8 pins Dip Switch, the board takes only two bytes of space. The **ABACO®** BUS is degined to work in the industrial automation environment providing an 8 bits data path with an 8 bits address word.

MAIN SERIAL LINE

RKD LT board is provided with a serial communication line which can be buffered as RS 232 or it can be buffered as current loop or RS 422 provided that the auxilary serial line is kept as RS 232.
The communiucation parameters are: 8 bits, no parity, 1 stop bit, the BaudRate is selectable amongst 2400, 4800, 9600, 19200 Baud. The User can set this last parameter by acting on the proper Dip Switches.

AUXILIARY SERIAL LINE

RKD LT board is provided with an auxiliary serial communication line which can be buffered as RS 232 or it can be buffered as current loop or RS 422 provided that the main serial line is kept as RS 232.
This line can perform received data echo functions, making possible harcopy operations on a different serial device, like a printer, that the board can drive directly.
The communiucation parameters are: 8 bits, no parity, 1 stop bit, the baud rate is selectable amongst 2400, 4800, 9600, 19200 Baud. The User can set this last parameter by acting on the proper Dip Switches.

KEYBOARD

RKD LT board is able to manage directly a **56 keys BG keyboard** or a 7x8 keys generic matrix keyboard; the keys are acquired by the board itself with AutoRepeat, Shift, Control and Caps-Lock functions performing.
It is also possible to disable the KeyClick function, that is the audio feedback, made by activating the on board buzzer, when a key is pressed.

BUZZER

The **RKD LT** board is provided with a circuitery able to utter a constant sound employing a capacitiva buzzer. This circuitry can be software enabled to perform beeps or, using the KeyClick function, can be enabled when a key is hit.

FLUORESCENT ALPHANUMERIC DISPLAYS

The **RKD LT** board can manage FUTABA fluorescent alphanumeric displays.
In detail it can manage displays with rows made of 20 or 40 characters: that is the 20x2, 20x4, 40x1, 40x2 and 40x4 models.

LCD GRAPHIC DISPLAYS

The **RKD LT** board can manage graphic displays mounting a HITACHI or TOSHIBA controller, both in alphanumeric and in graphic mode.
In detail these models can be connected: 120x64 pixels, corresponding to 15x8 characters; 128x128 pixels (16x16 characters); 240x64 pixels (40x8 characters); 160x128 pixels (20x16 characters); 240x128 pixels (40x16 characters).
Contrast regulation is made by an on board circuit provided with a trimmer to perform this operation.

EXTERNAL LEDS

The **RKD LT** board can manage 8 external LEDs, in the common cathod connection. The LEDs can be used to make visual signalations like alarms, showing machine status, etc.
The same lines may also be used for a low-level management of a seven segments display.

EEPROM

The **RKD LT** board is provided with an EEPROM, whose size ranges from 512 Bytes to 2K Bytes, to store messages,
The main purpose of this section is to give a non volatile memory device where to store informations even when th master unit can't perform this feature.

FIRMWARE TRKDGL

It is the terminal emulation firmware, able to run both in graphic and alphanumeric mode.
It performs the scansion and the debouncing on the BG keyboard; every key pressed is recognized, coded and transmitted to the control system by the desired communication mode (serial or parallel).
In addition the special keys are managed in autonomy, assigning them their functions, also the AutoRepeat, Shift, Control, Caps-Lock and Keyclick functions are managed.

Command and/or data received by the control system are interpreted and executed interacting with the display; remarkable commands are: cursor positioning, lines, rectangles and circumferences drawing, clear diplay, graphic zoom of characters, etc.
In alphanumeric mode, the **RKD LT** board can emulate the **TVI 950** o **r Adds-ViewPoint** standard terminal protocols.
A 64KBytes EPROM area is reserved by th firmware to store alphanumeric screens that need to be displayed very quickly, this allows also a remarkable memory saving for the master control unit.
To create, correct and geberate the alphanumeric screens to be diplayed by **RKD LT**; **grifo®** has realized a powerful text editor running on P. C., called **RKD_EDIT**, available both in Italian and in English. The software is also able to program directly the EPROM on **RKD LT**.
The User interface is based on scroll-down menus and the employ of the mouse, making possible to write the text quickly and comfortably.
For further informations please refer to the next chapters.

FIRMWARE TRKDAF

The alphanumeric terminal emulatore designed to work with fluorescent displays.
It performs the scansion and the debouncing on the BG keyboard; every key pressed is recognized, coded and transmitted to the control system by the desired communication mode (serial or parallel).
In addition the special keys are managed in autonomy, assigning them their functions, also the AutoRepeat, Shift, Control, Caps-Lock and KeyClick functions are managed.
Command and/or data received by the control system are interpreted and executed interacting with the display; remarkable commands are: cursor positioning, clear line, turn off or on cursor, etc.
The **RKD LT** board can emulate the **TVI 950** o **r Adds-ViewPoint** standard terminal protocols.
Also this firmware reservs a 64KBytes EPROM area to store alphanumeric screens that need to be displayed very quickly, this allows also a remarkable memory saving for the master control unit.
Also for this firmware it is possible to create quicly the text screenshots using **RKD_EDIT**, as described in the previous paragraph.
For further informations please refer to the next chapters.

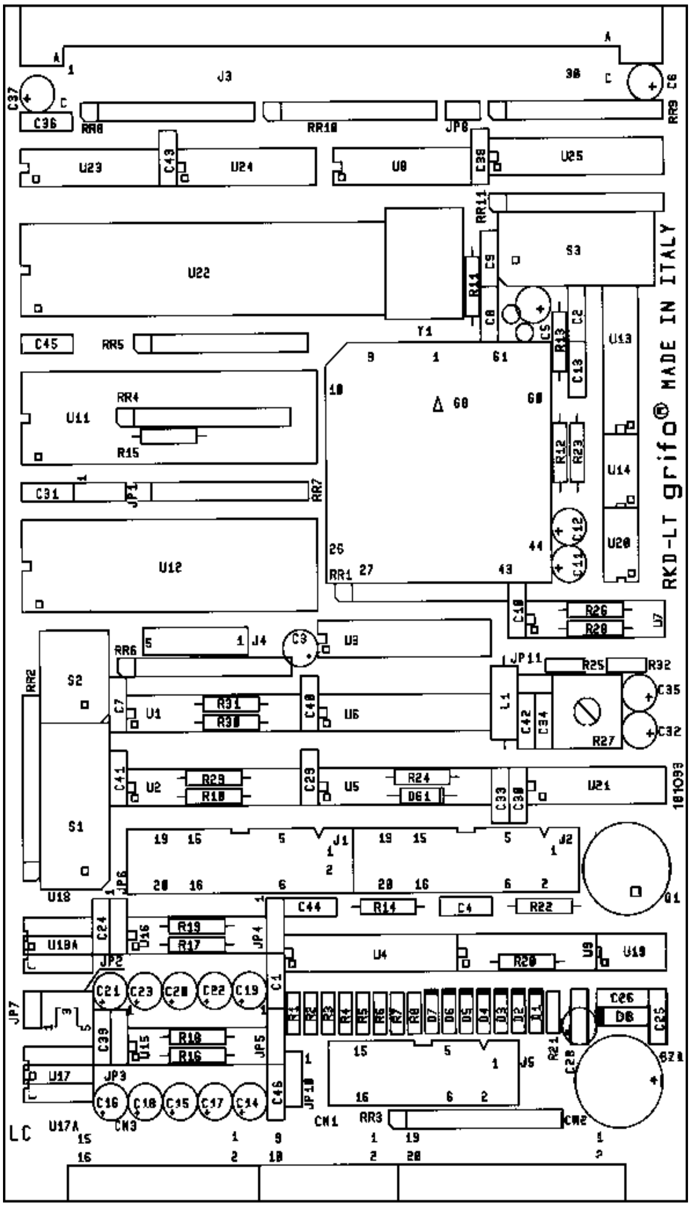


Figure 2: Components map

TECHNICAL FEATURES

GENERAL FEATURES

Devices:

- **Interface for** 56 keys BG Keyboard**, or generic** 7x8 matrix keyboard**.**
- **Interface for** graphic LCD display **mounting TOSHIBA or HITACHI controller, with trimmer to regulate the contrast.**
- **Interface for FUTABA** fluorescent alphanumeric display **.**
- **Interface to drive directly** 8 external LEDs**.**
- Buzzer **for BELL signalations or audio feedback of a key pressed.**
- 2 **Full Duplex** RS 232, RS 422 **or** Current-Loop **serial lines.**
- EEPROM **to store messages, parameters, etc. Max 2 Kbytes.**
- 2 dip switch **(11 total pins) for the board configuration.**
- **Parallel interface for ABACO®** BUS **.**
- **8 pins** Dip switch **to set the I/O address.**

On board CPU: **Z180** with 18.432 MHz quartz.

Parallel Interface: 8 bit BUS for data and address.
256 byte total addressing space.
2 byte of I/O space taken.

Serial Communication: BAUD RATE: 2400, 4800, 9600 or 19200 Baud.
1 Stop Bit.
No Parity.
8 Bit.

Managed Displays: **Alphanumeric fluorescent displays**

Futaba M202SD01BA and compatible - 20x2 characters “small”
Futaba M202SD08GK and compatible - 20x2 characters “wide”
Futaba M204SD01AA and compatible - 20x4 characters
Futaba M40SD04GJ and compatible - 40x1 characters
Futaba M402SD07GK and compatible - 40x2 characters
Futaba M404SD01BA and compatible - 40x4 characters (**+IAF 404)**

LCD graphic displays mounting Toshiba T6963C controller

Toshiba TLX-1021 and compatible - 120x64 pixels
Toshiba TLX-711A and compatible - 240x64 pixels
Toshiba TLX-1391 and compatible- 128x128 pixels
Toshiba TLX-1013 and compatible-160x128 pixels
Toshiba TLX-1301 and compatible- 240x128 pixels
Toshiba TLX-1091 and compatible- 240x128 pixels

LCD graphic displays mounting Hitachi HD61830B controller

Varitronix MGLS240128V2 and compatible - 240x128 pixels

PHISICAL FEATURES

Size:	100x160 mm standard EUROPE format.
Weight:	185 g
Temperature range:	From 0 to 70 centigrad degreeses
Relative Humidity:	20% up to 90% (without condens).
Connectors:	J1: 20 pins,male, vertical, low profile connector J2: 20 pins,male, vertical, low profile connector J3: DIN 41612 64 pins M 90 degreeses A+C type C J5: 16 pins,male, vertical, low profile connector CN1: 10 pins,male, 90 degreeses, low profile connector CN2: 20 pins,male, 90 degreeses, low profile connector CN3: 16 pins,male, 90 degreeses, low profile connector

ELETCTRIC FEATURES

Power Supply Voltage:	+5 Vdc.		
Consumption:	**130 mA** without keyboard and display.		
RS 422 termination ciruit:	Line termination resistance	=	**120 Ω**
	Positive pin pull-up resistance	=	**not present**
	Negative pin pull-up resistance	=	**not present**
Serial Resistance for LEDs:	**330 Ω**		
Voltage range for LCD contrast:	**-6 ÷ -12 Vdc** or **-7.5 ÷ -15 Vdc**		

INSTALLATION

In this chapter there are all information for a right installation and correct use of the card. The User can find the location and functions of each connectors, LEDs, jumpers and some explanatory diagram.

CONNECTIONS

The **RKD LT** module has seven connectors that can be linked to other devices or directly to the field, according to system requirements. In this paragraph there are connectors pin out, a brief signals description (including the signal direction), connectors location and some electrical diagrams that show the on board circuit of each connector.

CN1 - EXTERNAL LEDS CONNECTOR

Connector CN1, used to connect 8 external LEDs to the board is a 2.54 mm pitch, 10 pins,male, 90 degreeses, low profile connector. Here follows its pinout and an example of connection for external LEDs.
The connection mode must be common cathod as shown in figure 4.

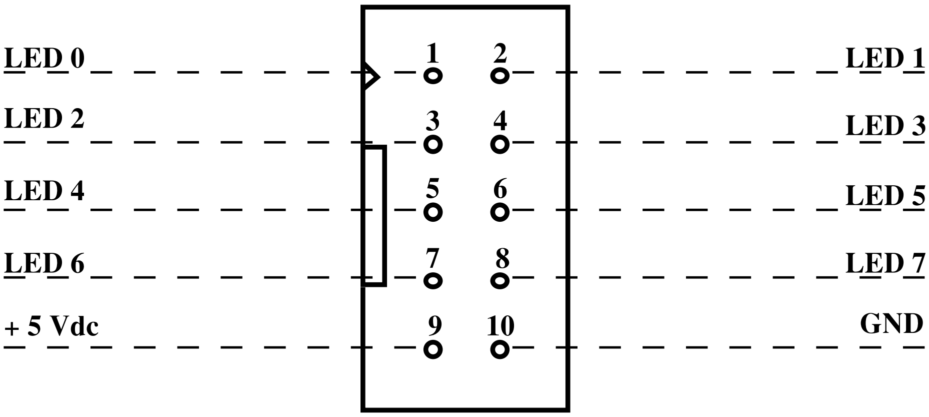


FIGURE 3: CN1 - EXTERNAL LEDS CONNECTOR

Symbols description:

LED n	=	O	- Output signal for n external LED
GND	=		- LEDs common cathod ground signal
+5 Vdc	=	O	- Power supply voltage for eventual externaml loads

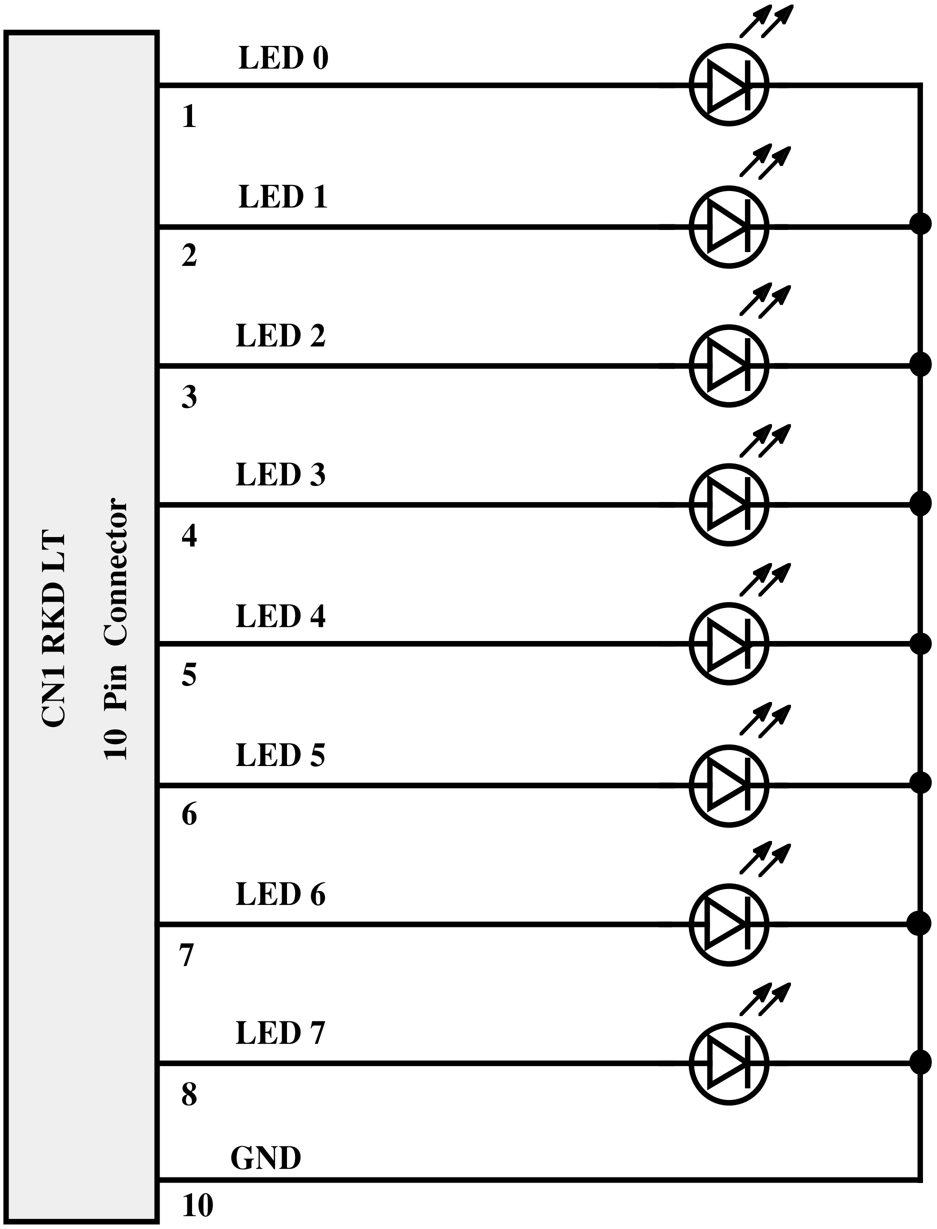


FIGURE 4: COMMON CATHOD EXTERNAL LEDS EXAMPLE CONNECTION DIAGRAM

CN2 - CONNECTOR FOR BG KEYBOARD

The CN2 connector, dedicated to connect the BG keyboard, is a 2.54 mm pitch, 20 pins,male, 90 degreeses, low profile connector. Here follows it pin-out, a frontal view of the keyboard and the keyboard's electric diagram, to ease the User in designing his/her own 7x8 matrix keyboard.

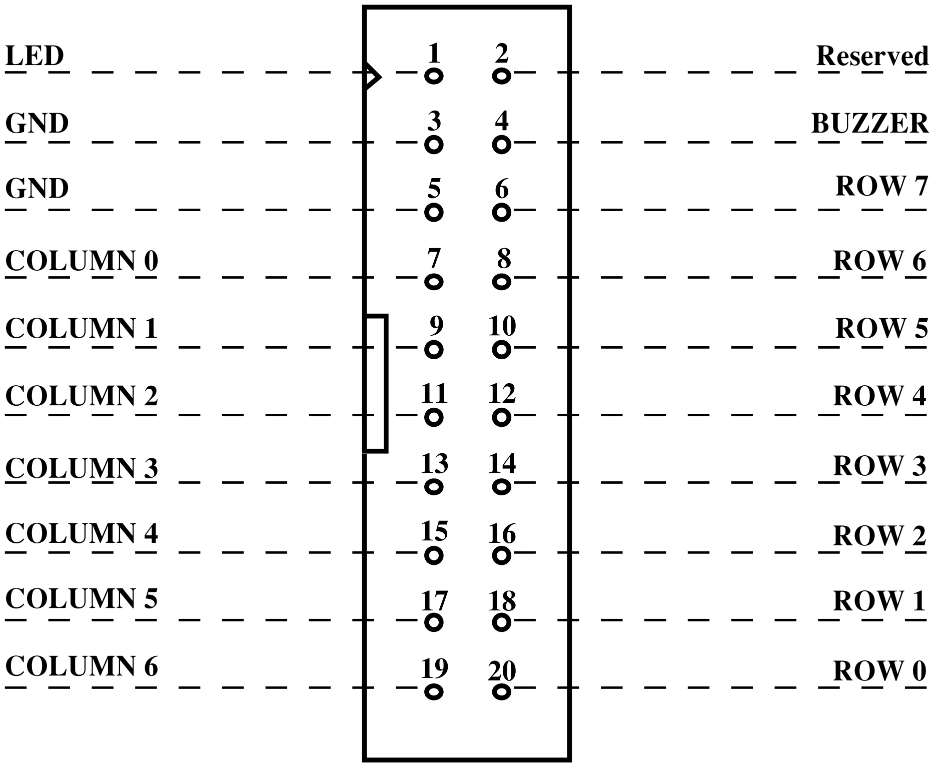


FIGURE 5: CN2 - CONNECTOR FO BG KEYBOARD

Symbols description:

ROW n	=	I	- Input signal to connect the n-th row of 7x8 matrix
COLUMN n	=	O	- Output signal to connect the n-th column of 7x8 matrix
LED	=	O	- Output signal to manage the LED of BG keyboard
BUZZER	=	O	- Output signal to manage the buzzer of BG keyboard
Reserved	=		- Reserved sugnal, must be not connected
GND	=		- Ground signal

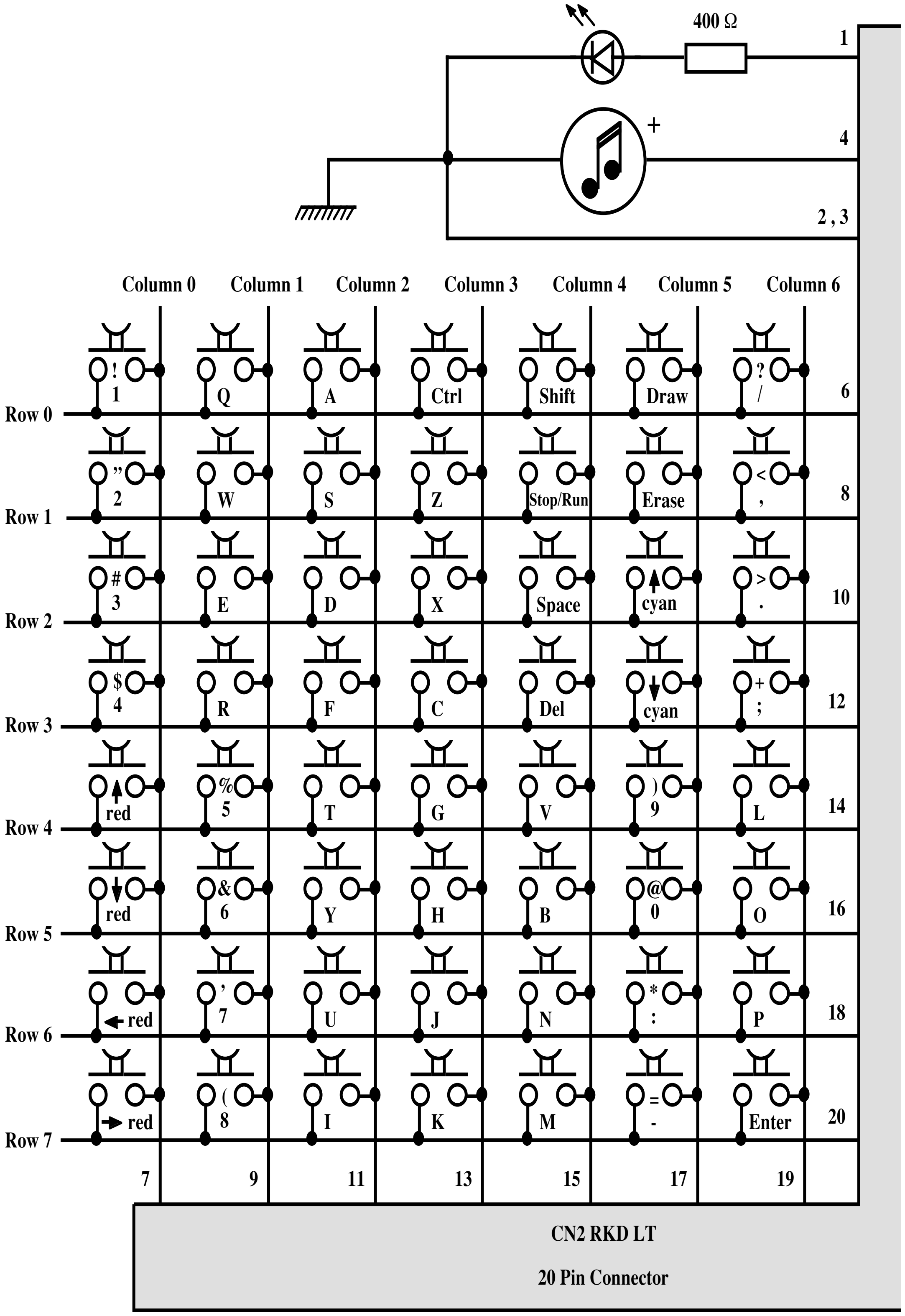


FIGURE 6: BG KEYBOARD ELECTRIC DIAGRAM

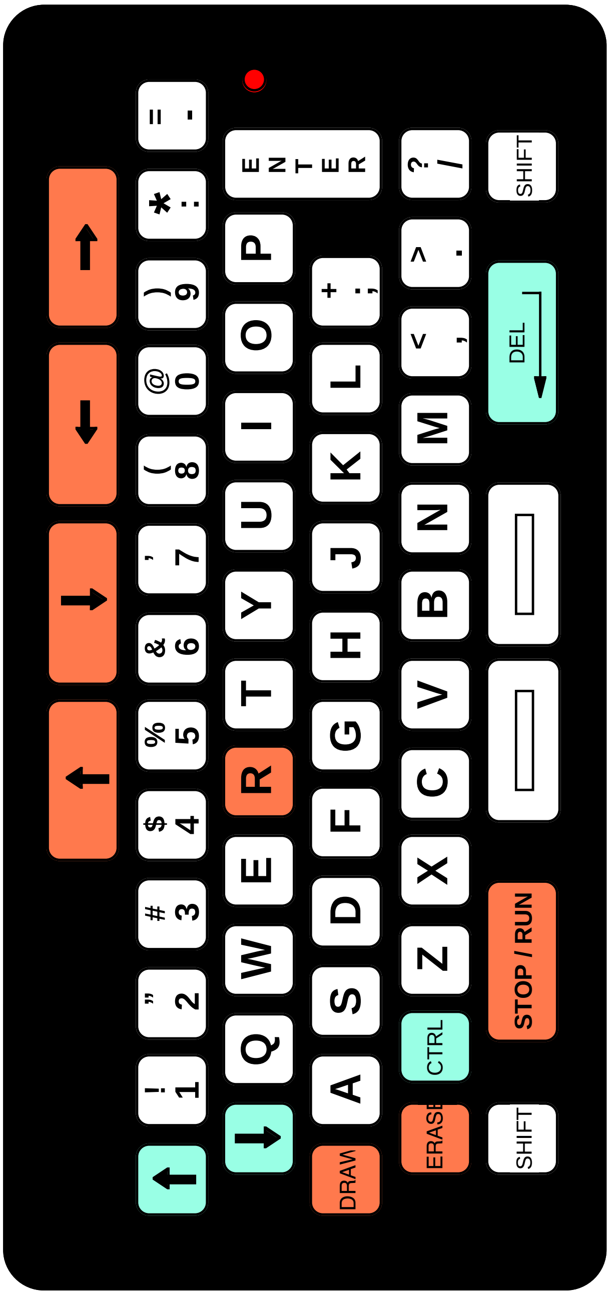


FIGURE 7: FRONT VIEW OF BG KEYBOARD

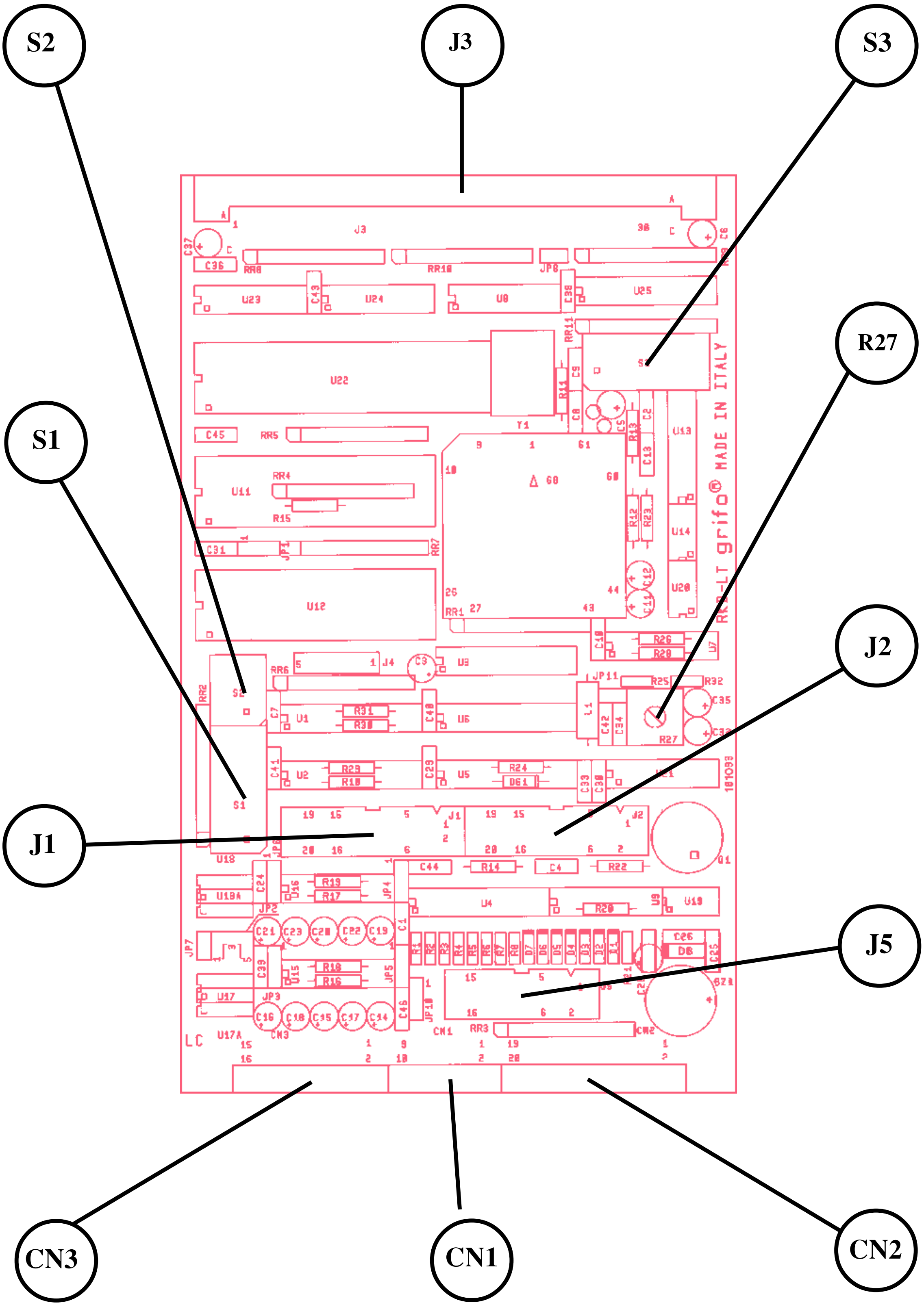


FIGURE 8: CONNECTORS, DIP SWITCH, TRIMMER, ETC. LOCATION

CN3 - SERIAL LINES CONNECTOR

The connector used for serial lines, called CN3, is a 2.54 mm pitch, 16 pins,male, 90 degreeses, low profile connector. Here follows the pin out of signals, that has been designed to minimize interferences and to easy the connection to the field, while the signals respect the CCITT rules relative to each of the communication standard used.

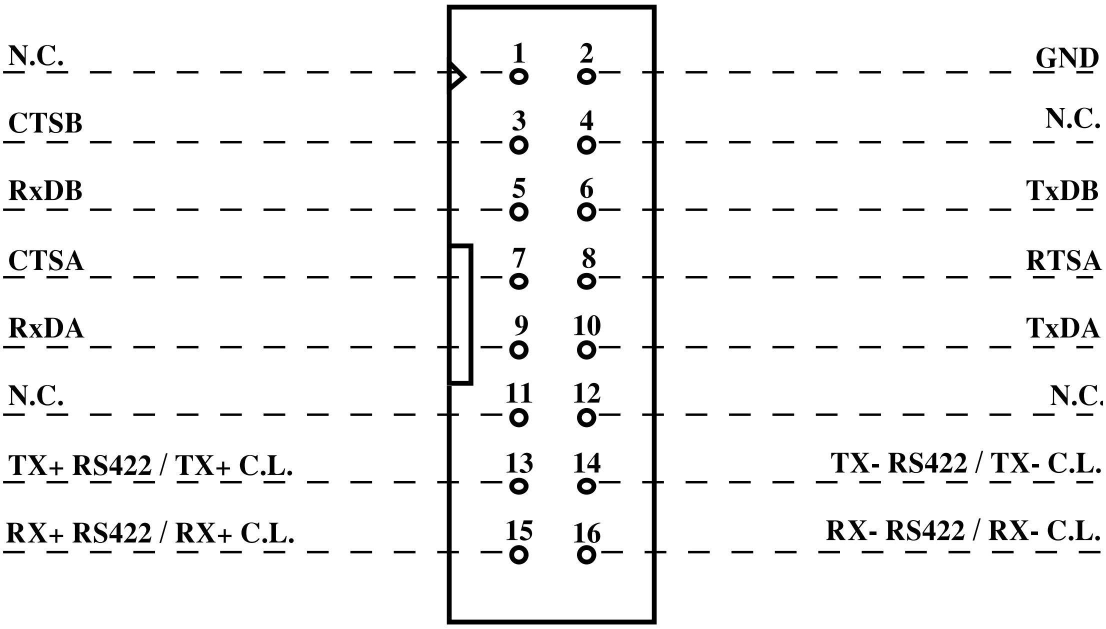


FIGURE 9: CONNECTOR FOR SERIAL LINES

Symbols description:

TxDA - TxDB = O - Transmit Data: transmit data signal for main (**A**) and auxiliary (**B**) RS 232 serial line

RxDA - RxDB = I - Transmit Data: transmit data signal for main (**A**) and auxiliary (**B**) RS 232 serial line

CTSA - CTSB = I - Clear To Send: clear to send signal for main (**A**) and auxiliary (**B**) RS 232 serial line

RTSA = O - Request To Send: request to send signal for main (**A**) RS 232 serial line (not present on auxiliary serial line)

RX- RS422 / RX- C.L. = I - Receive Data Negative: Main and auxiliary, RS 422 or Current Loop serial line, reception negative bipolar signal

RX+ RS422 / RX+ C.L. = I - Receive Data Positive: Main and auxiliary, RS 422 or Current Loop serial line, reception positive bipolar signal

TX- RS422 / TX- C.L. = O - Transmit Data Negative: Main and auxiliary, RS 422 or Current Loop serial line, transmission negative bipolar signal

TX+ RS422 / TX+ C.L. = O - Transmit Data Positive: Main and auxiliary, RS 422 or Current Loop serial line, transmission positive bipolar signal

GND = - Ground signal

N.C. = - Not connected

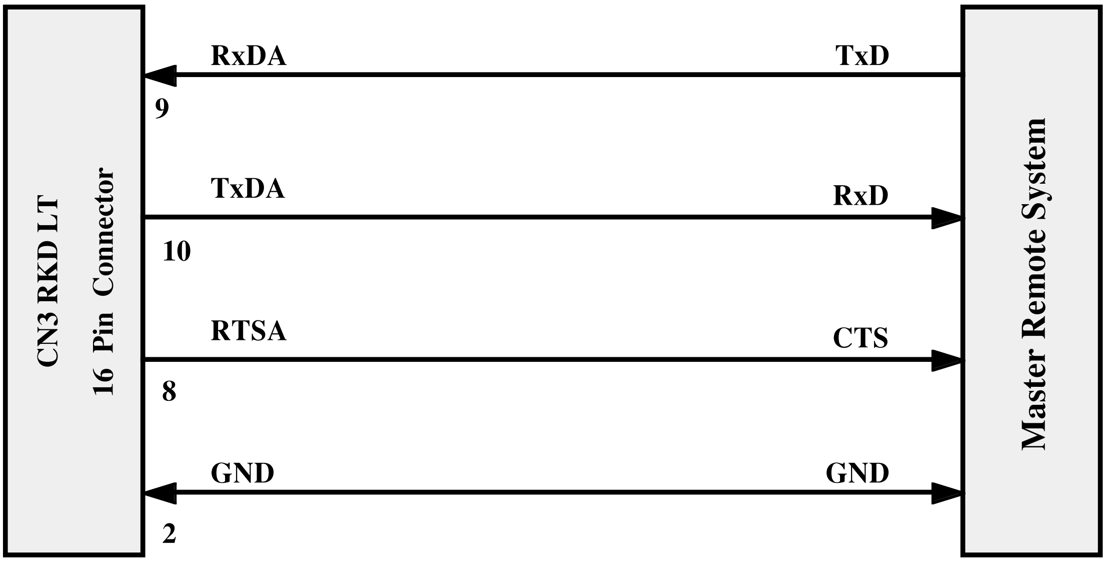


FIGURE 10: RS 232 EXAMPLE CONNECTION DIAGRAM

N.B.
In this diagram the reported RS 232 connection emploies the RTSA handshake signal. This signal is managed by the **RKD LT** board using the policy shown forward. However it is possible to communicate to the board without problrms also by a master contro lunit that doesn't support such signal.

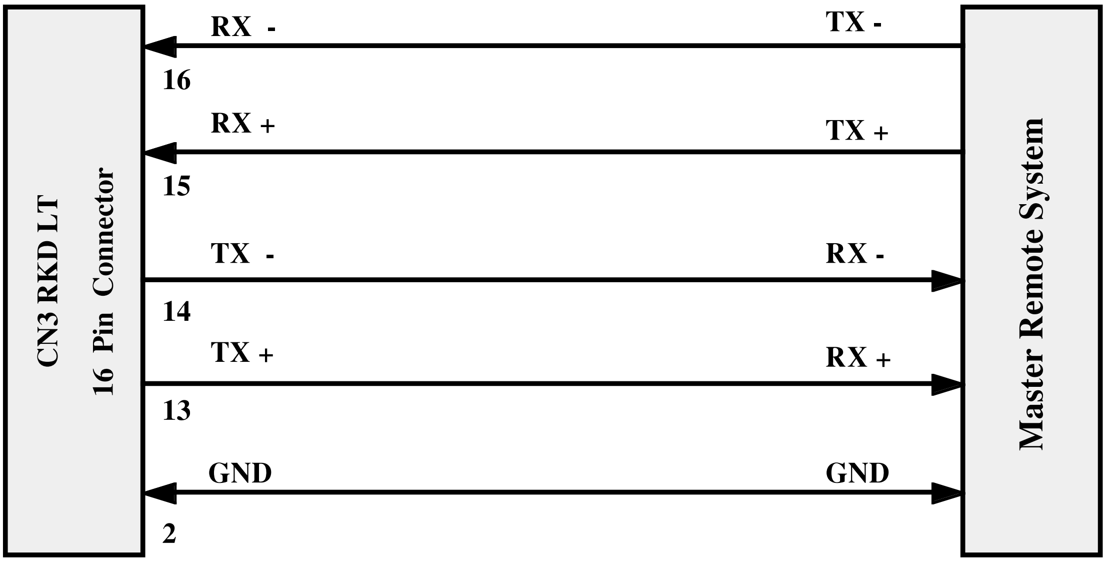


FIGURE 11: RS 422 EXAMPLE CONNECTION DIAGRAM

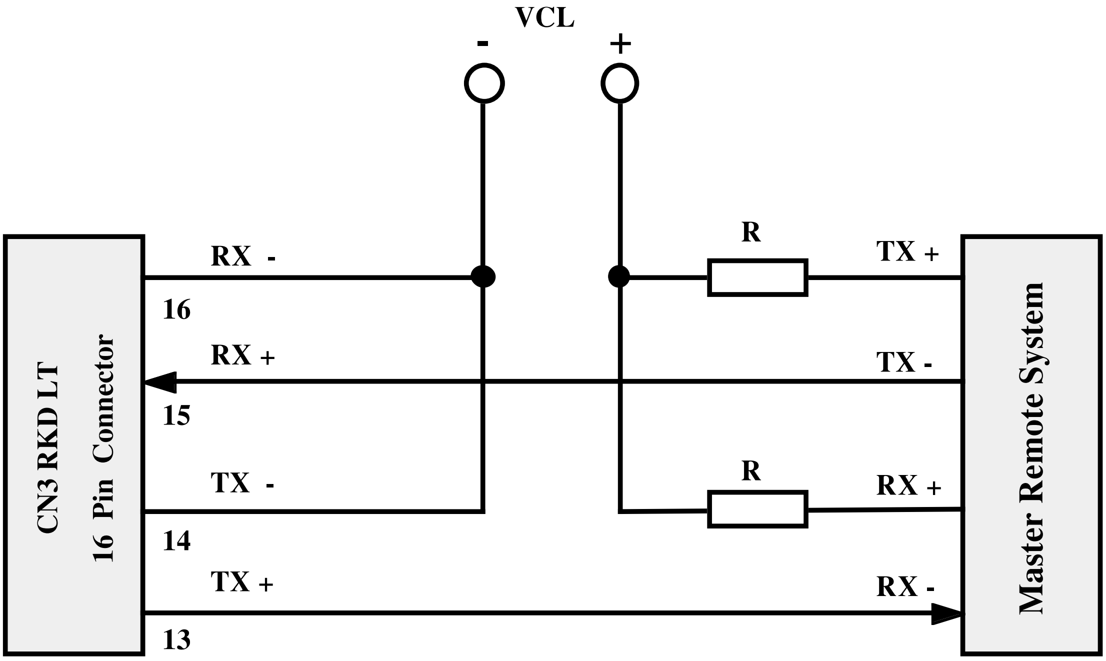


FIGURE 12: 4 WIRES CURRENT LOOP EXAMPLE CONNECTION DIAGRAM

N.B.
This diagram shows the voltage to supply the ring (**VCL**) and the current limitation resistors (**R**). The value of power supply voltage changes according to the number of connected devices, this because the maximun dissipated power for each device must be warranted to be at most **125 mW** for the transmitter and **90 mW** for the receiver when the maximun current allowed (**20 mA**) circulates.
The resisrtor **R** is needed to limit the current when the line is short-circuited; tipically for **VCL=5Vdc** the value of **R** is **220** Ω.
For further informations please refer to the HEWLETT-PACKARD Data-Book, in the sections about **HCPL 4100** and **HCPL 4200** Current Loop opto-couplers.

J5 - COMPLEMENTAR CONNECTOR FOR SERIAL LINES

The complementar connector for serial lines, called J5, is a 2.54 mm pitch, 16 pins,male, vertical, low profile connector. Its pin out is complementar respect to CN3 standard connector; this is made to allow a straight connection to all the control boards that follow this pin out (such as, for example, **GPC® 011**, **GPC®15A**, **GPC® 150**, **GPC® 188F**, etc.) by simply using a 16 pins flat cable.

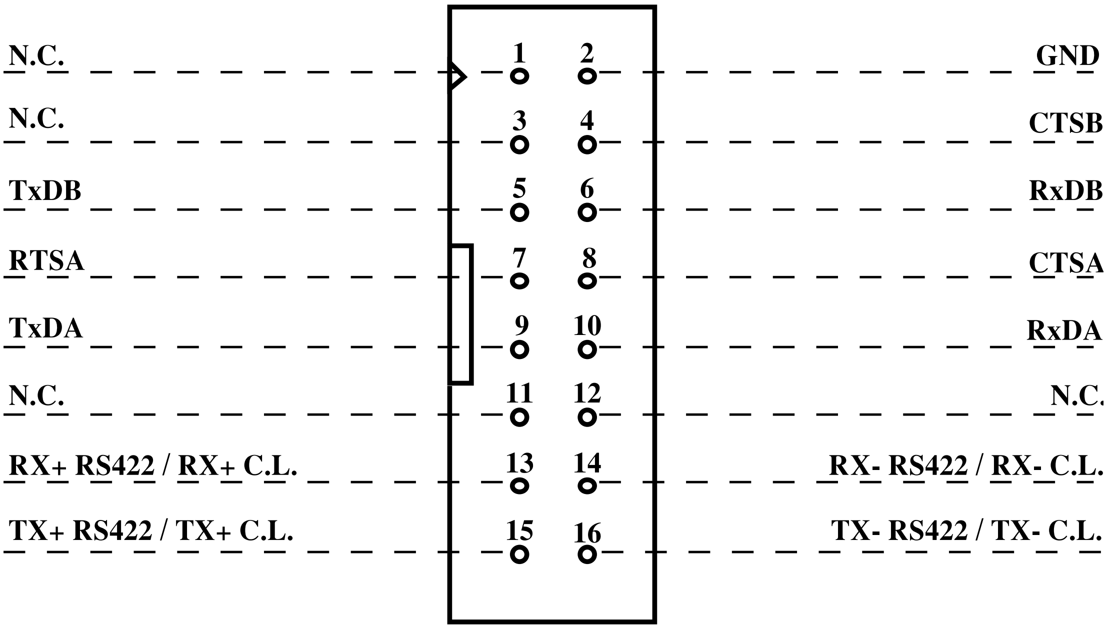


FIGURE 13: J5 - COMPLEMENTAR SERAIL LINES CONNECTOR

Symbols description:

TxDA - TxDB	= O -	Transmit Data: transmit data signal for main (**A**) and auxiliary (**B**) RS 232 serial line
RxDA - RxDB	= I -	Transmit Data: transmit data signal for main (**A**) and auxiliary (**B**) RS 232 serial line
CTSA - CTSB	= I -	Clear To Send: clear to send signal for main (**A**) and auxiliary (**B**) RS 232 serial line
RTSA	= O -	Request To Send: request to send signal for main (**A**) RS 232 serial line (not present on auxiliary serial line)
RX- RS422 / RX- C.L.	= I -	Receive Data Negative: Main and auxiliary, RS 422 or Current Loop serial line, reception negative bipolar signal
RX+ RS422 / RX+ C.L.	= I -	Receive Data Positive: Main and auxiliary, RS 422 or Current Loop serial line, reception positive bipolar signal
TX- RS422 / TX- C.L.	= O -	Transmit Data Negative: Main and auxiliary, RS 422 or Current Loop serial line, transmission negative bipolar signal
TX+ RS422 / TX+ C.L.	= O -	Transmit Data Positive: Main and auxiliary, RS 422 or Current Loop serial line, transmission positive bipolar signal
GND	= -	Ground signal
N.C.	= -	Not connected

J1 - CONNECTOR FOR FLUORESCENT ALPHNUMERIC DISPLAY

The connector for FUTABA alphanumeric fluorescent displays, called J1, is a 2.54 mm pitch, 20 pins,male, vertical, low profile connector. It allows a quick connection to the display by using a common 20 pins flat cable. Here follows the pin out.
Please remark that the power for the display is supplied directly by the **RKD LT** board.

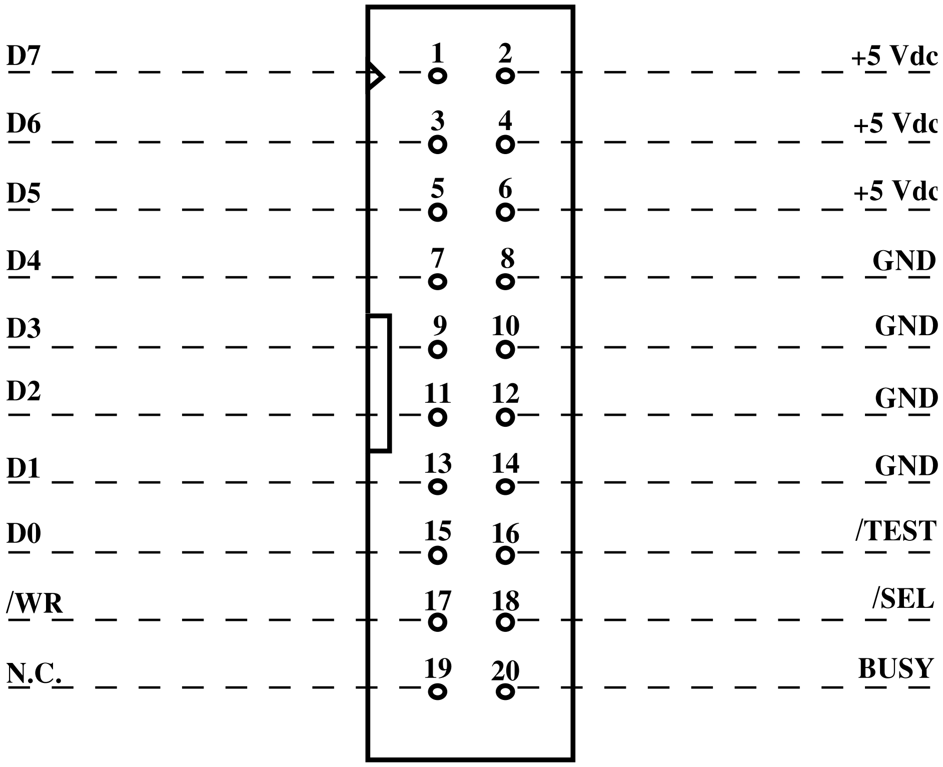


FIGURE 14: J1 - CONNECTOR FOR ALPHANUMERIC FLUORESCENT DISPLAY

Symbols description:

D0 - D7 = O - D0÷D7 data signals
/WR = O - Write signal
/RD = O - Read signal
/SEL = O - Device selection signal
/TEST = O - Test mode activation signal (must be always connected to +5 Vdc)
BUSY = I - Busy signal
+5 Vdc = O - Power supply for the display
GND = - Ground signal
N.C. = - Not Connected

N.B.
Connection between **RKD LT** and Futaba **M404SD01BA** (40x4 characters) must be performed using the proper adaptor, made by **grifo**®, called **IAF 404**.

J2 - CONNECTOR FOR GRAPHIC LCD DISPLAYS

The connector for LCD graphic displays, called J2, is a 2.54 mm pitch, 20 pins,male, vertical, low profile connector. It allows a quick connection to the display by using a common 20 pins flat cable, if present. Here follows the pin out.
Please remark that the all the supply voltages for the display are given directly by the **RKD LT** board.

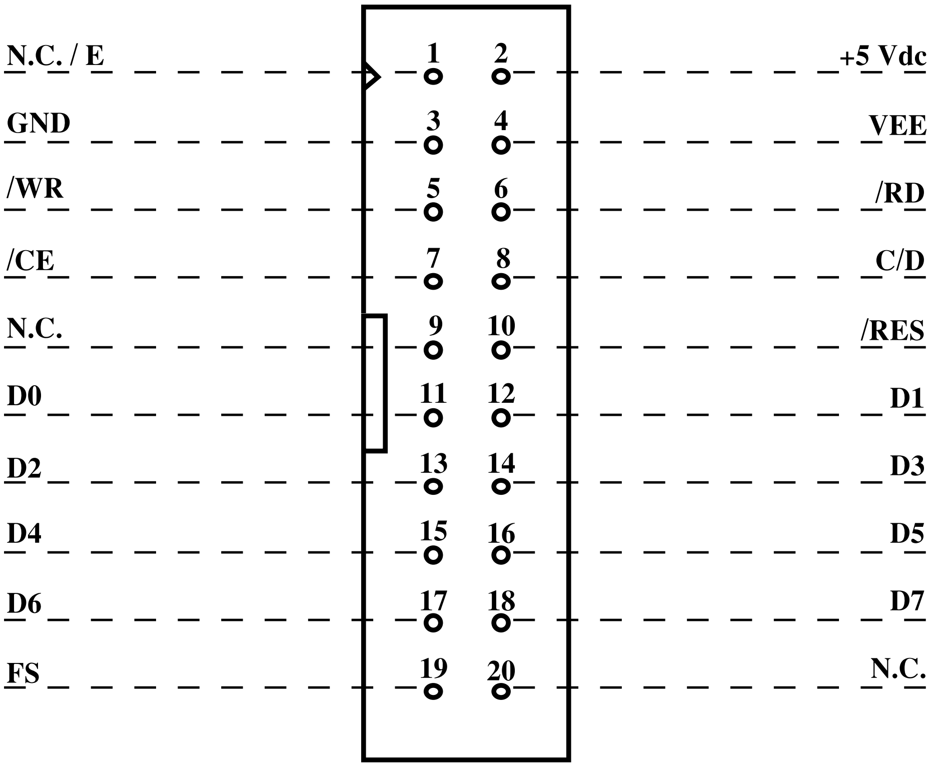


FIGURE 15: J2 - CONNECTOR FOR GRAPHIC LCD DISPLAY

Symbols description:

D0 - D7	= I/O	- D0÷D7 data signals
/WR	= O	- Write signal
/RD	= O	- Read signal
/CE	= O	- Device selection signal
/RES	= O	- Reset signal
C/D	= O	- Command/Data selection signal
E	= O	- Enable signal (only for displays mounting an Hitachi controller)
FS	= O	- Font selection in 6x8 or 8x8 alphanumeric mode (only for some displays)
VEE	= O	- Negative voltage for the display contrast
+5 Vdc	= O	- Power supply for the display
GND	=	- Ground signal
N.C.	=	- Not Connected

N.B.
RKD LT board must be configured according to the display controller used (Toshiba or Hitachi). This operation can be performed only by **grifo**®, so the controller type must be indicated in the order.

K1 - CONNECTOR FOR ABACO® BUS

The connector for **BUS ABACO®**, called J3, is aDIN 41612 64 pins M 90 degreeses A+C type C connector. Here follows the pin out for the oune installed on **RKD LT**, and the standard pin out of 8 and 16 bits **BUS ABACO®.**

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

FIGURE 16: J3 - CONNECTOR FOR ABACO® BUS

Symbols description:

8 bit CPU

A0-A15	=	O	- Address BUS
D0-D7	=	I/O	- Data BUS
INT	=	I	- Interrupt request
NMI	=	I	- Non Mascherable Interrupt
HALT	=	O	- Halt state
MREQ	=	O	- Memory Request
IORQ	=	O	- Input Output Request
RD	=	O	- Read cycle status
WR	=	O	- Write cycle status
BUSAK	=	O	- BUS Acknowledge
WAIT	=	I	- Wait
BUSRQ	=	I	- BUS Request
RESET	=	O	- Reset
M1	=	O	- Machine cycle one
RFSH	=	O	- Refresh (for dynamic memories)
MEMDIS	=	I	- Memory Display (signal output by a memory mapped peripherial device)
VDUSEL	=	O	- VDU Selection (abilitation to be memory mapped for the peripherial device)
IEI	=	I	- Interrupt Enable Input
CLK	=	O	- Clock:
R.T.	=	I	- Reset
+5 Vdc	=	I	- +5 Vdc power supply
+12 Vdc	=	I	- +12 Vdc power supply
-12 Vdc	=	I	- -12 Vdc power supply
GND	=		- Ground for all the signals of the BUS

16 bit CPU

A0-A22	=	O	- Address BUS
D0-D15	=	I/O	- Data BUS
RD UDS	=	O	- Read Upper Data Strobe
WR UDS	=	O	- Write Upper Data Strobe
IACK	=	O	- Interrupt Acknowledge
RD LDS	=	O	- Read Lower Data Strobe
WR LDS	=	O	- Write Lower Data Strobe

N.B.
The above mentioned indications of directionality are relative to a **GPC®** command board. These have been kept untouched to avoid ambiguities or misinterpretationa in case of multi-boards systems.

JUMPERS

On **RKD LT** there are 11 jumpers for card configuration. Connecting these jumpers, the User can define for example the peripheral devices functionality, the serial communication interface and so on. To easily locate the jumpers please refer to figure 19. Here below is the jumpers list, location and function:

JUMPER	PIN N.	USE
JP1	3	RESERVED
JP2	5	Enables the circuitry for RS 422 serial communication
JP3	3	In conjunction with JP4, JP5 e JP10, selects the type of communication for the two serial lines
JP4	3	In conjunction with JP3, JP5 e JP10, selects the type of communication for the two serial lines
JP5	3	In conjunction with a JP3, JP4 e JP10, selects the type of communication for the two serial lines
JP6	3	Enables the circuitry for RS 422 serial communication
JP7	2	It connects the termination resistor to the RS 422 reception line
JP8	2	It connects the BUS **ABACO®** /M1 signal
JP10	3	In conjunction with JP3, JP4 e JP5, selects the type of communication for the two serial lines
JP11	2	It selects the voltage range for the LCD contrast
S3.1	2	It connects the BUS **ABACO®** /RESET signa'

FIGURE 17: JUMPERS SUMMARIZING TABLE

The "*" denotes the default connection, or on the other hand the connection set up at the end of testing phase, that is the configuration the User receives.

5 PINS JUMPERS

JUMPER	CONNECTION	USE	DEF.
JP2	position 2-3	Enables the RS 422 communication circuitery	
	not connected	Disables the RS 422 communication circuitery	*

FIGURE 18: 5 PINS JUMPERS TABLE

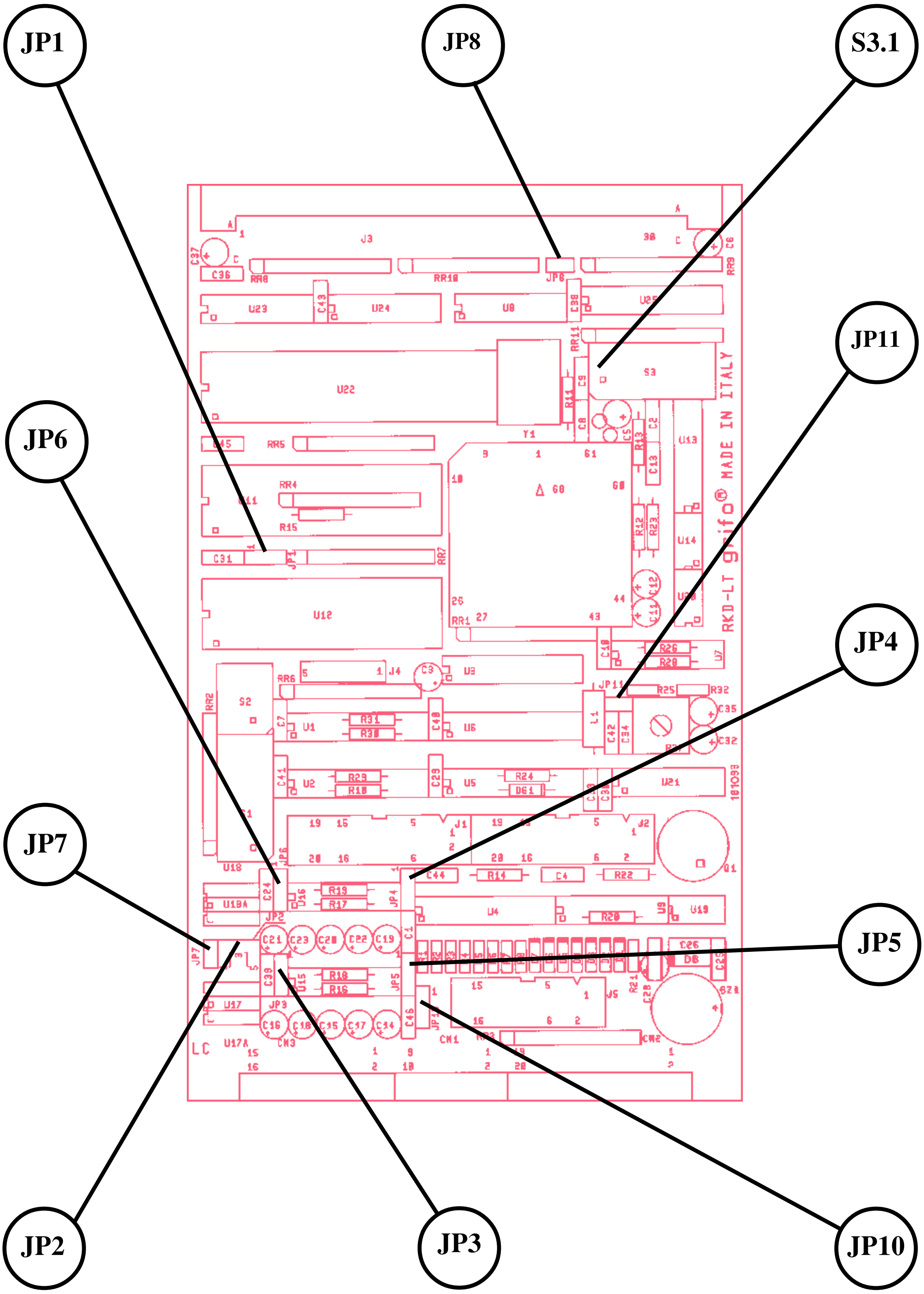


FIGURE 19: JUMPERS LOCATION

3 PINS JUMPERS

JUMPER	CONNECTION	USE	DEF.
JP3	position 1-2	In conjunction with JP4, JP5 and JP10, it configures the main serial line (A) for RS 422 or Current Loop and the auxiliary serial line (B) for RS 232	
	position 2-3	In conjunction with JP4, JP5 and JP10, it configures the main serial line (A) for RS 232 and the auxiliary serial line (B) for RS 422 or Current Loop	
	not connected	In conjunction with JP4, JP5 and JP10, it configures both serial lines for RS 232	*
JP4	position 1-2	In conjunction with JP3, JP5 and JP10, it configures the main serial line (A) for RS 422 or Current Loop and the auxiliary serial line (B) for RS 232	
	position 2-3	In conjunction with JP3, JP5 and JP10, it configures the main serial line (A) for RS 232 and the auxiliary serial line (B) for RS 422 or Current Loop	
	not connected	In conjunction with JP3, JP5 and JP10, it configures both serial lines for RS 232	*
JP5	position 1-2	In conjunction with JP3, JP4 and JP10, it configures the main serial line (A) for RS 422 or Current Loop and the auxiliary serial line (B) for RS 232	
	position 2-3	In conjunction with JP3, JP4 and JP10, it configures the main serial line (A) for RS 232 and the auxiliary serial line (B) for RS 422 or Current Loop	
	not connected	In conjunction with JP3, JP4 and JP10, it configures both serial lines for RS 232	*
JP6	position 2-3	Enables the RS 422 communication circuitery	
	not connected	Disables the RS 422 communication circuitery	*
JP10	position 1-2	In conjunction with JP3, JP4 and JP5, it configures the main serial line (A) for RS 422 or Current Loop and the auxiliary serial line (B) for RS 232	
	position 2-3	In conjunction with JP3, JP4 and JP5, it configures the main serial line (A) for RS 232 and the auxiliary serial line (B) for RS 422 or Current Loop	
	not connected	In conjunction with JP3, JP4 and JP5, it configures both serial lines for RS 232	*

FIGURE 20: 3 PINS JUMPERS TABLE

N.B.
For a correct use of the serial communication section, jumpers JP3, JP4, JP5 and JP10 must be always connected in the same positions; other combinations are meaningless.

2 PINS JUMPERS

JUMPER	CONNECTION	USE	DEF.
JP7	not connected connesso	It doesn't connect the termination resistor to the RS 422 communication circuitry It connects the termination resistor to the RS 422 communication circuitry	*
JP8	not connected connesso	Parallel interface doesn't manage the BUS **ABACO®** /M1 signal Parallel interface manages the BUS **ABACO®** /M1 signal	*
JP11	not connected connesso	It selects the **-7.5 ÷ -15 Vdc voltage range** for the LCD contrast regolation It selects the **-6 ÷ -12 Vdc voltage range** for the LCD contrast regolation	*
S3.1	OFF ON	It doesn't connect the BUS **ABACO®** /RESET signal to its proper on board circuitery It connects the BUS **ABACO®** /RESET signal to its proper on board circuitery	 *

FIGURE 21: 2 PINS JUMPERS TABLE

GRAPHIC LCD CONTRAST REGULATION

RKD LT mounts a circuitery capable to give out the negative voltage needed to regulate the contrast of the LCD graphic displays.
According to the needs of the display being used, it is possible to select, by acting on JP11 (please see previous tebles), a **-7.5 ÷ -15 Vdc** or a **-6 ÷ -12 Vdc** voltage range.
The regulation of this voltage inside its range and so the regulation of the contrast intensity is made by acting on a trimmer called **R27**.
To easily locate this component please refer to figure 8.
Optimal contrast regulation betters the display visibility also when environmental light is variable.

SERIAL COMMUNICATION

The communication lines of **RKD LT** board can be buffered as RS 232, RS 422 or Current Loop. The selection of which communicaton mode to employ is made by acting on proper jumpers, whose location and connection has been shown and expalined in the previous paragraphs.
Here follows the description of the possible configurations; please remark that jumpers not mentioned have no influence for the communication whatever their connection is.

- MAIN SERIAL LINE (**A**) AS **RS 422**, AUXILIARY SERIAL LINE (**B**) AS **RS 232**
  MAX 202 serial driver must be installed on U15, driver SN 75176 must be installed on U17 and U18 while no driver must be installed on U17A and U18A. No jumper must be connected.

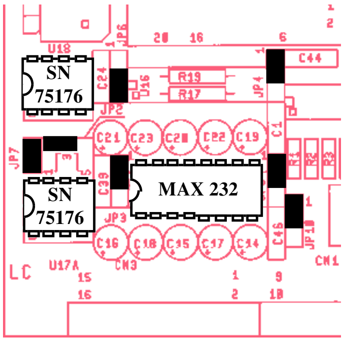


FIGURE 22: MAIN SERIAL LINE AS RS 422, AUXILIARY SERIAL LINE RS 232

- MAIN SERIAL LINE (**A**) AS **CURRENT LOOP**, AUXILIARY SERIAL LINE (**B**) AS **RS 232**
  MAX202 serial driver must be installed on U15, driver HCPL4200 must be installed on U17A, driver HCPL4100 must be installed on U18A while no driver must be installed on U16, U17 and U18. Jumpers JP3, JP4, JP5 and JP10 must be connected in position 1-2, while the remaining jumpers must be disconnected.

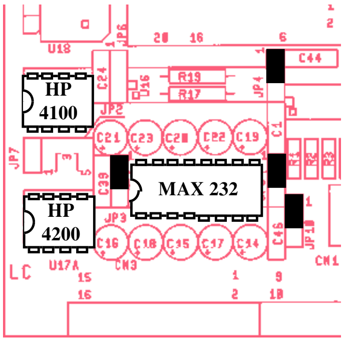


FIGURE 23: MAIN SERIAL LINE AS CURRENT LOOP, AUXILIARY SERIAL LINE RS 232

- MAIN SERIAL LINE (**A**) AS **RS 232**, AUXILIARY SERIAL LINE (**B**) AS **RS 422**
  MAX 202 serial driver must be installed on U16, driver SN 75176 must be installed on U17 and U18 while no driver must be installed on U17A and U18A.
  Jumper JP2, JP3, JP4, JP5, JP6 and JP10 must be connected in position 2-3, jumper JP7 must be connected.

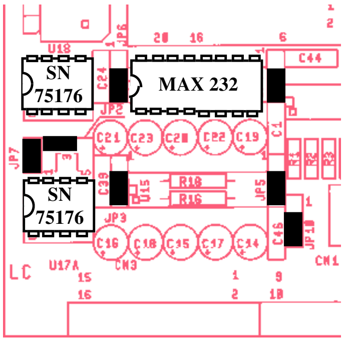


FIGURE 24: MAIN SERIAL LINE AS RS 232, AUXILIARY SERIAL LINE RS 232

- MAIN SERIAL LINE (**A**) AS **RS 232**, AUXILIARY SERIAL LINE (**B**) AS **CURRENT LOOP**
  MAX202 serial driver must be installed on U16, driver HCPL4200 must be installed on U17A, driver HCPL4100 must be installed on U18A while no driver must be installed on U16, U17 and U18. Jumpers JP3, JP4, JP5 and JP10 must be connected in position 2-3, while the remaining jumpers must be disconnected.

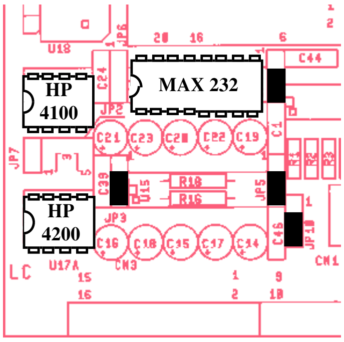


FIGURE 25: MAIN SERIAL LINE AS RS 232, AUXILIARY SERIAL LINE CURRENT LOOP

- MAIN SERIAL LINE (**A**) AS **RS 232**, AUXILIARY SERIAL LINE (**B**) AS **RS 422**
  MAX 202 serial driver must be installed on U15 and U16, while no driver must be installed on U17, U18, U17A and U18A. No jumper must be connected.

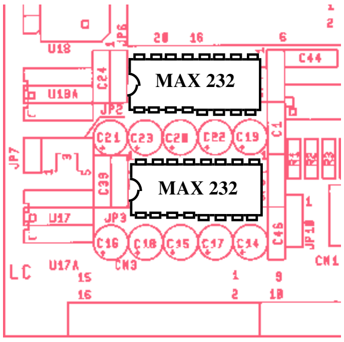


FIGURE 26: MAIN SERIAL LINE AS RS 232, AUXILIARY SERIAL LINE RS 232

PARALLEL INTERFACE DESCRIPTION

INTRODUCTION

This chapter contains an exhaustive hardware and software informations about how to use the parallel interface installed on **RKD LT** board. Part of these informations consist of the occupation of **ABACO®** BUS I/O mapping by the **RKD LT** board and the communication modalities to the master control unit.
Of course, if the **RKD LT** board is configured to communicate by the serial line, the informations contained in this chapter are needless.

BOARD MAPPING

The **RKD LT** board takes only two consecutive bytes in the I/O addressing space, these bytes can be allocated from a different base address according to how the board is mapped. This feature allows to install more than one **RKD LT** board across the same BUS **ABACO®**, or to install the board across a bus where other boards are present, obtaining a structure easy to expand without any modification to the software already written.
The BUS interface circuitry is installed on the board itself, to select the base address it uses an 8 pins Dip Switch, called S3, where the User can directly set the base address.
Here follows the correspondance between the Dip Switch disposition and the selected address space management mode:

S3.1	->	*See paragraph "JUMPER"*
S3.2	->	A1 address signal
S3.3	->	A2 address signal
S3.4	->	A3 address signal
S3.5	->	A4 address signal
S3.6	->	A5 address signal
S3.7	->	A6 address signal
S3.8	->	A7 address signal

The Dip Switch works in complemented logic, this means that an **ON** position generates a **logic zero**, while on **OFF** position generates a **logic one**.
Also jumper JP8, described in the previous chapter, influences the addressing logic and must be set according to the type of control board (**GPC®** serie) used. In detail, if the control board is provided with /M1 signal on the BUS **ABACO®** connector then JP8 must be connected and viceversa.

Please be careful not to install more than one board inside the same addressing space, considering also the amount of bytes occupied by every board in the addressing space.
In case this condition should not be respected a BUS conflict happens, compromising the regular work of the whole system.

Here follows an example.
If the User wants to map the **RKD LT** board on address 04AH in a BUS where also a control board provided with /M1 signal works, the configuration must be:

JP8	->	Connected
S3.2	->	OFF
S3.3	->	ON
S3.4	->	OFF
S3.5	->	ON
S3.6	->	ON
S3.7	->	OFF
S3.8	->	ON

To locate on the board the devices here mentioned, please refer to figure 8 and 19, printed in the previous pages.

PARALLEL COMMUNICATION REGISTERS

The parallel communication registers of **RKD LT** board are mapped at the below mentioned addresses, having indicated with <base add> the base address for the board mapping, that is the address set by S3 as explained in the previous paragraph.

REGISTER	ADDRESS	R/W	DESCRIPTION
STATUS	<base add>+00	R	Status register of the parallel interface
DATA	<base add>+01	R/W	Data register of the parallel interface

FIGURE 27: PARALLEL COMMUNICATION ADDRESSES TABLE

PARALLEL COMMUNICATION MANAGEMENT

This paragraph explains the sofware management modalities of the parallel communication between **RKD LT** board and a master control unit.
The read/write DATA register, described in the previous table, allows to receive from and to transmit informations to the board, while the read only STATUS registers must be interpreted as follows:

	bit7	*bit6*	*bit5*	*bit4*	*bit3*	*bit2*	*bit1*	*bit0*
STATUS =	**OBF**	**IBF**	NU	NU	NU	NU	NU	NU

where:

NU = Not Used

IBF = If active (**1**) indicates that the **RKD LT** board is ready to receive a new data; the master control unit can write it to the DATA register.

OBF = If active (**1**) indicates that the **RKD LT** board has written a data into its transmission buffer; the master control unit can read the data performing a read operation of the DATA register.

After a Reset or a Power-On the master control unit must verify that the **RKD LT** board has completed its initialization phase and is ready to receive data.
This condition is indicated by the logic state of the flags IBF and OBF, which must show the values: **IBF=1** and **OBF=0**.
The program running on the control board must start by performing a cycle (eventually timed out for more safety) to test the status of these signals and wait for the **RKD LT** board to be ready for the commmunication.

To show the above described modalities, here follow two examples subroutines written in CBZ 80:

```
"sendtorkd"
REM Transmits to RKD LT the value contained in the variable var%
REM Begin
        DO          : REM Waiting for bit IBF to activate
            ibf%=INP(status%)
        UNTIL ((ibf% AND &040)=&040)
        OUT data%, var%
REM End
RETURN
```

```
"recfromrkd"
REM Tests whether RKD LT has sent a char; in such case returns it in the variable var%,
REM Otherwise returns -1
REM Begin
        obf%=INP(status%)   : REM Tests bit OBF
        IF ((obf% AND &080)=&080) THEN var%=INP(data%) ELSE var%=-1
REM End
RETURN
```

SOFTWARE DESCRIPTION

The **RKD LT** board is a full functional video terminal; so whatever is received, exept the command sequences, is diplayed and the codes of the keys pressed on the BG Keyboard are sent to the master control unit.

To properly use the board, the User must first configure it by selecting: the kind of display used, the communication protocol, etc. This operation can be performed simply by acting on the configuration Dip Switches, as will be explained in the following pages.

In addtion to the description of the many commands recognized by the board, there is a complete description of the command sequences accepted by the on board firmware.

Every code or code sequence is reported with a double description: the mnemonic one, employing ASCII characters, and the numeric one, expressed both in decimal and hexadecimal format.

BOARD CONFIGURATION

To configure the **RKD LT** board the User must act on the **S1** and **S2** Dip Switches, to easily locate these components on the board please refer to figure 8.

Here follow the configuration modalities, accepted by the **TRKDGL** and **TRKDAF** firmware. Please remark that the below described settings must be performed when the board is not supplied, because the acquisition of the Dip Switch configuration is made only after a Reset or a Power-On.

CONFIGURATION USING TRKDGL FIRMWARE

The configuration using the LCD graphic display manager **TRKDGL** firmware must be performed as described in the following table.

N. DIP	FUNCTION	ON	OFF
S2.1	NOT USED	---	---
S2.2 S2.3	Select the BAUD RATE of the main serial line	*See next table*	
S2.4	Selects the communication interface	**ABACO®** BUS	Main serial line
S1.1 S1.2 S1.3 S1.4	Select the display model	*See next table*	
S1.5	Selects the LCD display working mode	Graphic	Alphanumeric
S1.7	Select the BAUD RATE of the auxiliary serial line	*See next table*	
S1.8	Selects the standard of the commands	ADDS VIEWPOINT	TVI950

FIGURE 28: TRKDGL DIP SWITCH CONFIGURATION TABLE

Display model selection:

The selection of the LCD graphic display is performed by the first 4 dips of S1, as described in the following table.

S1.1	S1.2	S1.3	S1.4	DISPLAY (format - controller)
OFF	OFF	OFF	OFF	TLX-711A and compatibles with char matrix **6x8** (240 x 64 pixels - TOSHIBA T6963C)
ON	OFF	OFF	OFF	TLX-711A and compatibles with char matrix **8x8** (240 x 64 pixels - TOSHIBA T6963C)
OFF	OFF	ON	OFF	TLX-1391 and compatibles with char matrix **6x8** (128 x 128 pixels - TOSHIBA T6963C)
ON	OFF	ON	OFF	TLX-1391 and compatibles with char matrix **8x8** (128 x 128 pixels - TOSHIBA T6963C)
OFF	ON	OFF	OFF	TLX-1021 and compatibles (120 x 64 pixels - TOSHIBA T6963C)
ON	ON	OFF	OFF	TLX-1013 or compatibles (160 x 128 pixels - TOSHIBA T6963C)
OFF	ON	ON	OFF	TLX-1301 and compatibles with char matrix **6x8** (240 x 128 pixels - TOSHIBA T6963C)
ON	ON	ON	OFF	TLX-1301 and compatibles with char matrix **8x8** (240 x 128 pixels - TOSHIBA T6963C)
OFF	OFF	OFF	ON	TLC-1091 and compatibles (240 x 128 pixels - TOSHIBA T6963C)
OFF	OFF	ON	ON	MGLS240128V2 and compatibles (240 x 128 pixels - HITACHI HD61830B)

FIGURE 29: GRAPHIC LCD MODEL SELECTION TABLE

N. B.:
The character matrix format is managed only in alphanumeric mode, when graphic mode is selected (dip S1.5 ON) the minimum character size is 8x8 pixels and the S1 dips must be configured opportunely.
The selection of 6x8 character format in graphic mode is not acceptable and causes visualization problems.

Main serial line Baud Rate selection:

If the parallel communication mode is selected (dip S2.4 ON) the configuration of dips S1.2 and S1.3 has no influence to the Baud Rate selection.
Otherwise, to set this parameter the User must act on S2 as follows.

S2.2	S2.3	BAUD RATE
OFF	OFF	19200 Baud
ON	OFF	9600 Baud
OFF	ON	4800 Baud
ON	ON	2400 Baud

FIGURE 30: TRKDGL MAIN SERIAL LINE BAUDE RATE SELECTION TABLE

Auxiliary serial line Baud Rate selection:

The auxiliary serial line Baud Rate selection must be performed by acting on dips S1.6 and S1.7, according to the folowing table.

S1.6	S1.7	BAUD RATE
OFF	OFF	19200 Baud
ON	OFF	9600 Baud
OFF	ON	4800 Baud
ON	ON	2400 Baud

FIGURE 31: TRKDGL AUXILIARY SERIAL LINE BAUDE RATE SELECTION TABLE

CONFIGURATION USING TRKDAF FIRMWARE

The **TRKDAF** firmware is used to manage the Futaba fluorescent alphanumeric displays. To perform the configuration of **RKD LT** board using this firmware please see the following table.

N. DIP	FUNCTION	ON	OFF
S2.1	NOT USED	---	---
S2.2 S2.3	Select the BAUD RATE of the main serial line	*See next table*	
S2.4	Selects the communication interface	**ABACO®** BUS	Main serial line
S1.1 S1.2 S1.3	Select the display model	*See next table*	
S1.4	Selects the standard of the commands	ADDS VIEWPOINT	TVI950
S1.5	NOT USED	---	---
S1.6 S1.7 S1.8	Select the BAUD RATE of the auxiliary serial line	*See next table*	

FIGURE 32: TRKDAF DIP SWITCH CONFIGURATION TABLE

Display model selection:

The selection of the FUTABA fluorescent display is performed by the first 3 dips of S1, as described in the following table.

S1.1	S1.2	S1.3	DISPLAY (format)
OFF	OFF	OFF	Futaba M202SD01BA and compatibles (20x2 characters "small")
OFF	OFF	ON	Futaba M202SD08GK and compatibles (20x2 characters "big")
OFF	ON	OFF	Futaba M204SD01AA and compatibles (20x4 characters)
OFF	ON	ON	Futaba M40SD04GJ and compatibles (40x1 characters)
ON	OFF	OFF	Futaba M402SD07GK and compatibles (40x2 characters)
ON	OFF	ON	Futaba M404SD01BA and compatibles (40x4 characters)

FIGURE 33: FUTABA FLUORESCENT DISPLEY SELECTION TABLE

Main serial line Baud Rate selection:

If the parallel communication mode is selected (dip S2.4 ON) the configuration of dips S1.2 and S1.3 has no influence to the Baud Rate selection.
Otherwise, to set this parameter the User must act on S2 as follows.

S2.2	S2.3	BAUD RATE
OFF	OFF	19200 Baud
ON	OFF	9600 Baud
OFF	ON	4800 Baud
ON	ON	2400 Baud

FIGURE 34: TRKDAF MAIN SERIAL LINE BAUDE RATE SELECTION TABLE

Auxiliary serial line Baud Rate selection:

The auxiliary serial line Baud Rate selection must be performed by acting on dips S1.6, S1.7 and S1.8, according to the folowing table.

S1.6	S1.7	S1.8	BAUD RATE
OFF	OFF	OFF	19200 Baud
ON	OFF	OFF	9600 Baud
OFF	ON	OFF	4800 Baud
ON	ON	OFF	2400 Baud
OFF	OFF	ON	1200 Baud
ON	OFF	ON	600 Baud
OFF	ON	ON	300 Baud

FIGURE 35: TRKDAF AUXILIARY SERIAL LINE BAUDE RATE SELECTION TABLE

RECEPTION BUFFER

RKD LT board is provided with a reception buffer to make it faster during the communication to the master control unit and reduce its stand by times. Every byte received, both in parallel and in serial communcation mode, is immediatly saved into this buffer (sized **3200 bytes**), then processed at the end of the operation in progress.
Of course, in the case of continuous transmission of command sequences that require a long execution time (such graphic mode commands, EEPROM management commands, etc.), the buffer is going to overflow. This is why the **RKD LT** board advices the other boards of the imminence of this event as hereunder described:

RS 232 serial communication	->	**/RTS** deactived (-12 Vdc)
Parallel communication	->	**IBF** deactived (logic level **0**)

So the master control unit will have to manage opportunely these signals, suspending the transmission up to when the buffer of the **RKD LT** board will be empty again, and ready to receive new data. Please remark that, in case of **RS 422** or **Current Loop** serial communication, this feature is not available, because the **/RTS** signal is not present. The User will have to insert delays in the communication, long enough to allow the **RKD LT** board to perform the operations requested without filling completely the reception buffer.

KEYBOARD ACQUISITION

RKD LT board manages directly the 56 keys BG keyboard, or a generic 7x8 matrix keyboard. When the board detects a key hit transmits its code to the master control unit, according the figures reported in the next table.
There is also an AutoRepeat feature, in fact when the **RKD LT** board detects that the pressure of a key lasts longer than **0.8 seconds**, it will begin to send repeteadly its code every about **0.2 seconds**, up to when that key will be released.
In the following table the codes of the "special" keys (ENTER, arrows, etc.) are reported; it is remarkabe that some of these codes depend on how the **RKD LT** board has been configured, that is to work under TVI 950 or ADDS Viewpoint terminal emulation.

KEY	ADDS VIEWPOINT		TVI 950	
	Dec.	Hex	Dec.	Hex
UP Arrow "red"	26	1A	11	0B
DOWN Arrow "red"	10	0A	10	0A
LEFT Arrow "red"	21	15	08	08
RIGHT Arrow "red"	06	06	09	09
UP Arrow "cyan"	133	85	133	85
DOWN Arrow "cyan"	134	86	134	86
STOP / RUN	130	82	130	82
DEL	127	7F	127	7F
ERASE	132	84	132	84
ENTER	13	0D	13	0D
Spacebar	32	20	32	20

FIGURE 36: SPECIAL KEYS CODE TABLE

All the remainig keys are managed by the Shift, Caps Lock and Control functions, that can be obtained holding respectively the **SHIFT**, **DRAW** and **CTRL** keys.
In the following table the decimal, hexadecimal and ASCII codes of these keys in the many possible combinations, are reported.

KEY	**NORMAL**			**+ SHIFT**			**+ DRAW (Caps lock)**			**+ DRAW + SHIFT**		
	Dec.	**Hex**	**ASCII**	**Dec.**	**Hex**	**ASCII**	**Dec.**	**Hex**	**ASCII**	**Dec.**	**Hex**	**ASCII**
1 !	49	31	1	33	21	!	49	31	1	33	21	!
2 "	50	32	2	34	22	"	50	32	2	34	22	"
3 #	51	33	3	35	23	#	51	33	3	35	23	#
4 $	52	34	4	36	24	$	52	34	4	36	24	$
5 %	53	35	5	37	25	%	53	35	5	37	25	%
6 &	54	36	6	38	26	&	54	36	6	38	26	&
7 '	55	37	7	39	27	'	55	37	7	39	27	'
8 (	56	38	8	40	28	(	56	38	8	40	28	(
9)	57	39	9	41	29	)	57	39	9	41	29	)
0 @	48	30	0	64	40	@	48	30	0	64	40	@
: *	58	3A	:	42	2A	*	58	3A	:	42	2A	*
- =	45	2D	-	61	3D	=	45	2D	-	61	3D	=
; +	59	3B	;	43	2B	+	59	3B	;	43	2B	+
, <	44	2C	,	60	3C	<	44	2C	,	60	3C	<
. >	46	2E	.	62	3E	>	46	2E	.	62	3E	>
/ ?	47	2F	/	63	3F	?	47	2F	/	63	3F	?
A	97	61	a	65	41	A	65	41	A	97	61	a
B	98	62	b	66	42	B	66	42	B	98	62	b
C	99	63	c	67	43	C	67	43	C	99	63	c
D	100	64	d	68	44	D	68	44	D	100	64	d
E	101	65	e	69	45	E	69	45	E	101	65	e
F	102	66	f	70	46	F	70	46	F	102	66	f
G	103	67	g	71	47	G	71	47	G	103	67	g
H	104	68	h	72	48	H	72	48	H	104	68	h
I	105	69	i	73	49	I	73	49	I	105	69	i
J	106	6A	j	74	4A	J	74	4A	J	106	6A	j
K	107	6B	k	75	4B	K	75	4B	K	107	6B	k
L	108	6C	l	76	4C	L	76	4C	L	108	6C	l
M	109	6D	m	77	4D	M	77	4D	M	109	6D	m
N	110	6E	n	78	4E	N	78	4E	N	110	6E	n
O	111	6F	o	79	4F	O	79	4F	O	111	6F	o
P	112	70	p	80	50	P	80	50	P	112	70	p
Q	113	71	q	81	51	Q	81	51	Q	113	71	q
R	114	72	r	82	52	R	82	52	R	114	72	r
S	115	73	s	83	53	S	83	53	S	115	73	s
T	116	74	t	84	54	T	84	54	T	116	74	t
U	117	75	u	85	55	U	85	55	U	117	75	u
V	118	76	v	86	56	V	86	56	V	118	76	v
W	119	77	w	87	57	W	87	57	W	119	77	w
X	120	78	x	88	58	X	88	58	X	120	78	x
Y	121	79	y	89	59	Y	89	59	Y	121	79	y
Z	122	7A	z	90	5A	Z	90	5A	Z	122	7A	z

FIGURE 37: ALPHANUMERIC KEYS CODE TABLE

In this table the combination obtainable pressing the **CTRL** key are not reported because the codes are very easy to calculate by simply subtracting **64** (**40** Hex).
For example the combination **CTRL+SHIFT+A** generates the code: 65-64=**1**.
Also, we would remark that the Control function is available only for keys, or keys combinations, whose code is equal to or greater than 64 (@, A, B, C, ...).

To locate the keys in the 7x8 matrix keyboard, please refer to the previously reported **BG keyboard** electric diagram. In that picture also a LED and abuzzer are shown. **RKD LT** board manages them using the following modalities:

LED	->	Indicates the Caps Lock mode status: *ON = capitol letters*
Buzzer	->	Used for the feedback when a key is hit (*KeyClick*), when this feature is enabled.

CHARACTERS REPRESENTATION

RKD LT board running **TRKDAF** or **TRKDGL** in alphanumeric mode visualizes directly on the connected display all the characters having a code included in the range **32÷255** (**20÷FF Hex**); while when running **TRKDGL** in graphic mode this range reduces to **32÷126**(**20÷7E Hex**).
Characters having code out of these ranges, if not commands, are ignored.
The characters having code included in the range **32÷127**(**20÷7F Hex**) are ASCII standard, while the ones having code in the range **128÷255** (**80÷FF Hex**), have different effects according to the model of display used. For further informations please refer to the tables reported in the specific documentation.
The character will be displayed in the current cursor position, the cursor will advance to the next position, if it was in the last position (bottom right corner) it will be placed in the Home position (top left corner).

CURSOR POSITIONING COMMANDS

Here follows a list of the several cursor positioning commands available according to the installed firmware.

CURSOR LEFT

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code:	***08***
Mnemonic:	***BS***

ADDS Viewpoint

Code:	***21 (15 Hex)***
Mnemonic:	***NACK***

The cursor is moved one position to the left without altering the content of the display.
If the cursor is in the Home position, it will be placed in the last position (bottom right) of the display.

CURSOR RIGHT

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: ***09***
Mnemonic: ***HT***

ADDS Viewpoint

Code: ***06***
Mnemonic: ***ACK***

The cursor is moved one position to the right without altering the content of the display. If the cursor is in the last position (bottom right) of the display, it will be placed in the Home position.

CURSOR DOWN

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 and ADDS Viewpoint

Code: ***10 (0A Hex)***
Mnemonic: ***LF***

The cursor is moved to the row under the one where if finds, remaining in the same column. If the cursor finds in the last row, it will be moved to the first row.

CURSOR UP

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: ***11 (0B Hex)***
Mnemonic: ***VT***

ADDS Viewpoint

Code: ***26 (1A Hex)***
Mnemonic: ***SUB***

The cursor is moved to the row above the one where if finds, remaining in the same column. If the cursor finds in the first row, it will be moved to the last row.

CARRIAGE RETURN

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 and ADDS Viewpoint

Code: ***13 (0D Hex)***
Mnemonic: ***CR***

The cursor is moved to the beginning of the row where it finds.

CARRIAGE RETURN + LINE FEED

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950 and ADDS Viewpoint

Code: *29* *(1D Hex)*

Mnemonic: *GS*

The cursor is moved to the beginning of the row above the one where it finds.
If the cursor finds in the last row, it will be moved to the beginning of the first line, that is in the Home position.

HOME

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: *28* *(1C Hex)*

Mnemonic: *FS*

ADDS Viewpoint

Code: *01*

Mnemonic: *SOH*

The cursor is moved to the Home position, that is the first row of the first column of the display.

CURSOR ABSOLUT POSITIONING

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: *27 61 r c (1B 3D r c Hex)*

Mnemonic: ESC = ASCII(r) ASCII(c)

ADDS Viewpoint

Codice: *27 89 r c (1B 59 r c Hex)*

Mnemonic: ESC Y ASCII(r) ASCII(c)

The cursor is moved to the absolut position indicated by "**r**" and "**c**".
These codes indicate the numbers of row and column of the display, plus an offset of **32** (**20 Hex**).
If, for example, the User wants to place the cursor in the Home position (row 0, column 0) under the ADDS Viewpoint terminal emulation, he/she will have to send the sequence: **27 89 32 32**.
If the values of row and column are not compatible to the type of display installed, this command is ignored.

CHARACTERS DELETION COMMANDS

Here follows a list of the several characters deletion commands available according to the installed firmware.

CLEAR PAGE

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 ed ADDS Viewpoint

Code: *12 (0C Hex)*

Mnemonic: *FF*

The whole display is cleared and the cursor is placed in the Home position.

CLEAR END OF LINE

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: *23 (17 Hex)*

Mnemonic: *ETB*

ADDS Viewpoint

Code: *27 75 (1B 4B Hex)*

Mnemonic: *ESC K*

All characters between the cursor and the end of the line where it finds are deleted, included the one under the cursor.
The cursor remains in the same position.
If, for example, the cursor finds at the beginning of a line, this command will delete the whole line.

CURSOR ATTRIBUTES MANAGEMENT COMMANDS

Here follows a list of the several cursor attributes management commands available according to the installed firmware.

CURSOR DISABLED

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

Code: ***27 80*** ***(1B 50 Hex)***

Mnemonic: ***ESC P***

The cursor is disabled and no more visible.

CURSOR ENABLED AND FIXED

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

Code: ***27 79*** ***(1B 4F Hex)***

Mnemonic: ***ESC O***

The cursor is enabled and visible as a fixed line placed under the character.

NOTE

This command is not available when **Futaba 20x4 and 40x4 fluorescent displays** are connected.

CURSOR ENABLED AND FLASHING

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

Code: ***27 77*** ***(1B 4D Hex)***

Mnemonic: ***ESC M***

The cursor is enabled and visible as a flashing line placed under the character.

NOTE

This command is not available when **Futaba 20x4 and 40x4 fluorescent displays** are connected.

FLASHING BLOCK CURSOR

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

Code: ***27 81*** ***(1B 51 Hex)***

Mnemonic: ***ESC Q***

The cursor is enabled and visible as a flashing block.

NOTE

This command is not available for displays mounting an **Hitachi HD61830B** controller.

AUXILIARY SERIAL PORT MANAGEMENT COMMANDS

Here follows a list of the several auxiliary serial port management commands available according to the installed firmware.

ECHO ACTIVATION ON AUXILIARY SERIAL PORT

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: ***27 33 65 (1B 21 41 Hex)***

Mnemonic: ***ESC ! A***

ADDS Viewpoint

Code: ***27 51 (1B 33 Hex)***

Mnemonic: ***ESC 3***

This command activates the echo (re-transmission) on auxiliary serial port of all the characters received from the master control unit.

ECHO DEACTIVATION ON AUXILIARY SERIAL PORT

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: ***27 33 65 (1B 21 41 Hex)***

Mnemonic: ***ESC ! A***

ADDS Viewpoint

Code: ***27 52 (1B 34 Hex)***

Mnemonic: ***ESC 4***

The echo (re-transmission, see previous command) on the auxiliary serial port is disabled.

NOTE

In **TVI 950** terminal emulation mode, the echo is managed by the same command.
In that case the command has a bistable behaviour, in fact it activates and deactivates this feature according to its previous status (echo is disabled by **default**).

CHARACTERS ATTRIBUTES MANAGEMENT COMMANDS

Here follows a list of the several characters attributes management commands available only when running **TRKDGL** firmware.

REVERSE ATTRIBUTE ACTIVATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: ***27 33 50*** ***(1B 21 32 Hex)***
Mnemonic: ***ESC ! 2***

ADDS Viewpoint

Code: ***27 48 80 14*** ***(1B 30 50 0E Hex)***
Mnemonic: ***ESC 0 P SO***

This command activates the reverse representation mode for alphanumeric characters.
After this command, the representable characters will be visualized in reverse mode, up to when this attribute wiil be disabled.

NOTE
This command is not available for displays mounting an **Hitachi HD61830B** controller.

REVERSE ATTRIBUTE DEACTIVATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950

Code: ***27 33 48*** ***(1B 21 30 Hex)***
Mnemonic: ***ESC ! 0***

ADDS Viewpoint

Code: ***27 48 80 15*** ***(1B 30 50 0F Hex)***
Mnemonic: ***ESC 0 P SI***

This command deactivates the reverse representation mode for alphanumeric characters.
After this command, the representable characters will be visualized in normal mode, restoring the default visualization mode.

NOTE
This command is not available for displays mounting an **Hitachi HD61830B** controller.

KEYBOARD MANAGEMENT COMMANDS

Here follows a list of the several **BG keyboard** management commands available only when running **TRKDGL** firmware.

KEYCLICK ON LOCAL BUZZER ACTIVATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 56* *(1B 38 Hex)*

Mnemonic: *ESC 8*

The **KeyClick** feature, that is a buzzer sound feedback when a key is pressed, is enabled. The sound will be produced by the buzzer installed on the **RKD LT** board.

KEYCLICK ON LOCAL BUZZER DEACTIVATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 57* *(1B 39 Hex)*

Mnemonic: *ESC 9*

The **KeyClick** feature, that is a buzzer sound feedback when a key is pressed, is disabled. No sound will be produced by the buzzer installed on the **RKD LT** board on a key hit.

KEYCLICK ON BG KEYBOARD ACTIVATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 58* *(1B 3A Hex)*

Mnemonic: *ESC :*

The **KeyClick** feature, that is a buzzer sound feedback when a key is pressed, is enabled. The sound will be produced by the buzzer installed on the **BG keyboard**.

KEYCLICK ON BG KEYBOARD DEACTIVATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 59* *(1B 3B Hex)*

Mnemonic: *ESC ;*

The **KeyClick** feature, that is a buzzer sound feedback when a key is pressed, is disabled. No sound will be produced by the buzzer installed on the **BG keyboard** on a key hit.

EXTERNAL LEDS MANAGEMENT COMMANDS

Here follows a list of the several external LEDs management commands available according to the installed firmware.
Please remark that the LEDs work correctly only if they are connected in common cathod mode, as previously explained.

EXTERNAL LEDS ACTIVATION

Available in	
TRKDAF	YES
TRKDGL Alphanumeric	YES
TRKDGL Graphic	YES

TVI 950 e ADDS Viewpoint

Code: *27 50 mask (1B 32 mask Hex)*
Mnemonic: *ESC 2 ASCII(mask)*

The 8 external LEDs, connected to CN1, are actived, according to the instructions given in **mask**. The bits of **mask** have the following meaning:

	bit 7	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**	**bit0**
mask =	*LED7*	*LED6*	*LED5*	*LED4*	*LED3*	*LED2*	*LED1*	*LED0*

If a certain bit is set to **0** the corresponding LED will be OFF, viceversa il wil be ON if the bit is set to **1**.
Please refer to figures 3 and 4, reported in the previous pages, to match the LEDs numbers here used to the corresponding pins on CN1 connector of **RKD LT** board.
If, for example, the User wants to activate LED1 and LED4, he/she will have to send the sequence: **27 50 18**.

EXTERNAL LEDS REVERSED ACTIVATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	YES
TRKDGL Graphic	YES

TVI 950 e ADDS Viewpoint

Code: *27 53 mask (1B 35 mask Hex)*
Mnemonic: *ESC 5 ASCII(mask)*

The 8 external LEDs, connected to CN1, are actived, according to the instructions given in **mask** but complemented in comparision with the previous command.
The bits of **mask** have the following meaning:

	bit 7	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**	**bit0**
mask =	*LED7*	*LED6*	*LED5*	*LED4*	*LED3*	*LED2*	*LED1*	*LED0*

If a certain bit is set to **1** the corresponding LED will be OFF, viceversa il wil be ON if the bit is set to **0**.
Please refer to figures 3 and 4, reported in the previous pages, to match the LEDs numbers here used to the corresponding pins on CN1 connector of **RKD LT** board.
If, for example, the User wants to activate LED1 and LED4, he/she will have to send the sequence: **27 53 237**.

ALPHANUMERIC SCREENSHOTS MANAGEMENT COMMANDS

Here follows a list of the several alphanumeric screen management commands available according to the installed firmware.

SCREENSHOT VISUALIZATION

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

Code: ***27 122 n.scr. H n.scr. L***
(1B 7A n.scr. H n.scr. L Hex)

Mnemonic: ***ESC z ASCII(n.scr. H) ASCII(n.scr. L)***

The alphanumeric screenshot whose number is **n.scr.** is displayed.
n.scr. is a 16 bits parameter that must be sent as two bytes, first the high byte (bits 15÷8) then the low bytes (bits 7÷0).

An alphanumeric screenshot is a page of text that contains enough characters to cover exactly the screen of the display being used; these screenshots are generated using the **RKD_EDIT.EXE** program and are stored in the EPROM on the **RKD LT**, where also the firmware is sored.
It is so possible to create the text masks to recall next in visualization; this allows to reduce the size of the User program and the amount of data to transmit. The **RKD_EDIT.EXE** program creates a **binary** file sized 64 KBytes; this file will be stored in the EPROM on the **RKD LT**, using an EPROM programmer, starting from the following addresses:

*EPROM for **TRKDAF** firmware -> 8000 Hex*
*EPROM for **TRKDGL** firmware -> 10000 Hex*

Please remark that the EPROM involved is a 27C010 sized 128 KBytes whose content must not be modified.
For further informations about how to use the **RKD_EDIT.EXE** program, please refer to its own on line help.

The **n.scr.** parameter must be equal to or grater than **1** and equal to or lower than the number of the last screenshot available, indicated by the **RKD_EDIT.EXE** program during the creation of the alphanumeric screenshots.
If, for example, the User wants to display the screenshot number 258, he/she will have to sen the sequence: **27 122 1 2**.

NOTE
After the visualization of a screen shot, the cursor is placed in the Home position.

SCREENSHOT MEMORIZATION IN EEPROM

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

Code: 27 33 67 n.scr. car. 1 ... car. n
(1B 21 43 n.scr. car. 1 ... car. n Hex)
Mnemonic: ESC ! C ASCII(n.scr.) ASCII(car. 1)...ASCII(car. n)

This command stores in the EEPROM on the **RKD LT** board the alphanumeric screenshot whose number is indicated by **n.scr.**
An alphanumeric screenshot is a page of text that contains enough characters to cover exactly the screen of the fluorescent display being used; so the number of characters that must be sent is always:

n *= NUMBER OF ROWS x NUMBER OF COLUMNS*

The characters in the screen shot must be visualizable, so their codes must be in the range 32÷255 (20÷FF Hex), while the **n.scr.** parameter must be equal to or grater than **1** and equal to or lower than the number of the last screenshot available, which depends on the type of display used and the size of the EEPROM installed, according to the following table:

DISPLAY	EEPROM 24c04 (512 bytes)	EEPROM 24c16 (2048 bytes)
Futaba 20x2 characters	12	50
Futaba 20x4 characters	6	25
Futaba 40x1 characters	12	50
Futaba 40x2 characters	6	25
Futaba 40x4 characters	3	12

FIGURE 38: NUMBER OF SCREENSHOTS STORABLE IN EEPROM

SCREENSHOT VISUALIZATION FROM EEPROM

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

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

The screenshot, corresponding to number **n.scr.**, is recalled from the EEPROM and visualized on the display, starting from the first character.
The number of characters contained in the screenshot depends on the type of display used, as stated in the previous paragraph (***n*** *= Rows x Columns*), while the value of **n.scr.** must be included in the range **1÷n.max**, where the value n.max can be obtained by the previous table.

NOTE
After the visualization of a scree shot, the cursor is placed in the Home position.

SCREENSHOT READ FROM EEPROM

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

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

The screenshot, corresponding to number **n.scr.**, is recalled from the EEPROM and sent to the master control unit, starting from the first character.
The number of characters contained in the screenshot depends on the type of display used, as stated in the previously (***n*** = *Rows x Columns*), while the value of **n.scr.** must be included in the range **1÷n.max**, where the value n.max can be obtained by the previous table.

GRAPHIC MODE COMMANDS

Here follows a list of the several graphic mode commands, available only under **TRKDGL**.

REVERSE ATTRIBUTE SETTING

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 09 reverse* *(1B 09 reverse Hex)*
Mnemonic: *ESC HT ASCII(reverse)*

This command enables or diables the REVERSE attribute, according to the value of its parameter, in detail:

reverse:	**0**	->	REVERSE attribute *disabled* (**default**)
	1	->	REVERSE attribute *enabled*

This command is useful when the User wishes to delete a graphic primitive or parts of the display, in fact it is enough to activate the REVERSE attribute and redraw the picture to be deleted.

CHARACTERS ZOOM SETTING

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 01 zoom* *(1B 01 zoom Hex)*
Mnemonic: *ESC SOH 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:

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

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

CHARACTERS ELONGATION SETTING

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Codice:	***27 07 elong***	***(1B 07 elong Hex)***
Mnemonico:	***ESC BEL ASCII(elong)***	

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

elong:	**0**	->	*Normal:*	**8x8** pixels characters (**default**)
	1	->	*Vertical:*	**8x16** pixels characters
	2	->	*Orizzontal:*	**16x8** pixels characters

CHARACTER CURSOR PLACEMENT

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code:	***27 00 r c***	***(1B 00 r c Hex)***
Mnemonic:	***ESC NUL ASCII(r) ASCII(c)***	

This command moves the cursor to the row and column indicated by **r** and **c**; this position is not absolute but depends on the values of zoom or elongation being used, so this position depends on the character size (for example, position 1,1 and zoom 0 correspond to the pixel whose coordinates are **8, 8** while the same position and zoom 1 correspond to the pixel having coordinates **16, 16**).
The values of **r** and **c** must be in the range 1÷maximum, where maximun depends on the type of graphic display used and the value of zoom currently set.

ABSOLUTE CURSOR PLACEMENT

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Codice:	***27 06 y x 00***	***(1B 06 y x 00 Hex)***
Mnemonico:	***ESC ACK ASCII(y) ASCII(x) NUL***	

This command moves the cursor to the row and column indicated by **x** and **y**; this position is absolute, so it does not depend on the valueo of the other parameters.
The characters received after this command will be displayed from the indicated point, going to the right and to the bottom.
The values of **x** and **y** must be in the range 1÷maximum, where maximun depends on the type of graphic display used.

NOTE

The code **00** (**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.

CHARACTERS WRITE DIRECTION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Codice: *27 10 dir* *(1B 0A dir Hex)*
Mnemonico: *ESC LF ASCII(dir)*

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

dir: **0** -> *Horizontal* characters writing (**default**)
1 -> *Vertical* characters writing

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

STRING VISUALIZATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 08 car. 1 … car. n 13*
(1B 08 car. 1 … car. n 0D Hex)
Mnemonic: *ESC BS ASCII(car. 1)…ASCII(car. n) CR*

This command visualizes a string **n** characters long terminated by code **13** (**CR**), starting from the current cursor position and using the currently set attrubutes (zoom, elongation, reverse, etc.).
The string is displayed only after the reception of the **CR** and must contain only characters in the range **32÷126** (**20÷7E Hex**).

NOTE
This command displays the received characters only after the completion of the transmission, differently from the direct visualization commands. This way it is possible to avoid partial visualizations due to communication delays (buffer full, slow transmission from the master control unit, etc.).

ONE PIXEL VISUALIZATION

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 29 y x 00* *(1B 1D y x 00 Hex)*
Mnemonic: *ESC GS ASCII(y) ASCII(x) NUL*

This command turns ON the pixel at the coordinates **x** and **y**.
The values of the **x** and **y** parameters must range from **0** to a maximum that depends on the type of display used.

NOTE
The code **00** (**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.

DRAWING A LINE

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 03 y1 x1 00 y2 x2 00*
(1B 03 y1 x1 00 y2 x2 00 Hex)
Mnemonic: *ESC ETX 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 these parameters must range from **0** to a maximum that depends on the type of display used.

NOTE: The code **00** (**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.

DRAWING A RECTANGLE

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 02 y1 x1 00 y2 x2 00*
(1B 02 y1 x1 00 y2 x2 00 Hex)
Mnemonic: *ESC STX 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 these parameters must range from **0** to a maximum that depends on the type of display used.

NOTE: The code **00** (**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.

DRAWING A FILLED RECTANGLE

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 04 y1 x1 00 y2 x2 00*
(1B 04 y1 x1 00 y2 x2 00 Hex)
Mnemonic: *ESC EOT 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 these parameters must range from **0** to a maximum that depends on the type of display used. 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 **00** (**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.

DRAWING A CIRCUMFERENCE

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 05 y x 00 r arc.in. n.arc.*
(1B 05 y x 00 r arc.in. n.arc. Hex)
Mnemonic: *ESC ENQ ASCII(y) ASCII(x) NUL ASCII(r) ASCII(arc.in.) ASCII(n.arc.)*

This command draws a circumference, or an arc of circumference, having ray 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-circumferences or quarters of circumferences, 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.

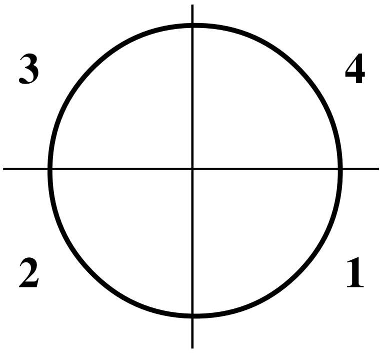


FIGURE 39: ARCS NUMERATION AND DISPOSITION

If, for example, the User wants to display a "C" shaped half-circumference having ray of 20 pixels and center located at (120, 60), he/she will have to send the sequence: **27 5 60 120 0 20 2 3**.

NOTE: The code **00** (**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.

DRAWING A CIRCUMFERENCE WHOSE RAY IS 3 PIXELS

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint
Code: *27 39 y x 00 (1B 27 y x 00 Hex)*
Mnemonic: *ESC ' ASCII(y) ASCII(x) NUL*

This command draws a circumference, having ray equal to **3** 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.

NOTE: The code **00** (**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.

DRAWING A FILLED CIRUMFERENCE WHOSE RAY IS 3 PIXELS

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

<u>TVI 950 e ADDS Viewpoint</u>

Code: *27 38 y x 00 (1B 26 y x 00 Hex)*
Mnemonic: *ESC & ASCII(y) ASCII(x) NUL*

This command draws a filled circumference, having ray equal to **3** 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.

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

NOTE: The code **00** (**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.

DRAWING AN ARROW POINTING TO THE TOP

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

<u>TVI 950 e ADDS Viewpoint</u>

Code: *27 37 y x 00 (1B 25 y x 00 Hex)*
Mnemonic: *ESC % ASCII(y) ASCII(x) NUL*

This command draws an arrow oriented to the top, as shown in the following figure:

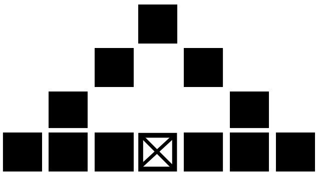


FIGURE 40: TOP ORIENTED ARROW

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.
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.

NOTE: The code **00** (**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.

DRAWING AN ARROW POINTING TO THE BOTTOM

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: ***27 33 y x 00 (1B 21 y x 00 Hex)***

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

This command draws an arrow oriented to the bottom, as shown in the following figure:

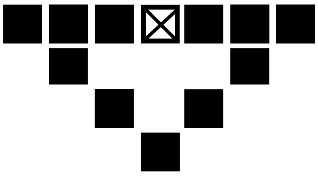


FIGURE 41: BOTTOM ORIENTED ARROW

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.
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.

NOTE: The code **00** (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.

DRAWING AN ARROW POINTING TO THE RIGHT

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: ***27 31 y x 00 (1B 1F y x 00 Hex)***

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

This command draws an arrow oriented to the right, as shown in the following figure:

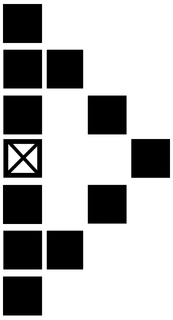


FIGURE 42: RIGHT ORIENTED ARROW

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.
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.

NOTE: The code **00** (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.

DRAWING AN ARROW POINTING TO THE LEFT

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: ***27 35 y x 00 (1B 23 y x 00 Hex)***

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

This command draws an arrow oriented to the left, as shown in the following figure:

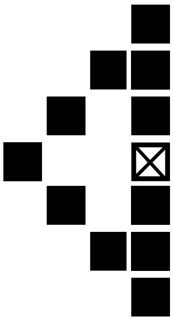


FIGURE 43: LEFT ORIENTED ARROW

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.
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.

NOTE: The code **00** (**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.

DRAWING A FILLED ARROW POINTING TO THE TOP

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Codice: ***27 36 y x 00 (1B 24 y x 00 Hex)***

Mnemonico: ***ESC $ ASCII(y) ASCII(x) NUL***

This command draws a filled arrow oriented to the top, as shown in the following figure:

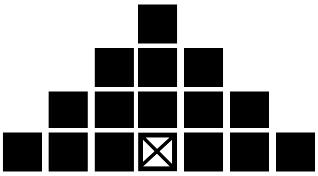


FIGURE 44: TOP ORIENTED FILLED ARROW

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.
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.

NOTE: The code **00** (**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.

DRAWING A FILLED ARROW POINTING TO THE BOTTOM

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 32 y x 00 (1B 20 y x 00 Hex)*

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

This command draws a filled arrow oriented to the bottom, as shown in the following figure:

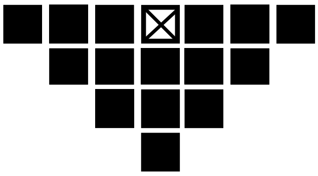


FIGURE 45: BOTTOM ORIENTED FILLED ARROW

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.

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.

NOTE: The code **00** (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.

DRAWING A FILLED ARROW POINTING TO THE RIGHT

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: *27 30 y x 00 (1B 1E y x 00 Hex)*

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

This command draws a filled arrow oriented to the right, as shown in the following figure:

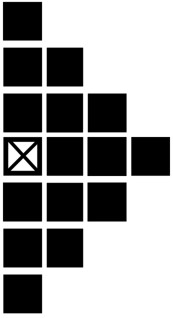


FIGURE 46: RIGHT ORIENTED FILLED ARROW

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.

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.

NOTE: The code **00** (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.

DRAWING A FILLED ARROW POINTING TO THE LEFT

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

__TVI 950 e ADDS Viewpoint__

Code: *27 34 y x 00 (1B 22 y x 00 Hex)*
Mnemonic: *ESC " ASCII(y) ASCII(x) NUL*

This command draws a filled arrow oriented to the left, as shown in the following figure:

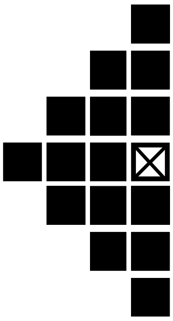


Figure 47: Left oriented filled arrow

The parameters **x** and **y** correspond to the coordinates of the pixel indicated by the "X" in the above figure.
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.

NOTE: The code **00** (**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.

DRAWING GRADUATED AXISES AND GRID

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	**YES**

__TVI 950 e ADDS Viewpoint__

Code: *27 28 y x 00 ly lx 00 gr dt*
(1B 1C y x 00 ly lx 00 gr dt Hex)
Mnemonic: *ESC FS ASCII(y) ASCII(x) NUL ...*
... ASCII(ly) ASCII(lx) NUL ASCII(gr) ASCII(dt)

This command allows to create quickly the cartesian axises with graduation and a grid on the display.

x : Vertical coordinate of the cartesian axises origin; must be in the range **3÷max**.
y : Horizontal coordinate of the cartesian axises origin; must be in the range **3÷max**.
lx : Length of X axis;must be in the range **1÷max**.
ly : Length of Y axis;must be in the range **1÷max**.
gr : **0** = Grid hidden; **1** = Gri 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 **00** (**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.

MISCELLANEOUS COMMANDS

Here follows a list of the several commands that manage various functions, available according to the installed firmware.

BUZZER TIMED ACTIVATION

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: ***07***

Mnemonic: ***BEL***

The buzzer on the **RKD LT** board is activated for about one tenth of second.

PRESENCE CODE REQUEST

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: ***27 78 (1B 4E Hex)***

Mnemonic: ***ESC N***

After having received this command, the **RKD LT** board send the code **170** (**AA Hex**) to the master control unit.
This allows to know quickly whether the board is working correctly or it has terminated its initialization after a Power ON or a Reset.

EEPROM DATA WRITE

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code: ***27 54 ind L ind H n byte 1 ... byte n***
(1B 34 ind L ind H n byte 1 ... byte n Hex)

Mnemonic: ***ESC 6 ASCII(ind L) ASCII(ind H) ASCII(n) ...***
... ASCII(byte 1)...ASCII(byte n)

This command stores a block of **n** bytes in the EEPROM on the **RKD LT** board, starting from the address **ind**. This last is a 16 bits parameter that must be sent as two bytes, respectively the low byte (bits 7÷0) first and the high byte (bits 15÷8) next; its value must renge from **32** (**20 Hex**) to a maximuum that depends on the size of the EEPROM installed (**24c04 = 512 bytes**, **24c16 = 2048 bytes**) and must allow to store wholly the **n** bytes sent.
Data to be written must be in the range **0÷255** (**0÷FF Hex**).

If, for example, the User wants to store the bytes 24, 65 and 2 from the address 258, he/she will have to send the sequence: **27 6 2 1 3 24 65 2**.

EEPROM DATA READ

Available in	
TRKDAF	*NO*
TRKDGL Alphanumeric	**YES**
TRKDGL Graphic	**YES**

TVI 950 e ADDS Viewpoint

Code:	*27 55 ind L ind H n*
	(1B 37 ind L ind H n Hex)
Mnemonic:	*ESC 7 ASCII(ind L) ASCII(ind H) ASCII(n)*

This command reads and returns a block of **n** bytes from the EEPROM on the **RKD LT** board, starting from the address **ind**. This last is a 16 bits parameter that must be sent as two bytes, respectively the low byte (bits 7÷0) first and the high byte (bits 15÷8) next; its value must renge from **32** (**20 Hex**) to a maximuum that depends on the size of the EEPROM installed (**24c04 = 512 bytes**, **24c16 = 2048 bytes**) and must allow to store wholly the **n** bytes sent.

DIMMING FOR FLUORESCENT DISPLAY

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	*NO*

TVI 950 e ADDS Viewpoint

Code:	*27 33 dimming (1B 21 dimming Hex)*
Mnemonic:	*ESC ! ASCII(dimming)*

Dimming for Futaba fluorescent display, as indicated by the **dimming** parameter:

dimming:	*00*	->	Sets luminosity to **0%**.
	32 (20 Hex)	->	Sets luminosity to **20%**.
	48 (30 Hex)	->	Sets luminosity to **30%**.
	64 (40 Hex)	->	Sets luminosity to **40%**.
	96 (60 Hex)	->	Sets luminosity to **60%**.
	112 (70 Hex)	->	Sets luminosity to **70%**.
	128 (80 Hex)	->	Sets luminosity to **80%**.
	255 (FF Hex)	->	Sets luminosity to **100%**.

NOTE

The luninosity levels available depend on the type of fluorescent display used, please refer to the display manufacturer documentation to know which valus can be used.

CUSTOM CHARACTERS

Available in	
TRKDAF	**YES**
TRKDGL Alphanumeric	*NO*
TRKDGL Graphic	*NO*

<u>TVI 950 e ADDS Viewpoint</u>

Code: ***27 33 66 ind byte 1 ... byte 5***
(1B 21 42 ind byte 1 ... byte 5 Hex)

Mnemonic: ***ESC ! B ASCII(ind) ASCII(byte 1) ... ASCII(byte 5)***

This command allows to create custom characters that will be stored starting from the **ind** address in the character memory of the fluorescent display connected to the **RKD LT** board. The board will display a custom character whenever it will receive its corresponding address **ind**.
This command is available only when using Futaba **M202SD01BA** (20x2 characters "small"), **M204SD01AA** (20x4 characters) , **M404SD01BA** (40x4 characters) fluorescent displays and compatible.
The values of **ind** depend on the display in use, in detail:

M202SD01BA : **ind** must be in the range **248÷255** (**F8÷FF Hex**)
M204SD01AA : **ind** must be in the range **205÷207** (**CD÷CF Hex**)
M404SD01BA : **ind** must be in the range **252÷254** (**FC÷FE Hex**)

The remaining 5 bytes of the sequence contain the pattern of the custom character; for further informations about the meaning of their bits please refer to the specific documentation of the display.

EXTERNAL CARDS FOR RKD LT

RKD LT board can be connected to a wide range of **grifo®** cards , all the CPU (**GPC®** serie) cards can act as master control card, increasing its versatility.
A complete set of modules makes easier the connection to the on board devices.
Hereunder follows a short description for some of these boards.

SPB 04-08

Switch Power BUS 4-8 slots

Motherboard with 4-8 slots industrial BUS **ABACO®**; pitch 4 TE; standard power supply connectors; termination resistors;F type connector for **SPC xxx** power supply board; holes for rack mounting.

ABB 05

ABACO® Block BUS 5 slots

5 slots **ABACO®** mother board with Power Supply. Double power supply built-in; 5Vdc 2,5A section for powering the on-board logic; second section at 24Vdc 400mA galvanically coupled, for the optocoupled input lines. Auxiliary connector for **ABACO®** I/O BUS. Housing with hooks for DIN Ω rails.

GPC® 51 - GPC® 51D

General Purpose Controller 51 family

11 MHz 51 INTEL or 22 MHz 320 DALLAS µP BASIC type included; 16/24 TTL I/O lines; 1 or 2 RS 232 lines; Buzzer; RTC and 32K RAM backed Lithium battery; EPROM and EEPROM programmer; readable dip switch; 3 Timer Counter; 4 11 bit A/D lines and Keyboard Display Controller.

GPC® 553

General Purpose Controller 80C552 (3 TYPE)

80C552 µP 33 MHz; 1 RS 232 line; 1 RS 232 or RS 422-485 or Current Loop line; 16 TTL I/O lines; 8 A/D lines at 10 bits; 3 Timer Counter; RTC; 64K EPROM; 64K RAM (32K backed RAM-32K DIL EEPROM); 8K serial EEPROM; 2 PWM lines; Watch dog; 5 readable Dip switch; LCD interface.

GPC® 188F

General Purpose Controller 80C188

80C188 µP 20 MHz; 256K FLASH; 256K RAM Lithium battery backed; 8K serial EEPROM; 1 RS 232 line; 1 RS 232 or RS 422-485 or Current Loop line; 24 TTL I/O lines; RTC; 8 A/D lines at 12 bits; Watch dog; 8 Dip switch; 3 Timer Counter.

GPC® 15A

General Purpose Controller 84C15

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

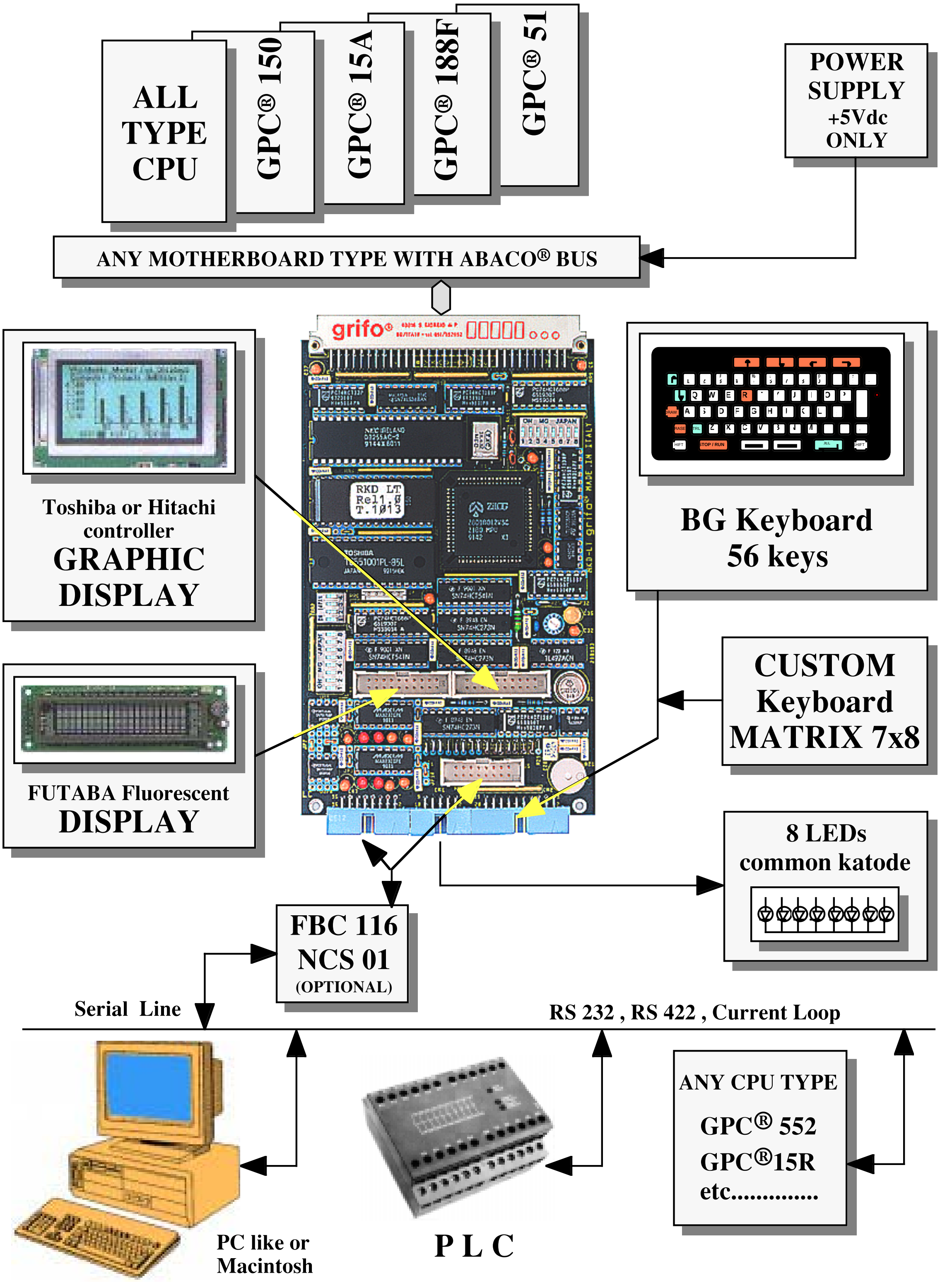


FIGURE 48: POSSIBLE CONNECTIONS DIAGRAM

GPC® 150

General Purpose Controller 84C15

CPU Z80 at 16 MHz. Full CMOS card; 512K EPROM or FLASH; 512K RAM; RTC; Backed by external lithium battery; 4M serial FLASH; 1 RS 232 line + 1 RS 232 or RS 422-485 or current loop line; 40 I/O TTL; 2 timer/counter; 2 watch dog; dip switch; EEPROM; 12 bits A/D lines; activity LED.

GPC® 15R

General Purpose Controller 84C15 with Relays

84C15 µP 16 MHz; 1 RS 232 line; 1 RS 232 or RS 422-485 or Current Loop line; 24 TTL I/O lines; 16 Opto-in 8 Relays; 4 Opto Coupled Timer Counter; RTC; 512K EPROM or FLASH; 512K backed RAM; 8K serial EEPROM; 8K Backed RAM Modul; Buzzer; Watch dog; 12 readable Dip switch; LCD interface.

GPC® 153

General Purpose Controller 84C15 (3 TYPE)

84C15 µP 16 MHz; 1 RS 232 line; 1 RS 232 or RS 422-485 or Current Loop line; 16 TTL I/O lines; 8 A/D lines at 12 bits; 4 Timer Counter; RTC; 512K EPROM or FLASH; 512K backed RAM; 8K serial EEPROM; Buzzer; Watch dog; 8 readable Dip switch; LCD interface.

GPC® 884

General Purpose Controller 80C188ES (4 TYPE)

80C188ES µP 40 MHz; 1 RS 232 line; 1 RS 232 or RS 422-485; 16 TTL I/O lines; 11 A/D lines at 12 bits; 3 Timer Counter; RTC; 512K EPROM or FLASH; 512K backed RAM; 8K serial EEPROM; Watch dog; 1 readable Dip switch; LCD interface.

GPC® 114

General Purpose Controller 68HC11 (4 TYPE)

68HC11 µP 16 MHz; 1 RS 232 or RS 422-485; 18 TTL I/O lines; 8 A/D lines at 8 bits; 3 Timer Counter; RTC; 32K EPROM; 32K backed RAM; 512 DIL EEPROM; Watch dog; 1 readable Dip switch; LCD interface.

NCS 01

New Connector Support

Serial communication support board;standard 16 pins RS 232 **ABACO®** serial connector; cquick extraction connectors to connect directly to the field; 2 D25 connectors with standard RS 232 pin out; selectable DCE/DTE.

FBC xxx

Flat BLOCK Contact

This interconnection system “wires to board” allows the connection to many types of flat cable connectors to a terminal for external connections. Other interfacing for most popular connectors such as D, mini DIN, ACCESS.bus™, and so on, are available. Connection for DIN C Type and Ω rails.

BG Keyboard

56 keys keyboard on a 7x8 matrix; coloured and serigraphed keys; LED; Buzzer; comfortable plastic container; 20 pins flat to connect directly to **RKD LT** and **MDU-RKD** board.

IAF 404

Interface Adapter Futaba 40x4

Interface module for 40x4 charaters Futaba **M404SD01BA** fluorescent displays; 20 pins connector to connect directly to **RKD LT** and **MDU-RKD** board; 20 pins standard I/O **ABACO®** connector; 26 pins connector to connect directly the display; screw terminal connector for power supply; holes for mounting.

APPENDIX A: COMMANDS SUMMARIZING TABLES

Here follow the commands summarizing tables,complete of command codes, for the **TRKDAL** and **TRKDGL**, considering their several operational modes.

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*
HOME	01	01	SOH
CURSOR LEFT	21	15	NACK
CURSOR RIGHT	06	06	ACK
CURSOR DOWN	10	0A	LF
CURSOR UP	26	1A	SUB
CARRIAGE RETURN	13	0D	CR
Carriage Return + Line Feed	29	1D	GS
Cursor Absolut Positioning	27 89 r c	1B 59 r c	ESC Y ASCII(r) ASCII(c)
CLEAR PAGE	12	0C	FF
CLEAR END OF LINE	27 75	1B 4B	ESC K
Cursor Disabled	27 80	1B 50	ESC P
Cursor Enabled And Fixed	27 79	1B 4F	ESC O
Cursor Enabled And Flashing	27 77	1B 4D	ESC M
BUZZER Timed Activation	07	07	BEL
ECHO Activation On Auxiliary Serial Port	27 51	1B 33	ESC 3
ECHO Deactivation On Auxiliary Serial Port	27 52	1B 34	ESC 4
Character Visualization	32÷255	20÷FF	SP÷ASCII(255)
Screenshot Visualization	27 122 n.scr. H n.scr. L	1B 7A n.scr. H n.scr. L	ESC z ASCII(n.scr. H) ASCII(n.scr. L)
Screenshot Memorization In EEPROM	27 33 67 n.scr. car. 1 … car. n	1B 21 43 n.scr. car. 1 … car. n	ESC ! C ASCII(n.scr.) ASCII(car. 1)…ASCII(car. n)
Screenshot Visualization From EEPROM	27 33 68 n.scr.	1B 21 44 n.scr.	ESC ! D ASCII(n.scr.)
Screenshot Read From EEPROM	27 33 69 n.scr.	1B 21 45 n.scr.	ESC ! E ASCII(n.scr.)
External LEDs Activation	27 50 mask	1B 32 mask	ESC 2 ASCII(mask)
Dimming For Fluorescent Displays	27 33 dimming	1B 21 dimming	ESC ! ASCII(dimming)
Custom Characters	27 33 66 ind byte 1 ... byte 5	1B 21 42 ind byte 1 ... byte 5	ESC ! B ASCII(ind) ASCII(byte 1) ... ASCII(byte 5)

FIGURE A1: TRKDAF COMMANDS (ADDS VIEWPOINT) TABLE

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*
HOME	28	1C	FS
CURSOR LEFT	08	08	BS
CURSOR RIGHT	09	09	HT
CURSOR DOWN	10	0A	LF
CURSOR UP	11	0B	VT
CARRIAGE RETURN	13	0D	CR
Carriage Return + Line Feed	29	1D	GS
Cursor Absolut Positioning	27 61 r c	1B 3D r c	ESC = ASCII(r) ASCII(c)
CLEAR PAGE	12	0C	FF
CLEAR END OF LINE	23	17	ETB
Cursor Disabled	27 80	1B 50	ESC P
Cursor Enabled And Fixed	27 79	1B 4F	ESC O
Cursor Enabled And Flashing	27 77	1B 4D	ESC M
BUZZER Timed Activation	07	07	BEL
ECHO Activation On Auxiliary Serial Port (bistable)	27 33 65	1B 21 41	ESC ! A
ECHO Deactivation On Auxiliary Serial Port (bistable)	27 33 65	1B 21 41	ESC ! A
Character Visualization	32÷255	20÷FF	SP÷ASCII(255)
Screenshot Visualization	27 122 n.scr. H n.scr. L	1B 7A n.scr. H n.scr. L	ESC z ASCII(n.scr. H) ASCII(n.scr. L)
Screenshot Memorization In EEPROM	27 33 67 n.scr. car. 1 … car. n	1B 21 43 n.scr. car. 1 … car. n	ESC ! C ASCII(n.scr.) ASCII(car. 1)…ASCII(car. n)
Screenshot Visualization From EEPROM	27 33 68 n.scr.	1B 21 44 n.scr.	ESC ! D ASCII(n.scr.)
Screenshot Read From EEPROM	27 33 69 n.scr.	1B 21 45 n.scr.	ESC ! E ASCII(n.scr.)
External LEDs Activation	27 50 mask	1B 32 mask	ESC 2 ASCII(mask)
Dimming For Fluorescent Displays	27 33 dimming	1B 21 dimming	ESC ! ASCII(dimming)
Custom Characters	27 33 66 ind byte 1 ... byte 5	1B 21 42 ind byte 1 ... byte 5	ESC ! B ASCII(ind) ASCII(byte 1) ... ASCII(byte 5)

FIGURE A2: TRKDAF COMMANDS (TVI 950) TABLE

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*
HOME	01	01	SOH
CURSOR LEFT	21	15	NACK
CURSOR RIGHT	06	06	ACK
CURSOR DOWN	10	0A	LF
CURSOR UP	26	1A	SUB
CARRIAGE RETURN	13	0D	CR
Carriage Return + Line Feed	29	1D	GS
Cursor Absolut Positioning	27 89 r c	1B 59 r c	ESC Y ASCII(r) ASCII(c)
CLEAR PAGE	12	0C	FF
CLEAR END OF LINE	27 75	1B 4B	ESC K
Cursor Disabled	27 80	1B 50	ESC P
Cursor Enabled And Fixed	27 79	1B 4F	ESC O
Cursor Enabled And Flashing	27 77	1B 4D	ESC M
Flashing Block Cursor	27 81	1B 51	ESC Q
Reverse Attribute Activation	27 48 80 14	1B 30 50 0E	ESC 0 P SO
Reverse Attribute Deactivation	27 48 80 15	1B 30 50 0F	ESC 0 P SI
BUZZER Timed Activation	07	07	BEL
ECHO Activation On Auxiliary Serial Port	27 51	1B 33	ESC 3
ECHO Deactivation On Auxiliary Serial Port	27 52	1B 34	ESC 4
Character Visualization	32÷255	20÷FF	SP÷ASCII(255)
Screenshot Visualization	27 122 n.scr. H n.scr. L	1B 7A n.scr. H n.scr. L	ESC z ASCII(n.scr. H) ASCII(n.scr. L)

EEPROM Data Write	27 54
	ind L ind H n
	byte 1 ... byte n
		1B 36
		ind L ind H n
		byte 1 ... byte n
			ESC 6 ASCII(ind L)
			ASCII(ind H) ASCII(n)
			ASCII(byte 1)...ASCII(byte n)

EEPROM Data Read	27 55 ind L ind H n	1B 37 ind L ind H n	ESC 7 ASCII(ind L) ASCII(ind H) ASCII(n)
External LEDs Activation	27 50 mask	1B 32 mask	ESC 2 ASCII(mask)
External LEDs Reversed Activation	27 53 mask	1B 35 mask	ESC 5 ASCII(mask)
Keyclick On Local Buzzer Activation	27 56	1B 38	ESC 8
Keyclick On Local Buzzer Deactivation	27 57	1B 39	ESC 9
Keyclick On BG Keyboard Activation	27 58	1B 3A	ESC :
Keyclick On BG Keyboard Deactivation	27 59	1B 3B	ESC ;
Presence Code Request	27 78	1B 4E	ESC N

Figure A3: TRKDGL commands for alphanumeric mode (Adds Viewpoint) table

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*
HOME	28	1C	FS
CURSOR LEFT	08	08	BS
CURSOR RIGHT	09	09	HT
CURSOR DOWN	10	0A	LF
CURSOR UP	11	0B	VT
CARRIAGE RETURN	13	0D	CR
Carriage Return + Line Feed	29	1D	GS
Cursor Absolut Positioning	27 61 r c	1B 3D r c	ESC = ASCII(r) ASCII(c)
CLEAR PAGE	12	0C	FF
CLEAR END OF LINE	23	17	ETB
Cursor Disabled	27 80	1B 50	ESC P
Cursor Enabled And Fixed	27 79	1B 4F	ESC O
Cursor Enabled And Flashing	27 77	1B 4D	ESC M
Flashing Block Cursor	27 81	1B 51	ESC Q
Reverse Attribute Activation	27 33 50	1B 21 32	ESC ! 2
Reverse Attribute Deactivation	27 33 48	1B 21 30	ESC ! 0
BUZZER Timed Activation	07	07	BEL
ECHO Activation On Auxiliary Serial Port (bistable)	27 33 65	1B 21 41	ESC ! A
ECHO Deactivation On Auxiliary Serial Port (bistable)	27 33 65	1B 21 41	ESC ! A
Character Visualization	32÷255	20÷FF	SP÷ASCII(255)
Screenshot Visualization	27 122 n.scr. H n.scr. L	1B 7A n.scr. H n.scr. L	ESC z ASCII(n.scr. H) ASCII(n.scr. L)

EEPROM Data Write	27 54
	ind L ind H n
	byte 1 ... byte n
		1B 36
		ind L ind H n
		byte 1 ... byte n
			ESC 6 ASCII(ind L)
			ASCII(ind H) ASCII(n)
			ASCII(byte 1)...ASCII(byte n)

EEPROM Data Read	27 54 ind L ind H n	1B 37 ind L ind H n	ESC 7 ASCII(ind L) ASCII(ind H) ASCII(n)
External LEDs Activation	27 50 mask	1B 32 mask	ESC 2 ASCII(mask)
External LEDs Reversed Activation	27 53 mask	1B 35 mask	ESC 5 ASCII(mask)
Keyclick On Local Buzzer Activation	27 56	1B 38	ESC 8
Keyclick On Local Buzzer Deactivation	27 57	1B 39	ESC 9
Keyclick On BG Keyboard Activation	27 58	1B 3A	ESC :
Keyclick On BG Keyboard Deactivation	27 59	1B 3B	ESC ;
Presence Code Request	27 78	1B 4E	ESC N

FIGURE A4: TRKDGL COMMANDS FOR ALPHANUMERIC MODE (TVI 950) TABLE

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*
CURSOR DOWN	10	0A	LF
CARRIAGE RETURN	13	0D	CR
Character Cursor Placement	27 00 r c	1B 00 r c	ESC NUL ASCII(r) ASCII(c)
Cursor Absolut Positioning	27 06 y x 00	1B 06 y x 00	ESC ACK ASCII(y) ASCII(x) NUL
CLEAR PAGE	12	0C	FF
Characters Zoom Setting	27 01 zoom	1B 01 zoom	ESC SOH ASCII(zoom)
Characters Elongation Setting	27 07 elong	1B 07 elong	ESC BEL ASCII(elong)
Characters Write Setting	27 10 dir	1B 0A dir	ESC LF ASCII(dir)
Reverse Attribute Setting	27 09 reverse	1B 09 reverse	ESC HT ASCII(reverse)
BUZZER Timed Activation	07	07	BEL
Character Visualization	32÷126	20÷7E	SP÷~
String Visualization	27 08 car. 1 … car. n 13	1B 08 car. 1 … car. n 0D	ESC BS ASCII(car. 1)...ASCII(car. n) CR

EEPROM Data Write	27 54
	ind L ind H n
	byte 1 ... byte n
		1B 36
		ind L ind H n
		byte 1 ... byte n
			ESC 6 ASCII(ind L)
			ASCII(ind H) ASCII(n)
			ASCII(byte 1)...ASCII(byte n)

EEPROM Data Read	27 55 ind L ind H n	1B 37 ind L ind H n	ESC 7 ASCII(ind L) ASCII(ind H) ASCII(n)
External LEDs Activation	27 50 mask	1B 32 mask	ESC 2 ASCII(mask)
External LEDs Reversed Activation	27 53 mask	1B 35 mask	ESC 5 ASCII(mask)
Keyclick On Local Buzzer Activation	27 56	1B 38	ESC 8
Keyclick On Local Buzzer Deactivation	27 57	1B 39	ESC 9
Keyclick On BG Keyboard Activation	27 58	1B 3A	ESC :
Keyclick On BG Keyboard Deactivation	27 59	1B 3B	ESC ;
Presence Code Request	27 78	1B 4E	ESC N
One Pixel Visualization	27 29 y x 00	1B 1D y x 00	ESC GS ASCII(y) ASCII(x) NUL
Drawing A Line	27 03 y1 x1 00 y2 x2 00	1B 03 y1 x1 00 y2 x2 00	ESC ETX ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2) NUL
Drawing A Rectangle	27 02 y1 x1 00 y2 x2 00	1B 02 y1 x1 00 y2 x2 00	ESC STX ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2) NUL
Drawing A Filled Rectangle	27 04 y1 x1 00 y2 x2 00	1B 04 y1 x1 00 y2 x2 00	ESC EOT ASCII(y1) ASCII(x1) NUL ASCII(y2) ASCII(x2) NUL

FIGURE A5: TRKDGL COMMANDS FOR GRAPHIC MODE TABLE 1

COMMAND	*CODE*	*HEX CODE*	*MNEMONIC*

Drawing A Circumference	27 05
	y x 00
	r
	arc.in. n.arc.
		1B 05
		y x 00
		r
		arc.in. n.arc.
			ESC ENQ
			ASCII(y) ASCII(x) NUL
			ASCII(r)
			ASCII(arc.in.) ASCII(n.arc.)

Drawing A Circumference Whose Ray Is 3 Pixels	27 39 y x 00	1B 27 y x 00	ESC ' ASCII(y) ASCII(x) NUL
Drawing A Filled Circumference Whose Ray Is 3 Pixels	27 38 y x 00	1B 26 y x 00	ESC & ASCII(y) ASCII(x) NUL
Drawing An Arrow Pointing To The Top	27 37 y x 00	1B 25 y x 00	ESC % ASCII(y) ASCII(x) NUL
Drawing An Arrow Pointing To The Bottom	27 33 y x 00	1B 21 y x 00	ESC ! ASCII(y) ASCII(x) NUL
Drawing An Arrow Pointing To The Right	27 31 y x 00	1B 1F y x 00	ESC US ASCII(y) ASCII(x) NUL
Drawing An Arrow Pointing To The Left	27 35 y x 00	1B 23 y x 00	ESC # ASCII(y) ASCII(x) NUL
Drawing A Filled Arrow Pointing To The Top	27 36 y x 00	1B 24 y x 00	ESC $ ASCII(y) ASCII(x) NUL
Drawing A Filled Arrow Pointing To The Bottom	27 32 y x 00	1B 20 y x 00	ESC SP ASCII(y) ASCII(x) NUL
Drawing A Filled Arrow Pointing To The Right	27 30 y x 00	1B 1E y x 00	ESC RS ASCII(y) ASCII(x) NUL
Drawing A Filled Arrow Pointing To The Left	27 34 y x 00	1B 22 y x 00	ESC " ASCII(y) ASCII(x) NUL
Drawing Graduated Axises And Grid	27 28 y x 00 ly lx 00 gr dt	1B 1C y x 00 ly lx 00 gr dt	ESC FS ASCII(y) ASCII(x) NUL ASCII(ly) ASCII(lx) NUL ASCII(gr) ASCII(dt)

FIGURE A6: TRKDGL COMMANDS FOR GRAPHIC MODE TABLE 2

APPENDIX B: ALPHABETICAL INDEX