

QTP 12/R84

Quick Terminal Panel 12 tasti,
8 ingressi, 4 uscite a Relé

MANUALE UTENTE



grifo®

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

<http://www.grifo.com>

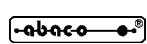
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



QTP 12/R84

Rel. 3.10

Edizione 07 Giugno 2004

 , GPC®, **grifo®**, sono marchi registrati della ditta **grifo®**



QTP 12/R84

Quick Terminal Panel 12 tasti,
8 ingressi, 4 uscite a Relé

MANUALE UTENTE

Ingombri: frontale **72x144 mm**; anteriore **9 mm**; posteriore **28 mm**. Contenitore in alluminio con cornice anteriore in plastica. Pannello anteriore con tastiera e mascherina in poliestere antigraffio. Contenitore completo di staffe di montaggio. Montaggio su pannello in modalità avanquadro. Protezione anteriore del pannello: **IP-54**. Il pannello operatore é disponibile con **3** modelli di display alfanumerici **LCD** e fluorescenti da **20x2** e grafico da **140** per **16** punti. Dimensione area visibile: **82x18 mm**. Tastiera a membrana da **12** tasti con doppia serigrafia: numerica e funzionale. **LED** di segnalazione comandabile da software, con diverse modalità. Tasca per la personalizzazione del nome del dispositivo e/o del **LED** di segnalazione. **Buzzer** per segnalazioni acustiche, attivabile via software. Microprocessore famiglia **8051**, con clock da **14,7** o **29,4 MHz** selezionabile via software. Tre diversi tipi di memorie: **32K FLASH EPROM**; **1,2K RAM**; **2K EEPROM**. Linea seriale configurabile in **RS 232**, **RS 422**, **RS 485**, **Current Loop**. Linea di comunicazione **CAN** dotata di apposito driver di linea. **8** ingressi optoisolati che possono essere indifferentemente **NPN** o **PNP**. Alcuni ingressi sono collegati a linee di conteggio ed **interrupt** hardware. **4 uscite** a relé da **5 A**. **2** linee di **I/O TTL** su apposito connettore, disponibile per gestione di dispositivi **I2C BUS**, **1-Wire**, **SPI**, ecc. Comodi connettori a **morsettiera a rapida estrazione**, per un facile cablaggio con il campo. Alimentazione ad ampio range, in **DC** o in **AC** a partire da **5Vdc**, fino a **24Vac**. **Potenza** complessiva, in funzione della configurazione usata, tra **2,5** a **3,3 W**. Protezione della logica di bordo tramite **TransZorb™**. Vasta disponibilità di linguaggi ed ambienti evoluti per lo sviluppo del firmware di gestione. Fornita con due diversi **firmware** di gestione: **seriale** o **libreria**. Possibilità di esecuzioni dedicate di pannello e programmi.

grifo®

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

<http://www.grifo.com>

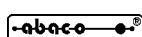
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



QTP 12/R84

Rel. 3.10

Edizione 07 Giugno 2004



, GPC®, **grifo®**, sono marchi registrati della ditta **grifo®**

Nessuna parte del presente manuale può essere riprodotta, trasmessa, trascritta, memorizzata in un archivio o tradotta in altre lingue, con qualunque forma o mezzo, sia esso elettronico, meccanico, magnetico ottico, chimico, manuale, senza il permesso scritto della **grifo®**.

IMPORTANTE

Tutte le informazioni contenute sul presente manuale sono state accuratamente verificate, ciononostante **grifo®** non si assume nessuna responsabilità per danni, diretti o indiretti, a cose e/o persone derivanti da errori, omissioni o dall'uso del presente manuale, del software o dell' hardware ad esso associato.

grifo® altresì si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l' intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo per **grifo®**.

Per le informazioni specifiche dei componenti utilizzati sui nostri prodotti, l'utente deve fare riferimento agli specifici Data Book delle case costruttrici o delle seconde sorgenti.

LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:



Attenzione: Pericolo generico

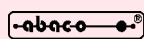


Attenzione: Pericolo di alta tensione



Attenzione: Dispositivo sensibile alle cariche elettrostatiche

MARCHI REGISTRATI



GPC®, **grifo®** : sono marchi registrati della **grifo®**.

Altre marche o nomi di prodotti sono marchi registrati dei rispettivi proprietari.

INDICE GENERALE

INTRODUZIONE	1
VERSIONE HARDWARE, FIRMWARE	2
INFORMAZIONI GENERALI	2
LINEA SERIALE	4
LED DI SEGNALEZIONE	6
DISPLAY	6
TASTIERA	6
BUZZER	7
MEMORIE	7
ALIMENTATORE DI BORDO	7
PROCESSORE	8
INTERFACCIA CAN	8
LINEE I/O TTL	8
INGRESSI DIGITALI OPTOISOLATI	8
USCITE DIGITALI A RELÉ	8
SPECIFICHE TECNICHE	10
CARATTERISTICHE GENERALI	10
CARATTERISTICHE FISICHE	11
CARATTERISTICHE ELETTRICHE	12
INSTALLAZIONE	14
CONNESSIONI	14
CN1 - CONNETTORE PER ALIMENTAZIONE	14
CN2 - CONNETTORE PER LINEA SERIALE	16
CN3 - CONNETTORE PER LINEA CAN	22
CN4 - CONNETTORE PER USCITE DIGITALI A RELÉ	24
CN5 - CONNETTORE PER INGRESSI DIGITALI OPTOISOLATI	26
J6 - CONNETTORE PER I/O TTL, 1-WIRE, I2C BUS	28
TRIMMER REGOLAZIONE CONTRASTO	30
JUMPERS	30
CONFIGURAZIONE LINEA SERIALE	32
COLLEGAMENTO LINEA CAN	34
INTERRUPTS	35
INTERFACCIAMENTO DEGLI I/O CON IL CAMPO	35
ALIMENTAZIONE	36
DESCRIZIONE SOFTWARE	38
DATI IN EEPROM	38
RAPPRESENTAZIONE DI CARATTERI SUL DISPLAY	39
BUFFER DI RICEZIONE	40
ACQUISIZIONE DELLA TASTIERA	40
CODICI DEI TASTI	40

FIRMWARE SERIALE .SER	42
SETUP LOCALE CON FIRMWARE SERIALE	42
MODALITA' DI COMUNICAZIONE CON FIRMWARE SERIALE	43
COMUNICAZIONE MASTER SLAVE CON FIRMWARE SERIALE	43
COME INIZIARE CON FIRMWARE SERIALE	46
PROGRAMMI DEMO PER FIRMWARE SERIALE	47
FIRMWARE DI LIBRERIA .LIB	48
INTEGRAZIONE ED USO DEL FIRMWARE DI LIBRERIA	48
RISORSE USATE DA FIRMWARE DI LIBRERIA	52
PROGRAMMAZIONE FLASH EPROM	54
AMBIENTI DI SVILUPPO PER FIRMWARE DI LIBRERIA	54
COME INIZIARE CON FIRMWARE DI LIBRERIA	56
PROGRAMMI DEMO PER FIRMWARE DI LIBRERIA	60
COMANDI	61
COMANDI PER IL POSIZIONAMENTO DEL CURSORE	61
CURSORE A SINISTRA	61
CURSORE A DESTRA	61
CURSORE IN BASSO	61
CURSORE IN ALTO	62
CURSORE A INIZIO	62
RITORNO A CAPO RIGA	62
RITORNO A CAPO + NUOVA RIGA	62
POSIZIONAMENTO ASSOLUTO DEL CURSORE ALFANUMERICO	62
COMANDI PER LA CANCELLAZIONE DEI CARATTERI	63
SPAZIO INDIETRO	63
CANCELLA PAGINA	63
CANCELLA RIGA	63
CANCELLA FINO A FINE RIGA	63
CANCELLA FINO A FINE PAGINA	63
COMANDI PER LA GESTIONE DELLA EEPROM	64
RICHIESTA DISPONIBILITA' AD USARE EEPROM	64
SCRITTURA DEL BYTE DI PRESENZA	64
LETTURA DEL BYTE DI PRESENZA	64
SCRITTURA BYTE SU EEPROM	65
LETTURA BYTE DA EEPROM	65
COMANDI PER FUNZIONI VARIE	66
LETTURA DEL NUMERO DI VERSIONE	66
LETTURA DEL CODICE SCHEDA	66
SETTAGGIO LUMINOSITÀ DISPLAY FLUORESCENTE	66
SETTAGGIO MODALITÀ OPERATIVA	67
RESET DELLA COMUNICAZIONE	67
RESET GENERALE	67
ATTIVAZIONE TEMPORIZZATA DEL BUZZER	68
ATTIVAZIONE LED E BUZZER	68
COMANDI PER LA GESTIONE DEI MESSAGGI	69
LETTURA NUMERO DELL'ULTIMO MESSAGGIO MEMORIZZABILE	69
MEMORIZZAZIONE DI UN MESSAGGIO	69
LETTURA DI UN MESSAGGIO	70

VISUALIZZAZIONE DI MESSAGGI	70
VISUALIZZAZIONE DI MESSAGGI A SCORRIMENTO	71
COMANDI PER GESTIONE DELLA TASTIERA	72
RICONFIGURAZIONE DI UN TASTO	72
ATTIVAZIONE KEYCLICK SENZA MEMORIZZAZIONE	72
DISATTIVAZIONE KEYCLICK SENZA MEMORIZZAZIONE	72
ATTIVAZIONE KEYCLICK CON MEMORIZZAZIONE	73
DISATTIVAZIONE KEYCLICK CON MEMORIZZAZIONE	73
COMANDI PER CARATTERI UTENTE	74
DEFINIZIONE DI UN CARATTERE UTENTE	75
DEFINIZIONE E SALVATAGGIO DI UN CARATTERE UTENTE	75
COMANDI PER GESTIONE DEGLI ATTRIBUTI DEL CURSORE	76
DISATTIVAZIONE DEL CURSORE	76
ATTIVAZIONE DEL CURSORE FISSO	76
ATTIVAZIONE DEL CURSORE "BLOCCO" LAMPEGGIANTE	76
COMANDI PER GESTIONE I/O DIGITALI BUFFERATI	77
SCRITTURA DI TUTTE LE USCITE DIGITALI	77
LETTURA DI TUTTI GLI INGRESSI DIGITALI	77
ATTIVAZIONE SINGOLA USCITA DIGITALE	78
ATTIVAZIONE SINGOLA USCITA DIGITALE TEMPORIZZATA	78
DISATTIVAZIONE SINGOLA USCITA DIGITALE	79
DISATTIVAZIONE SINGOLA USCITA DIGITALE TEMPORIZZATA	79
ACQUISIZIONE SINGOLO INGRESSO DIGITALE	80
ACQUISIZIONE SINGOLO INGRESSO DIGITALE CON DEBOUCING	80
ONDA QUADRA TEMPORIZZATA DA 1 SU SINGOLA USCITA DIGITALE	81
ONDA QUADRA TEMPORIZZATA DA 0 SU SINGOLA USCITA DIGITALE	82
ONDA QUADRA SU SINGOLA USCITA DIGITALE	83
COMANDI PER GESTIONI ALTERNATIVE DEGLI I/O DIGITALI	84
SETTAGGIO DEI PIN MULTIFUNZIONE	84
LETTURA SETTAGGIO DEI PIN MULTIFUNZIONE	84
LETTURA CONTATORE	85
AZZERAMENTO CONTATORE	85
COMANDI PER GRAFICA	86
SETTAGGIO MODALITÀ ALFANUMERICA	86
SETTAGGIO MODALITÀ GRAFICA	88
POSIZIONAMENTO ASSOLUTO DEL CURSORE GRAFICO	88
SETTAGGIO AREA GRAFICA	88
SETTAGGIO FONT GRAFICO	92
COMANDI PER LA COMUNICAZIONE 1-WIRE	93
RESET LINEA 1-WIRE	94
SCRITTURA DI UN BIT SU LINEA 1-WIRE	94
LETTURA DI UN BIT DA LINEA 1-WIRE	94
SCRITTURA DI N BYTES SU LINEA 1-WIRE	95
LETTURA DI N BYTES DA LINEA 1-WIRE	95
"READ ROM" SU LINEA 1-WIRE	96
"SKIP ROM" SU LINEA 1-WIRE	96
"MATCH ROM" SU LINEA 1-WIRE	97
"ALARM SEARCH" SU LINEA 1-WIRE	98

SCHEDE ESTERNE	99
BIBLIOGRAFIA	102
APPENDICE A: TABELLE RIASSUNTIVE COMANDI.....	A-1
APPENDICE B: CARATTERI DEI DISPLAY	B-1
APPENDICE C: NOTE PER IL MONTAGGIO	C-1
QUOTE DEL TERMINALE	C-1
MONTAGGIO MODALITÀ AVANQUADRO	C-2
INSERIMENTO ETICHETTA DI PERSONALIZZAZIONE	C-3
FISSAGGIO DEL PANNELLO FRONTALE AL CONTENITORE	C-4
APPENDICE D: DATI TECNICI	D-1
T89C51CC01 E T89C51AC2	D-2
APPENDICE E: INDICE ANALITICO	E-1

INDICE DELLE FIGURE

FIGURA 1: SCHEMA A BLOCCHI	3
FIGURA 2: FOTO DEI VARI MODELLI DISPONIBILI	5
FIGURA 3: FOTO CON VISTA COMPLESSIVA	9
FIGURA 4: TABELLA DEI CONSUMI	12
FIGURA 5: FOTO QTP 12/R84-C2	13
FIGURA 6: FOTO QTP 12/R84-F2	13
FIGURA 7: FOTO QTP 12/R84-GF2	13
FIGURA 8: CN1 - CONNETTORE PER ALIMENTAZIONE	14
FIGURA 9: COLLEGAMENTO ALIMENTAZIONE ALTERNATA $8\div 24\text{ VAC} \pm 5\%$	15
FIGURA 10: COLLEGAMENTO ALIMENTAZIONE CONTINUA $+10\div 38\text{ VDC} \pm 5\%$	15
FIGURA 11: COLLEGAMENTO ALIMENTAZIONE STABILIZZATA $+5\text{ VDC} \pm 5\%$	15
FIGURA 12: CN2 - CONNETTORE PER LINEA SERIALE	16
FIGURA 13: SCHEMA DI COMUNICAZIONE SERIALE	17
FIGURA 14: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 232	18
FIGURA 15: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 422	18
FIGURA 16: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 485	18
FIGURA 17: ESEMPIO COLLEGAMENTO IN RETE IN RS 485	19
FIGURA 18: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI	20
FIGURA 19: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI	20
FIGURA 20: ESEMPIO DI COLLEGAMENTO IN RETE IN CURRENT LOOP	21
FIGURA 21: CN3 - CONNETTORE PER LINEA CAN	22
FIGURA 22: COLLEGAMENTO LINEA CAN	22
FIGURA 23: ESEMPIO COLLEGAMENTO IN RETE CAN	23
FIGURA 24: CN4 - CONNETTORE PER USCITE DIGITALI A RELÈ	24
FIGURA 25: COLLEGAMENTO USCITE A RELÉ	24
FIGURA 26: SCHEMA DELLE USCITE A RELÉ	25
FIGURA 27: CN5 - CONNETTORE PER INGRESSI DIGITALI OPTOISOLATI	26
FIGURA 28: SCHEMA DEGLI INGRESSI OPTOISOLATI	27
FIGURA 29: COLLEGAMENTO INGRESSI OPTOISOLATI	27
FIGURA 30: J6 - CONNETTORE PER I/O TTL, 1-WIRE, I2C BUS	28
FIGURA 31: SCHEMA DEGLI I/O TTL, 1-WIRE, I2C BUS	28
FIGURA 32: DISPOSIZIONE JUMPERS, CONNETTORI, TRIMMER, ECC.	29
FIGURA 33: TABELLA JUMPERS	30
FIGURA 34: PIANTE COMPONENTI LATO STAGNATURE	31
FIGURA 35: PIANTE COMPONENTI LATO COMPONENTI	31
FIGURA 36: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE	33
FIGURA 37: FOTO ALIMENTATORE EXPS-2	37
FIGURA 38: FOTO CARATTERI DISPONIBILI SU QTP 12/R84-GF2	39
FIGURA 39: NUMERAZIONE E DISPOSIZIONE TASTI	41
FIGURA 40: CODICI DI DEFAULT DEI TASTI	41
FIGURA 41: ESEMPIO COMUNICAZIONE CON PROTOCOLLO MASTER SLAVE	45
FIGURA 42: COLLEGAMENTO RS 232 CON P.C. E FIRMWARE SERIALE	46
FIGURA 43: ORGANIZZAZIONE AREA CODICE CON FIRMWARE DI LIBRERIA	49
FIGURA 44: USO RAM CON FIRMWARE DI LIBRERIA	51
FIGURA 45: MODALITÀ SVILUPPO FIRMWARE DI LIBRERIA	55
FIGURA 46: COLLEGAMENTO RS 232 CON P.C. E FIRMWARE DI LIBRERIA	56

FIGURA 47: FINESTRA SETTAGGIO FLIP (1 DI 4)	57
FIGURA 48: FINESTRA SETTAGGIO FLIP (2 DI 4)	58
FIGURA 49: FINESTRA SETTAGGIO FLIP (3 DI 4)	58
FIGURA 50: FINESTRA SETTAGGIO FLIP (4 DI 4)	59
FIGURA 51: FOTO DEL FRONTALE CON TASTIERA	73
FIGURA 52: MODELLO DEI CARATTERI UTENTE	74
FIGURA 53: COMANDO ATTIVAZIONE SINGOLA USCITA TEMPORIZZATA	78
FIGURA 54: COMANDO DISATTIVAZIONE SINGOLA USCITA TEMPORIZZATA	79
FIGURA 55: COMANDO ONDA QUADRA TEMPORIZZATA DA 1	81
FIGURA 56: COMANDO ONDA QUADRA TEMPORIZZATA DA 0	82
FIGURA 57: COMANDO ONDA QUADRA	83
FIGURA 58: COORDINATE PUNTI DISPLAY GRAFICO	86
FIGURA 59: PRIMO ESEMPIO DI RAPPRESENTAZIONE GRAFICA	87
FIGURA 60: SECONDO ESEMPIO DI RAPPRESENTAZIONE GRAFICA	87
FIGURA 61: ESEMPIO DISEGNO AREA GRAFICA	89
FIGURA 62: DATI GRAFICI ORIZZONTALI CON SPOSTAMENTO ORIZZONTALE	90
FIGURA 63: DATI GRAFICI ORIZZONTALI CON SPOSTAMENTO VERTICALE	90
FIGURA 64: DATI GRAFICI VERTICALI CON SPOSTAMENTO ORIZZONTALE	91
FIGURA 65: DATI GRAFICI VERTICALI CON SPOSTAMENTO VERTICALE	91
FIGURA 66: ESEMPIO DI COLLEGAMENTO CON DUE DISPOSITIVI 1-WIRE	93
FIGURA 67: SCHEMA FUNZIONALE PROTOCOLLO 1-WIRE	97
FIGURA 68: SCHEMA DELLE POSSIBILI CONNESSIONI	101
FIGURA A1: TABELLA CODICI DEI COMANDI (1 DI 4)	A-1
FIGURA A2: TABELLA CODICI DEI COMANDI (2 DI 4)	A-2
FIGURA A3: TABELLA CODICI DEI COMANDI (3 DI 4)	A-3
FIGURA A4: TABELLA CODICI DEI COMANDI (4 DI 4)	A-4
FIGURA B1: TABELLA CARATTERI QTP 12/R84-F2, GF2 IN MODALITÀ ALFANUMERICA	B-1
FIGURA B2: TABELLA CARATTERI QTP 12/R84-C2	B-2
FIGURA B3: TABELLA CARATTERI QTP 12/R84-GF2 IN MODALITÀ GRAFICA CON MINIFONT	B-3
FIGURA B4: TABELLA CARATTERI QTP 12/R84-GF2 IN MODALITÀ GRAFICA CON FONT KATAKA..	B-4
FIGURA B5: TABELLA CARATTERI QTP 12/R84-GF2 IN MODALITÀ GRAFICA CON FONT EUROPEO	B-5
FIGURA C1: QUOTE QTP 12/R84	C-1
FIGURA C2: SCASSO DI MONTAGGIO	C-2
FIGURA C3: DIMENSIONI DELL'ETICHETTA DI PERSONALIZZAZIONE	C-3
FIGURA C4: INSERIMENTO ETICHETTA DI PERSONALIZZAZIONE	C-3

INTRODUZIONE

L'uso di questi dispositivi è rivolto - **IN VIA ESCLUSIVA** - a personale specializzato.

Questo prodotto non è un componente di sicurezza così come definito dalla direttiva **98-37/CE**.



I pin della scheda non sono dotati di protezione contro le cariche elettrostatiche. Visto che esiste un collegamento diretto tra numerosi pin della scheda ed i rispettivi pin dei componenti di bordo e che quest'ultimi sono sensibili ai fenomeni ESD, il personale che maneggia la scheda è invitato a prendere tutte le precauzioni necessarie per evitare i possibili danni che potrebbero derivare dalle cariche elettrostatiche.

Scopo di questo manuale è la trasmissione delle informazioni necessarie all'uso competente e sicuro dei prodotti. Esse sono il frutto di un'elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

I dati di seguito riportati sono destinati - **IN VIA ESCLUSIVA** - ad un utenza specializzata, in grado di interagire con i prodotti in condizioni di sicurezza per le persone, per la macchina e per l'ambiente, interpretando un'elementare diagnostica dei guasti e delle condizioni di funzionamento anomale e compiendo semplici operazioni di verifica funzionale, nel pieno rispetto delle norme di sicurezza e salute vigenti.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall'ASSISTENZA TECNICA AUTORIZZATA, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

I dispositivi non possono essere utilizzati all'aperto. Si deve sempre provvedere ad inserire i moduli all'interno di un contenitore a norme di sicurezza che rispetti le vigenti normative. La protezione di questo contenitore non si deve limitare ai soli agenti atmosferici, bensì anche a quelli meccanici, elettrici, magnetici, ecc.

Per un corretto rapporto coi prodotti, è necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare direttamente l'Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell'uso di tali dispositivi, è conveniente che l'utente - **PRIMA DI COMINCIARE AD OPERARE** - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all'indice generale e all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

VERSIONE HARDWARE, FIRMWARE

Il presente manuale é riferito alla versione **100403** di stampato ed alla versione **1.3** di firmware ed alle eventuali versioni successive. La validità delle informazioni riportate é quindi subordinata ai numeri di versione del terminale in uso e l'utente deve quindi sempre verificarne la giusta corrispondenza. La versione stampato è riportata all'interno del dispositivo in più punti (ad esempio sopra al connettore seriale CN2, sul lato componenti) così come la versione del firmware che é invece riportata in un'etichetta posta sul microprocessore. Inoltre la versione del firmware può essere richiesta direttamente al terminale tramite un'apposito comando.

INFORMAZIONI GENERALI

La **QTP 12/R84** é fondamentalmente un modulo di controllo completa di ingressi, uscite ed interfaccia operatore. In un ingombro minimo essa contiene un completo controllore PLC in grado di gestire completamente una specifica applicazione civile e/o industriale. La **QTP 12/R84** é disponibile con display alfanumerico da 20 caratteri per 2 righe, del tipo LCD retroilluminato con piano luminoso a LED, o fluorescente oppure con display grafico da 140 per 16 punti. A completamento del frontale del pannello operatore c'è una tastiera da 12 tasti a membrana, una tasca di personalizzazione che permette di attribuire facilmente un nome, o di inserire il logo dell'utente, ed un LED di segnalazione.

Un pratico e robusto contenitore metallico nel formato standard DIN 72x144 consente di installare il terminale in modalità avanquadro, con un frontale protetto e con uno scasso posteriore con i connettori utilizzabili per i vari collegamenti. Le staffe fornite assieme alla **QTP 12/R84** consentono di montarla e/o smontarla praticando un semplice scasso rettangolare sul pannello di supporto. In alternativa può essere usata anche su tavolo senza dover aggiungere alcun elemento.

La presenza di una linea seriale che può essere bufferata con i più diffusi standard elettrici di comunicazione consente di collegare la **QTP 12/R84** alla maggioranza dei sistemi presenti sul mercato e di realizzare anche delle economiche reti in cui si possono gestire contemporaneamente numerose unità. L'interconnettività con altri dispositivi é inoltre garantita dalla linea CAN opzionale, che ne amplia i possibili settori di impiego ed allo stesso tempo migliora le prestazioni complessive della rete.

La sezione del microcontrollore, dotata di una FLASH da 32K, ha un'interfaccia ISP che le consente di essere programmata direttamente tramite la linea seriale; in questo modo la fase di sviluppo é notevolmente semplificata e non é necessario alcun sistema aggiuntivo, con un conseguente risparmio economico. L'utente può sviluppare il programma applicativo, con codice compatibile 8051, tramite vari linguaggi ad alto livello tra cui l'economico compilatore BASIC BASCOM 8051; l'efficiente compilatore C µC/51 oppure, per gli amanti dei linguaggi a contatti, con il noto LadderWORK. Al fine di semplificare ulteriormente tale sviluppo sono disponibili diversi firmware di gestione pronti all'uso, forniti in abbinamento al prodotto.

Le caratteristiche hardware della **QTP 12/R84**, comprensiva delle varie opzioni, sono le seguenti:

- Ingombri: frontale **72x144 mm**; anteriore **9 mm**; posteriore **28 mm**
- Contenitore in **alluminio** con cornice anteriore in **plastica**
- Pannello anteriore con tastiera e mascherina in **poliestere antigraffio**
- Contenitore completo di **staffe** di montaggio
- Montaggio su pannello in modalità **avanquadro**
- **Protezione** anteriore del pannello: **IP-54**

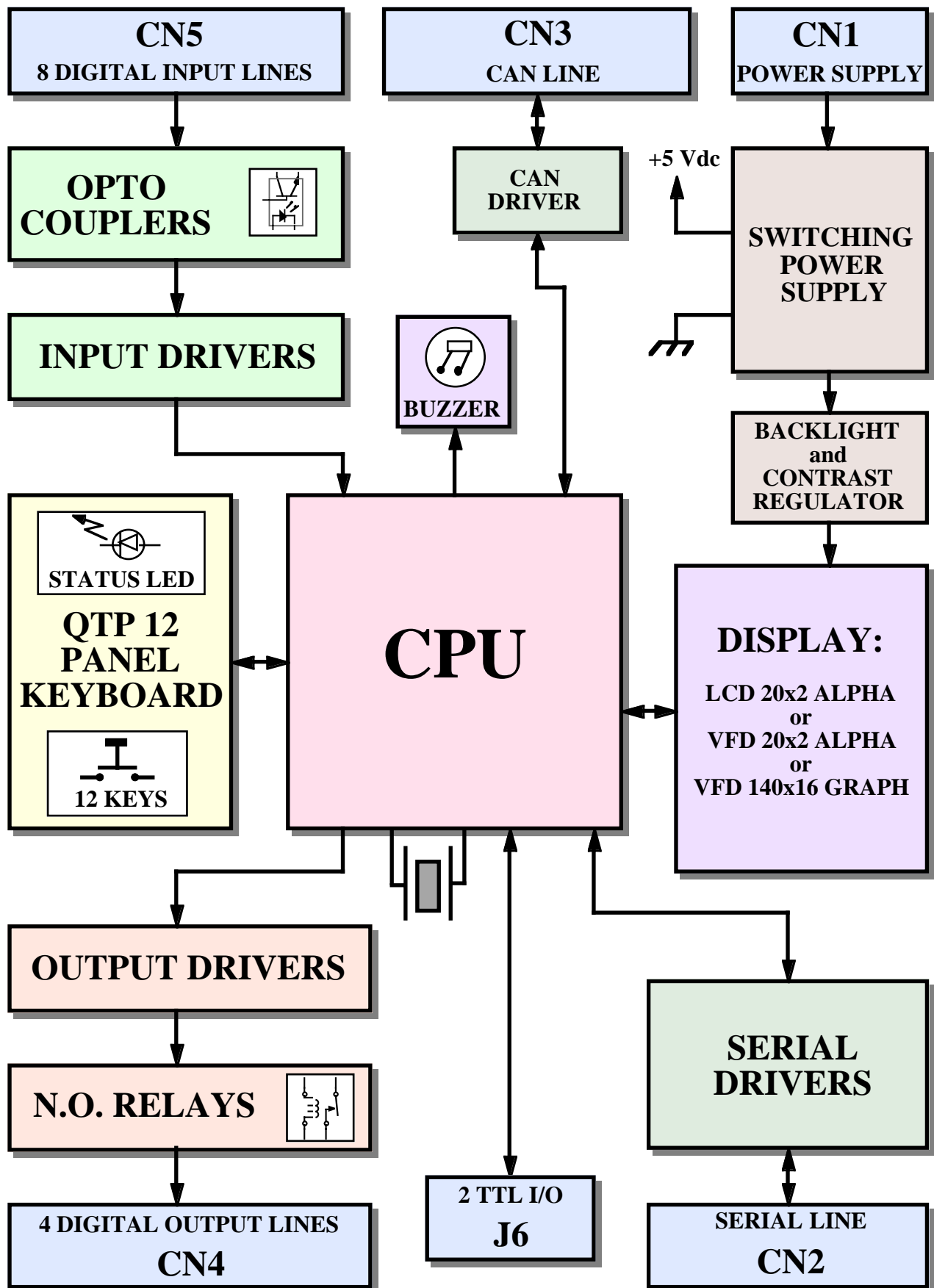


FIGURA 1: SCHEMA A BLOCCHI

- Il pannello operatore é disponibile con i seguenti 3 **modelli** di display:
 - QTP 12/R84-C2**: display LCD alfanumerico da 2 righe per 20 car. retroilluminato
 - QTP 12/R84-F2**: display Fluorescente alfanumerico da 2 righe per 20 caratteri
 - QTP 12/R84-GF2**: display Fluorescente grafico da 140 per 16 punti
- Dimensione **area visibile**: 82x18 mm
- Tastiera a membrana da **12 tasti** con doppia serigrafia: numerica e funzionale
- **LED** di segnalazione comandabile da software, con diverse modalità
- **Tasca** per la **personalizzazione** del nome del dispositivo e/o del LED di segnalazione
- **Buzzer** per segnalazioni acustiche, attivabile via software
- **Microprocessore** famiglia I51, con clock da **14,7** o **29,4 MHz** selezionabile via software
- Tre diversi tipi di memorie: **32K FLASH EPROM**; **1,2K RAM**; **2K EEPROM**
- Linea seriale configurabile in **RS 232**, **RS 422**, **RS 485**, **Current loop**
- Linea di comunicazione **CAN** dotata di apposito driver di linea
- **8 ingressi** optoisolati che possono essere indifferentemente **NPN** o **PNP**
- Alcuni ingressi sono collegati a linee di conteggio ed **interrupt** hardware
- **4 uscite** a relé da **5 A**
- **2 linee di I/O TTL** su apposito connettore, disponibile per gestione di dispositivi **I2C BUS**, **1-WIRE**, **SPI**, ecc
- Comodi connettori a **morsettiera a rapida estrazione**, per un facile cablaggio con il campo
- Alimentazione ad ampio range, in **DC** o in **AC** a partire da **5Vdc**, fino a **24Vac**
- **Potenza** complessiva, in funzione della configurazione usata, tra **2,5** a **3,3 W**
- Protezione della logica di bordo tramite **TransZorb™**
- Vasta disponibilità di linguaggi ed ambienti evoluti per lo sviluppo del firmware di gestione come: **assemblatori**; **FORTH**; compilatori **PASCAL** (SYS51PW); compilatori **C** (HTC 51, SYS51CW, DDS MICRO C 51, µC/51); **sistemi operativi real time** (CMX RTX); programmazione a **contatti** (LADDER WORK); compilatori **BASIC** (BXC51, BASCOM 8051); emulazioni terminale (GET 51), ecc.
- Fornita con due diversi **firmware** di gestione: **seriale** o **libreria**
- Possibilità di esecuzioni dedicate di pannello e programmi
- Per specifiche esigenze di consumi e costo il display LCD può essere non retroilluminato (contattare la **grifo®**)

Viene di seguito riportata una descrizione dei blocchi funzionali della scheda, con indicate le operazioni effettuate da ciascuno di essi. Per una più facile individuazione di tali blocchi e per una verifica delle loro connessioni, fare riferimento alla figura 1.

LINEA SERIALE

La comunicazione con le altre unità avviene tramite una linea seriale asincrona, che dal punto di vista elettrico é normalmente bufferata in **RS 232** ma che in fase di ordine, può essere configurata in:

RS 422	->	opzione .RS422
RS 485	->	opzione .RS485
Current Loop	->	opzione .CLOOP

Sia il protocollo fisico di comunicazione, che quello logico, sono completamente gestiti via firmware come descritto negli appositi paragrafi del capitolo DESCRIZIONE SOFTWARE.

Le molteplici configurazioni della linea seriale aprono numerose possibilità di connessione ed espansione del modulo, compreso la realizzazione di economiche reti di comunicazione del tipo master slave.

QTP 12/R84-GF2



QTP 12/R84-C2



QTP 12/R84-F2



FIGURA 2: FOTO DEI VARI MODELLI DISPONIBILI

LED DI SEGNALAZIONE

La **QTP 12/R84** dispone di **1 LED** per segnalazioni visive di vario tipo, che é posizionato a fianco della tasca di personalizzazione in modo da poterlo specializzare. La funzione principale di questo LED é quella di fornire un'indicazione visiva dello stato del sistema, facilitando le operazioni di verifica funzionamento della macchina anche ad una distanza che non consente la lettura del display. Per questa ragione i possibili tre stati (attivo, disattivo, lampeggiante) possono segnalare diverse situazioni operative come ad esempio un allarme, un'allerta, un pronto macchina, ecc.

Per una descrizione dei firmware che gestiscono il LED di segnalazione si faccia riferimento agli appositi paragrafi del capitolo COMANDI.

DISPLAY

La **QTP 12/R84** é disponibile con tre diversi display: **grafico fluorescente** da **140x16** punti, **alfanumerico fluorescente** da **20x2** caratteri, **alfanumerico LCD** da **20x2** caratteri retroilluminato. La retroilluminazione a LED garantisce una buona visibilità anche in condizioni di luce ambientale variabile ed in caso di necessità l'utente può comunque intervenire su un trimmer di regolazione del contrasto. Un'altra caratteristica di fondamentale importanza per i display della **QTP 12/R84** é il loro ampio angolo di visione che ne consente la lettura praticamente da ogni posizione frontale. Per informazioni più dettagliate sui display fare riferimento al capitolo SPECIFICHE TECNICHE.

Come ampiamente descritto nel capitolo che descrive i comandi riconosciuti, si ricorda che il modello con display grafico (**QTP 12/R84-GF2**) é in grado di eseguire tutti i comandi (sia grafici che alfanumerici) mentre i modelli con display alfanumerico (**QTP 12/R84-C2** e **QTP 12/R84-F2**) non possono naturalmente eseguire i comandi grafici.

Se il numero di caratteri rappresentabili é insufficiente si ricorda che, come descritto nel capitolo COMANDI, i firmware disponibili gestiscono la rappresentazione di messaggi anche in modalità a scorrimento; in questo modo é possibile visualizzare in un'unica riga del display delle informazioni che occupano uno spazio maggiore di quello normalmente rappresentabile.

L'utente deve quindi scegliere il display (e quindi il modello di **QTP 12/R84**) che sia sufficiente per il numero ed il tipo di informazioni da rappresentare e che soddisfi le sue esigenze di visibilità. In caso di particolari esigenze di consumi ridotti, visibilità, e costo si possono avere anche il display LCD senza retroilluminazione: per maggiori informazioni su questa possibilità e sulla sua disponibilità contattare direttamente la **grifo®**.

TASTIERA

La **QTP 12/R84** dispone di una tastiera a membrana a **12 tasti**, disposti attorno al display che risolvono economicamente il problema dell'inserimento dati, anche quando questi sono articolati ed eterogenei. Tutti i tasti sono del tipo a cupola metallica e forniscono quindi una sensazione tattile dell'avvenuta pressione e garantiscono una lunga durata anche in condizioni d'uso gravose. Tutti i tasti sono provvisti di una serigrafia standard (illustrata nella figura 49) che soddisfa le normali esigenze di interfacciamento uomo macchina, nel settore industriale. In particolare la presenza sia delle cifre numeriche che dell'intero alfabeto e di alcune funzioni consente di inserire qualsiasi dato e/o effettuare qualsiasi comando.

Si ricorda che in caso di display grafico i 4 tasti centrali possono assumere la cosiddetta funzionabilità variabile, ovvero via software si può disegnare la funzione dei tasti in prossimità degli stessi e e quindi variarla a seconda delle necessità (es. START, STOP, INS, DEL, ALRM, INFO, ecc.).

Infine una tasca adibita all'inserimento di una etichetta realizzata dall'utente può essere utilizzata per personalizzare o identificare la funzione del terminale come descritto in APPENDICE C.

La gestione della tastiera é completamente effettuata via firmware come descritto negli appositi paragrafi dei capitoli DESCRIZIONE SOFTWARE e COMANDI.

BUZZER

La **QTP 12/R84** dispone di una circuiteria in grado di emettere un suono costante basata su un buzzer capacitivo. Via software, tramite apposite modalità illustrate nei capitoli DESCRIZIONE SOFTWARE e COMANDI, questa può essere disattivata, attivata od intermittente, può generare un beep sonoro, può essere abbinata alla pressione di un tasto per avere la funzione di **keyclick** ed infine può segnalare eventuali anomalie di funzionamento.

Quando a seguito di un'accensione, la scheda genera continuamente un suono costante o intermittente e non lavora più correttamente, c'è una condizione anomala che deve essere eliminata: contattare direttamente i tecnici **grifo®**.

MEMORIE

La **QTP 12/R84** dispone di tre diversi tipi di memorie:

32K bytes di **FLASH** per il codice,

256 bytes di **RAM** interna per dati e stack ed **1K** bytes di **ERAM** esterna per dati,

2K bytes di **EEPROM** per dati

che possono essere utilizzate dall'utente a seconda delle sue esigenze. La configurazione delle memorie presenti sulla scheda é fissa e non può essere modificata dall'utente; qualora risultasse insufficiente si consiglia di contattare direttamente la **grifo®**.

Di particolare interesse ed utilità é la EEPROM che é in grado di mantenere i dati anche in assenza di alimentazione. Questa caratteristica fornisce alla scheda la possibilità di ricordare in ogni condizione, una serie di parametri come ad esempio la configurazione o lo stato del sistema.

Il mappaggio delle risorse di memoria avviene all'interno dello spazio d'indirizzamento del microprocessore in modo completamente automatico, in modo da adattarsi ai diversi pacchetti software disponibili per la **QTP 12/R84**.

I firmware di gestione sviluppati utilizzano tutte le memorie descritte per il loro normale funzionamento e vi allocano quindi buffers, contatori, stati, settaggi, codici dei caratteri, messaggi, caratteri utente, ecc. Per ulteriori informazioni vedere il capitolo DESCRIZIONE SOFTWARE.

ALIMENTATORE DI BORDO

Una delle caratteristiche fondamentali della **QTP 12/R84** é la presenza di un proprio alimentatore switching che richiede una tensione di alimentazione variabile nel range **8÷24 Vac** oppure **10÷38 Vdc ±5%**; tale sezione si occupa di generare tutte le tensioni necessarie al funzionamento del modulo. In alternativa si può ordinare la **QTP 12/R84** senza la sezione alimentatrice (specificando l'opzione **.5Vdc** o **.ALIM**) ed in questo caso si dovrà fornire un'alimentazione stabilizzata di +5 Vdc ±5% dall'esterno.

Per informazioni più dettagliate sulla sezione alimentatrice fare riferimento al paragrafo SPECIFICHE ELETTRICHE ed ALIMENTAZIONE.

PROCESSORE

La **QTP 12/R84** é predisposta per accettare i microcontrollori T89C51CC01 oppure T89C51AC2 prodotte dalla **ATMEL**. Tali processori ad 8 bits sono codice compatibile con la famiglia INTEL 8051, largamente diffusa a livello mondiale, e sono caratterizzati da: un esteso set di istruzioni, un'alta velocità di esecuzione e di manipolazione dati, da un'efficiente gestione degli interrupts e da una ricca serie di periferiche hardware integrate. Per maggiori informazioni si faccia riferimento all'apposita documentazione della casa costruttrice, oppure all'APPENDICE D di questo manuale.

INTERFACCIA CAN

La **QTP 12/R84** dispone sotto forma di opzione di una completa interfaccia CAN che supporta entrambi gli standard **BasicCAN** e **PeliCAN 2.0B**. Con questa possibilità si possono affrontare e risolvere problemi di trasferimento dati ad alta velocità, comunicazione su lunghe distanze, gestione autonoma degli errori, supporto di reti multimaster e multislave, ecc.

Il codice per ordinare questa opzione é: **.CAN**

LINEE I/O TTL

Su un connettore della **QTP 12/R84** sono disponibili 2 linee di I/O digitale a livello TTL. La funzione di queste linee é completamente definibile dall'utente e grazie ai firmware di gestione sono disponibili anche funzionalità autonome dedicate a periferiche con standard **SPI**, **1-WIRE** ed **I2C BUS**. Grazie a queste interfacce possono essere collegati dispositivi dotati dello stesso standard di comunicazione in modo da espandere localmente le potenzialità del modulo. Il connettore é stato previsto in modo da consentire sia il collegamento esterno che interno al contenitore metallico, in modo da soddisfare ogni possibile esigenza dell'utilizzatore.

Una ricca serie di esempi software prevede la gestione delle più comuni e diffuse interfacce come A/D e D/A converter, display driver, memorie, sensori di temperatura chiavi elettroniche, ecc.

INGRESSI DIGITALI OPTOISOLATI

La scheda dispone di 8 ingressi di tipo NPN e/o PNP, collegati ad un comodo connettore a rapida estrazione che, tramite un'interfaccia galvanicamente isolata ed un apposito driver, vengono direttamente acquisite da otto linee di I/O del microcontrollore. Tali linee sono state scelte in modo da poter sfruttare appieno le periferiche interne, quindi gli stessi ingressi possono generare interrupts, essere contati dai contatori hardware, ecc. Gli ingressi optoisolati sono alimentati da un'apposita tensione definita +V opto che deve essere fornita dall'esterno.

USCITE DIGITALI A RELÉ

La scheda é dotata di 4 uscite a relé da 5A, con contatto normale aperto. Ogni linea é pilotata da linee di I/O del microcontrollore, é bufferata da un apposito driver ed é collegata ad un comodo connettore a rapida estrazione, che permette un facile interfacciamento con i segnali del campo. Anche in questo caso le linee del microcontrollore sono state scelte in modo da sfruttare le sue periferiche hardware interne.



FIGURA 3: FOTO CON VISTA COMPLESSIVA

SPECIFICHE TECNICHE

CARATTERISTICHE GENERALI

Risorse di bordo:	<p>Frontale IP54</p> <p>LED di stato gestibile via software</p> <p>Tastiera a membrana da 12 tasti, a cupola metallica</p> <p>Buzzer per segnalazioni sonore gestibile via software</p> <p>Seriale in RS 232, oppure RS 422, RS 485, Current Loop (opzione)</p> <p>Interfaccia CAN (opzione)</p> <p>Display alfanumerico o grafico in 3 diversi modelli</p> <p>Trimmer regolazione contrasto display LCD</p> <p>8 ingressi digitali optoisolati NPN e PNP</p> <p>4 uscite digitali bufferate con relé</p> <p>2 linee di I/O digitale TTL</p> <p>1 sezione alimentatrice switching</p>
Display:	<p>alfanumerico LCD 20x2 retroilluminato a LED</p> <p>alfanumerico Fluorescente 20x2</p> <p>grafico Fluorescente 140x16</p>
Microcontrollore:	T89C51AC2 oppure T89C51CC01
Memoria indirizzabile:	<p>32K x 8 FLASH EPROM</p> <p>256 x 8 RAM interna</p> <p>1K x 8 RAM esterna</p> <p>2K x 8 EEPROM</p>
Frequenza clock:	14.7456 MHz (raddoppiabile via software)
Frequenza taglio ingressi opto:	13 KHz
Durata reset dopo accensione:	160 msec
Tempo max scrittura EEPROM:	8 msec
Protocollo fisico comunicazione:	<p>Baud rate: 1200, 2400, 4800, 9600, 19200, 38400</p> <p>(#) Stop Bit: 1, 2</p> <p>Parità: nessuna</p> <p>Bit per carattere: 8, 9</p> <p>Default: 19200 Baud, 1 Stop, Nessuna parità, 8 Bit</p>
Protocollo logico com. (#):	Selezionabile tra normale e master slave (<i>Default: normale</i>)
Dimensioni buffer ricezione (#):	30 caratteri
Dimensioni buffer trasm. (#):	20 caratteri
Risoluzione temporizzazioni (#):	10 msec

Precisione temporizzazioni (#): 2,5 msec

Autorepeat tasti (#): Dopo 500 ms e poi ogni 100 ms

Numero messaggi (#): 95

Bytes EEPROM utente (#): 40

(#) = Caratteristica con firmware seriale o libreria.

CARATTERISTICHE FISICHE

Dimensioni esterne: DIN 72x144: 144 x 72 x 37 mm (L x A x P)
156 x 72 x 80 mm (L x A x P) con staffe
Vedere quote in APPENDICE C

Dimensioni scasso montaggio: 138 (min) x 66 (min) x 10 (max) mm (L x A x P)
Vedere quote in APPENDICE C

Dimensioni caratteri: LCD 20x2: 5x7 punti=3,20 x 4,85 mm (LxA)
Fluorescente 20x2: 5x7 punti=2,40 x 4,70 mm (LxA)
Fluorescente 140x16: da 5x7 punti=1,50 x 3,62 mm (LxA)
a 10x14 punti=5,00 x 10,15 mm (LxA)

Peso: 330 g massimi

Montaggio: Su pannello, in modalità avanquadro, tramite apposite staffe
A vista su piano d'appoggio.

Range di temperatura: Da 0 a 50 gradi centigradi

Umidità relativa: 20% fino a 90% (senza condensa)

Connettori: CN1: morsettiera a rapida estrazione, 2 vie, maschio, passo 3,5
CN2: vaschetta D a 9 vie, femmina
CN3: morsettiera a rapida estrazione, 3 vie, maschio, passo 3,5
CN4: morsettiera a rapida estrazione, 6 vie, maschio, passo 3,5
CN5: morsettiera a rapida estrazione, 9 vie, maschio, passo 3,5
J6: strip verticale, 4 vie, maschio, passo 2,54

CARATTERISTICHE ELETTRICHE

Tensione di alimentazione:	+10÷38 Vdc , 8÷24 Vac oppure +5 Vdc \pm 5% (*)
Consumo di alimentazione:	Vedere tabella successiva
Tensione alimentazione d'uscita:	+5,0 Vdc
Corrente disponibile su +5Vdc d'uscita:	400 mA - consumo max +5 Vdc (*) 1000 mA - consumo max +5 Vdc (solo .GF2)
Corrente massima su relé:	5A (carico resistivo)
Tensione massima su relé:	35 Vdc
Corrente per ingresso opto:	2÷9 mA
Tensione per ingressi optoisolati:	+V opto = 8÷30 Vdc (*)
Impedenza linea RS 422-485:	60 Ω
Rete terminazione RS 422-485:	Resistenza terminazione linea= 120 Ω Resistenza di pull up sul positivo= 3,3 K Ω Resistenza di pull down sul negativo= 3,3 K Ω
Impedenza linea CAN:	60 Ω
Rete terminazione CAN:	Resistenza da 120 Ω , disinseribile

(*) I dati riportati sono riferiti ad un lavoro a temperatura ambiente di 20 gradi centigradi (per ulteriori informazioni fare riferimento al paragrafo ALIMENTAZIONE).

Sono di seguito riportati i consumi relativi al terminale **QTP 12/R84** nelle versioni con i vari modelli di display installabili; per l'alimentazione ad ampio range sono riportate le potenze richieste, che tengono già conto del rendimento della sezione alimentatrice di bordo.

<i>Modello DISPLAY</i>	<i>Consumo max +5 Vdc \pm5%</i>	<i>Consumo max 10÷38 Vdc \pm5% 8÷24 Vac \pm5%</i>
LCD20x2 alfanumerico retroilluminato: QTP 12/R84-C2	380 mA	2,7 W
Fluorescente alfanumerico 20x2: QTP 12/R84-F2	360 mA	2,5 W
Fluorescente grafico 140x16: QTP 12/R84-GF2	475 mA	3,3 W

FIGURA 4: TABELLA DEI CONSUMI

Si ricorda che qualora sia necessario ridurre i consumi delle **QTP 12/R84** con display LCD si possono ordinare anche i display LCD **senza retroilluminazione**: per maggiori informazioni su questa possibilità e sulla sua disponibilità contattare direttamente la **grifo®**.



FIGURA 5: FOTO QTP 12/R84-C2



FIGURA 6: FOTO QTP 12/R84-F2

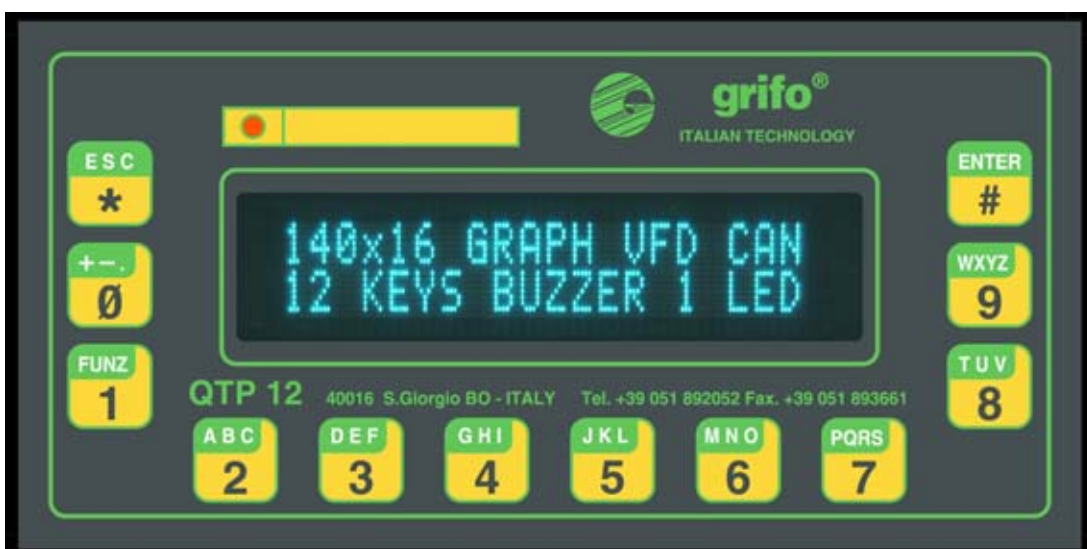


FIGURA 7: FOTO QTP 12/R84-GF2

INSTALLAZIONE

In questo capitolo saranno illustrate tutte le operazioni necessarie per utilizzare correttamente il terminale **QTP 12/R84**. A questo scopo viene riportata l'ubicazione e la funzione dei connettori, dei jumpers modificabili dall'utente, del trimmer ed ogni altra informazione relativa alla configurazione hardware del prodotto.

CONNESSIONI

Il modulo **QTP 12/R84** è provvisto di 6 connettori con cui vengono effettuati tutti i collegamenti con il campo e con le altre unità del sistema di controllo da realizzare. Di seguito viene riportato il loro pin out ed il significato dei segnali collegati; per una facile individuazione di tali connettori, si faccia riferimento alla figura 30, mentre per ulteriori informazioni a riguardo del tipo di connessioni, fare riferimento alle figure successive che illustrano il tipo di collegamento effettuato a bordo scheda. Si ricorda che i connettori sono accessibili sulla parte posteriore del modulo tramite un'apposito scasso sulla parete del contenitore, che ne consente l'inserimento e l'estrazione.

CN1 - CONNETTORE PER ALIMENTAZIONE

CN1 é un connettore a morsettiera, verticale, a rapida estrazione, a 2 vie, con passo 3,5 mm. Tramite CN1 deve essere fornita l'unica tensione di alimentazione per il terminale che può essere di tre diversi tipi, come descritto dalle figure seguenti:

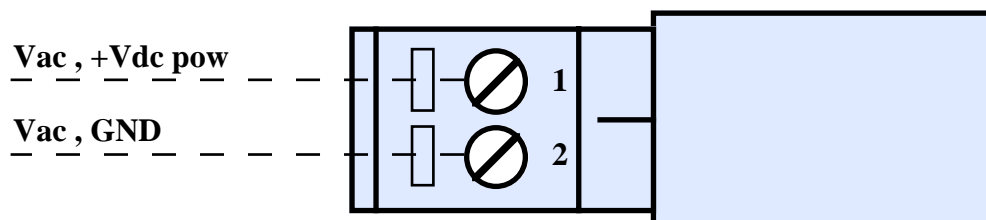


FIGURA 8: CN1 - CONNETTORE PER ALIMENTAZIONE

Legenda:

- Vac** = I - Linee di alimentazione alternata collegate alla sezione switching di bordo; tali segnali coincidono con una tensione da **8 ÷ 24 Vac ±5%**
- +Vdc pow** = - Linea di alimentazione continua, collegata alla sezione switching di bordo (**+10 ÷ +38 Vdc ±5%**) oppure tensione stabilizzata collegata alla logica di bordo (**+5 Vdc ±5%**), a seconda della configurazione ordinata
- GND** = - Linea di massa per alimentazione in continua.

N.B. Per ulteriori informazioni sull'alimentazione e le sue possibili configurazioni, fare riferimento al paragrafo ALIMENTAZIONE.

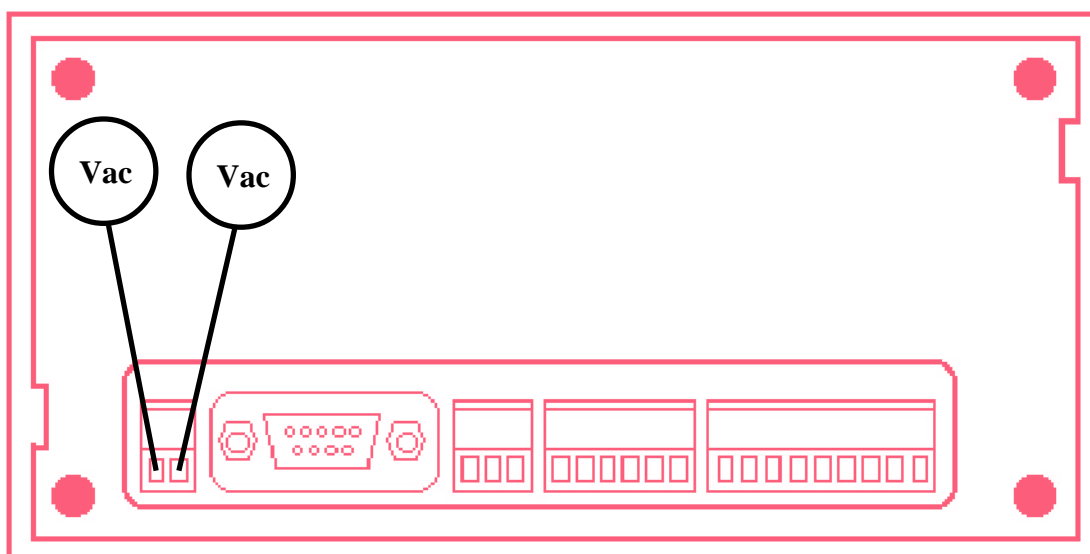


FIGURA 9: COLLEGAMENTO ALIMENTAZIONE ALTERNATA 8÷24 VAC $\pm 5\%$

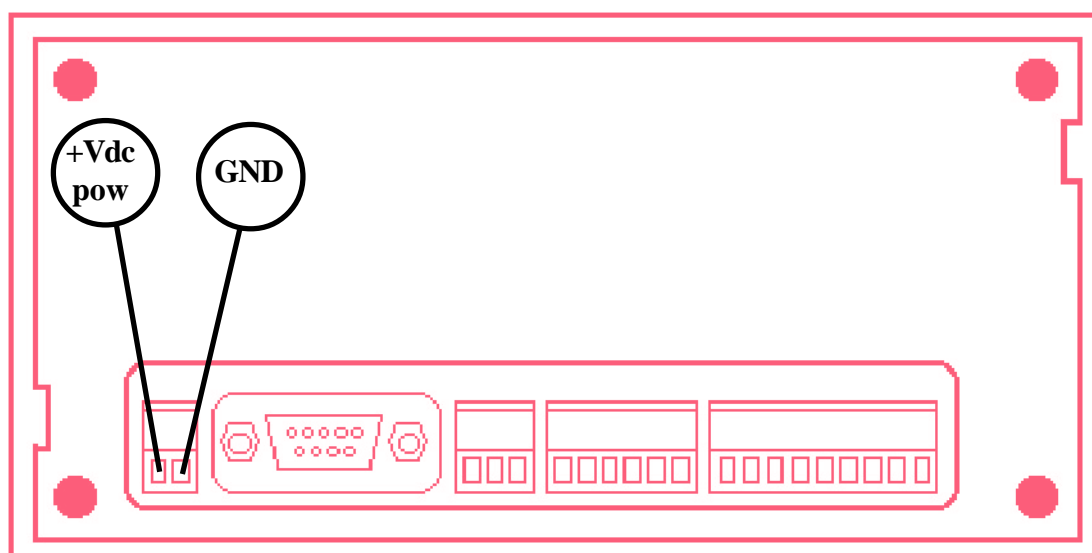


FIGURA 10: COLLEGAMENTO ALIMENTAZIONE CONTINUA +10÷38 VDC $\pm 5\%$

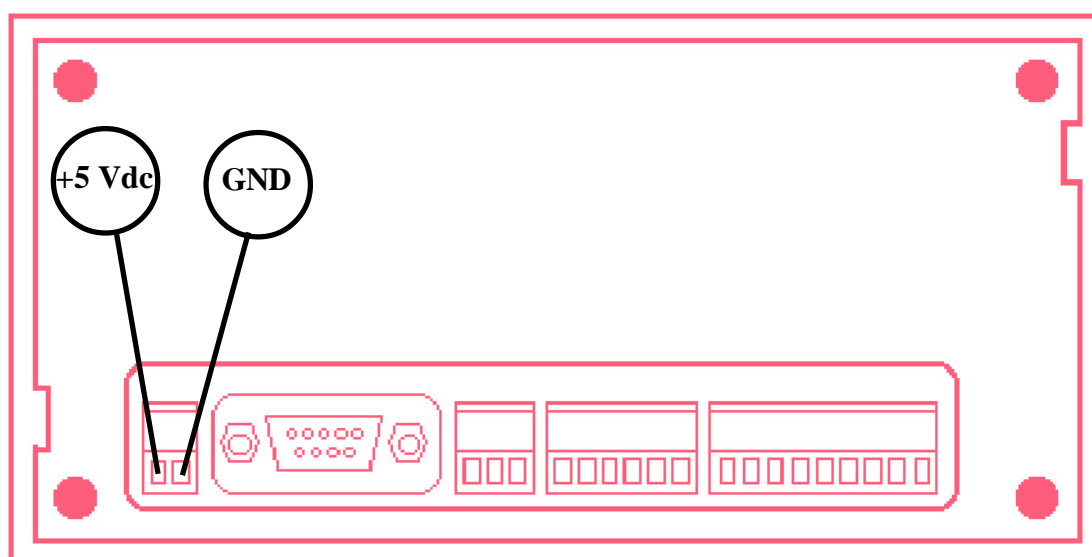


FIGURA 11: COLLEGAMENTO ALIMENTAZIONE STABILIZZATA +5 VDC $\pm 5\%$

CN2 - CONNETTORE PER LINEA SERIALE

CN2 é un connettore a vaschetta D, femmina, verticale, a 9 vie.

Sul connettore sono disponibili i segnali per la comunicazione della linea seriale, in RS 232, RS 422, RS 485 o Current loop. La disposizione dei segnali, riportata di seguito, é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo, mentre i segnali rispettano le normative definite dal CCITT relative allo standard utilizzato.

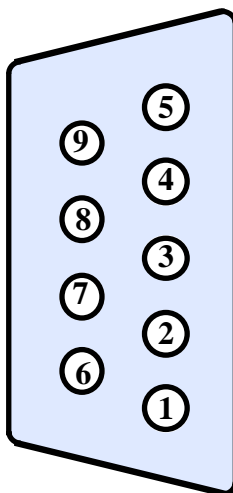


FIGURA 12: CN2 - CONNETTORE PER LINEA SERIALE

Pin	Segnale	Direzione	Descrizione
-----	---------	-----------	-------------

Linea seriale in RS 232:

2	RX RS232	= I	- Linea ricezione in RS 232.
3	TX RS232	= O	- Linea trasmissione in RS 232.
5	GND	=	- Linea di massa.

Linea seriale in RS 422:

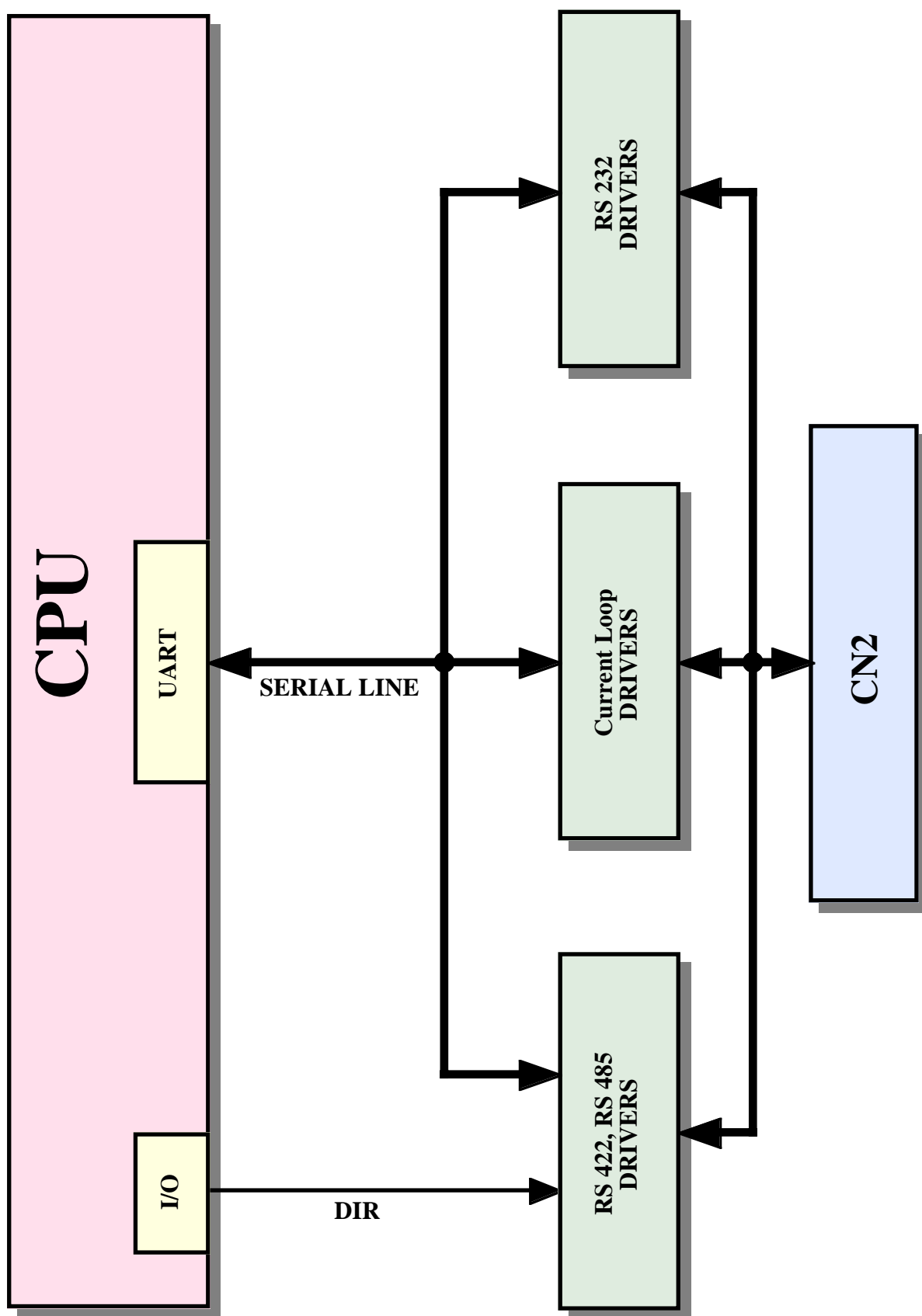
1	RX- RS422	= I	- Linea bipolare negativa di ricezione differenziale in RS 422.
2	RX+ RS422	= I	- Linea bipolare positiva di ricezione differenziale in RS 422.
3	TX- RS422	= O	- Linea bipolare negativa di trasmissione differenziale in RS 422.
4	TX+ RS422	= O	- Linea bipolare positiva di trasmissione differenziale in RS 422.
5	GND	=	- Linea di massa.

Linea seriale in RS 485:

1	RXTX- RS485	= I/O	- Linea bipolare negativa di ricezione e trasmissione differenziale in RS 485.
2	RXTX+ RS485	= I/O	- Linea bipolare positiva di ricezione e trasmissione differenziale in RS 485.
5	GND	=	- Linea di massa.

Linea seriale in Current loop:

9	RX- C.L.	= I	- Linea bipolare negativa di ricezione in Current loop.
8	RX+ C.L.	= I	- Linea bipolare positiva di ricezione in Current loop.
7	TX- C.L.	= O	- Linea bipolare negativa di trasmissione in Current loop.
6	TX+ C.L.	= O	- Linea bipolare positiva di trasmissione in Current loop.

**FIGURA 13: SCHEMA DI COMUNICAZIONE SERIALE**

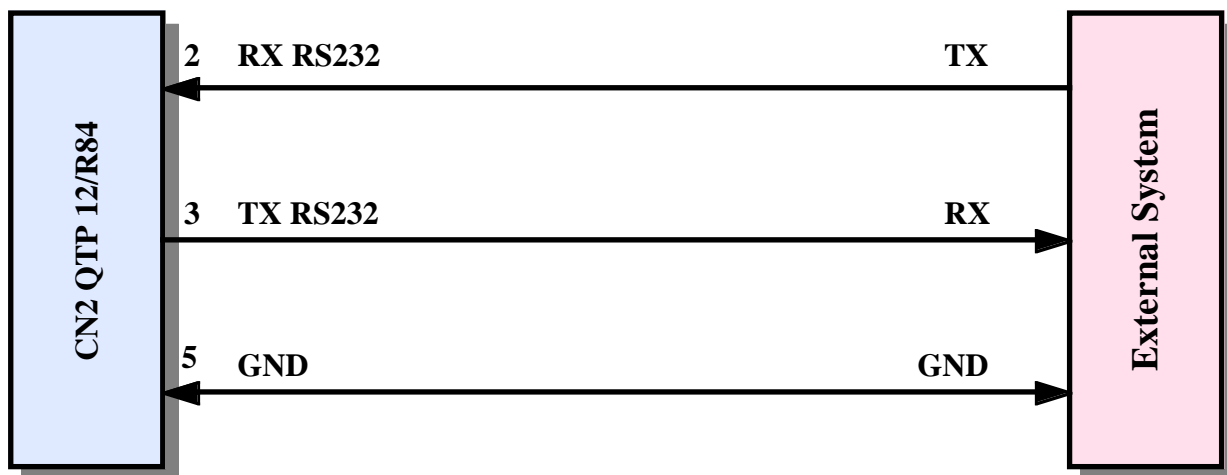


FIGURA 14: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 232

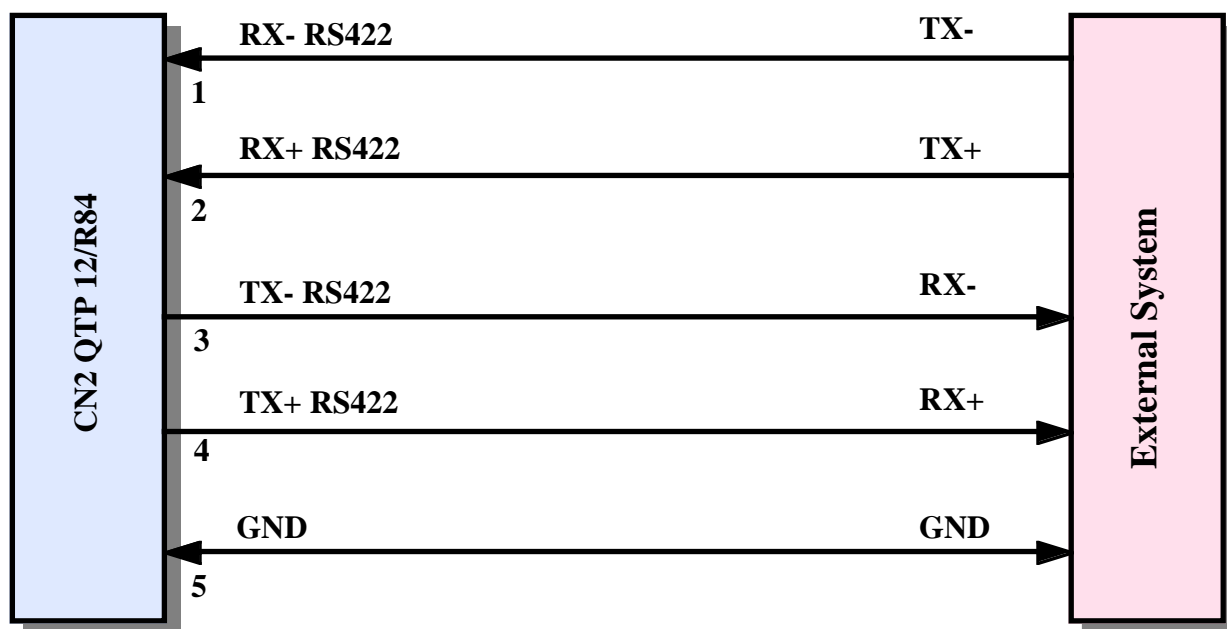


FIGURA 15: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 422

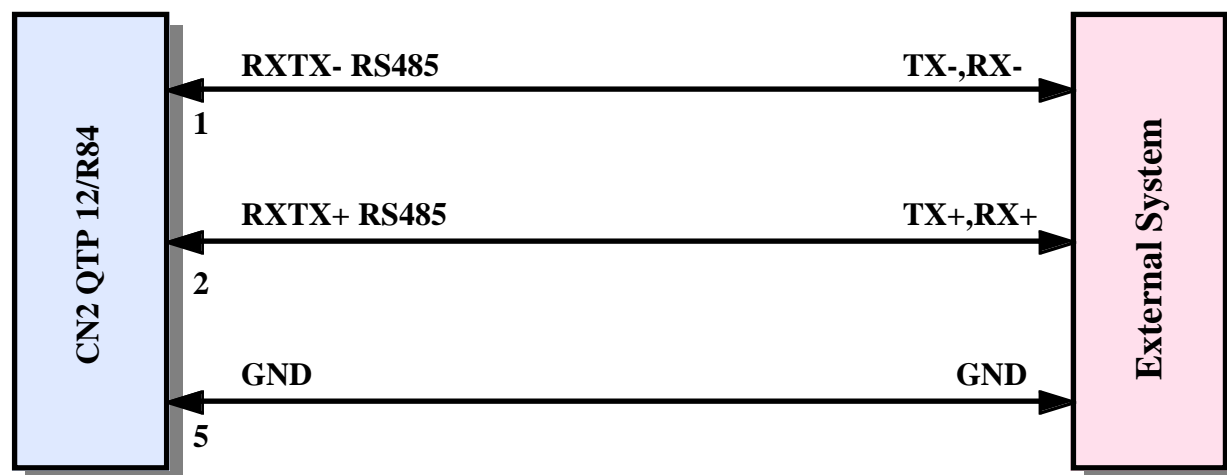


FIGURA 16: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 485

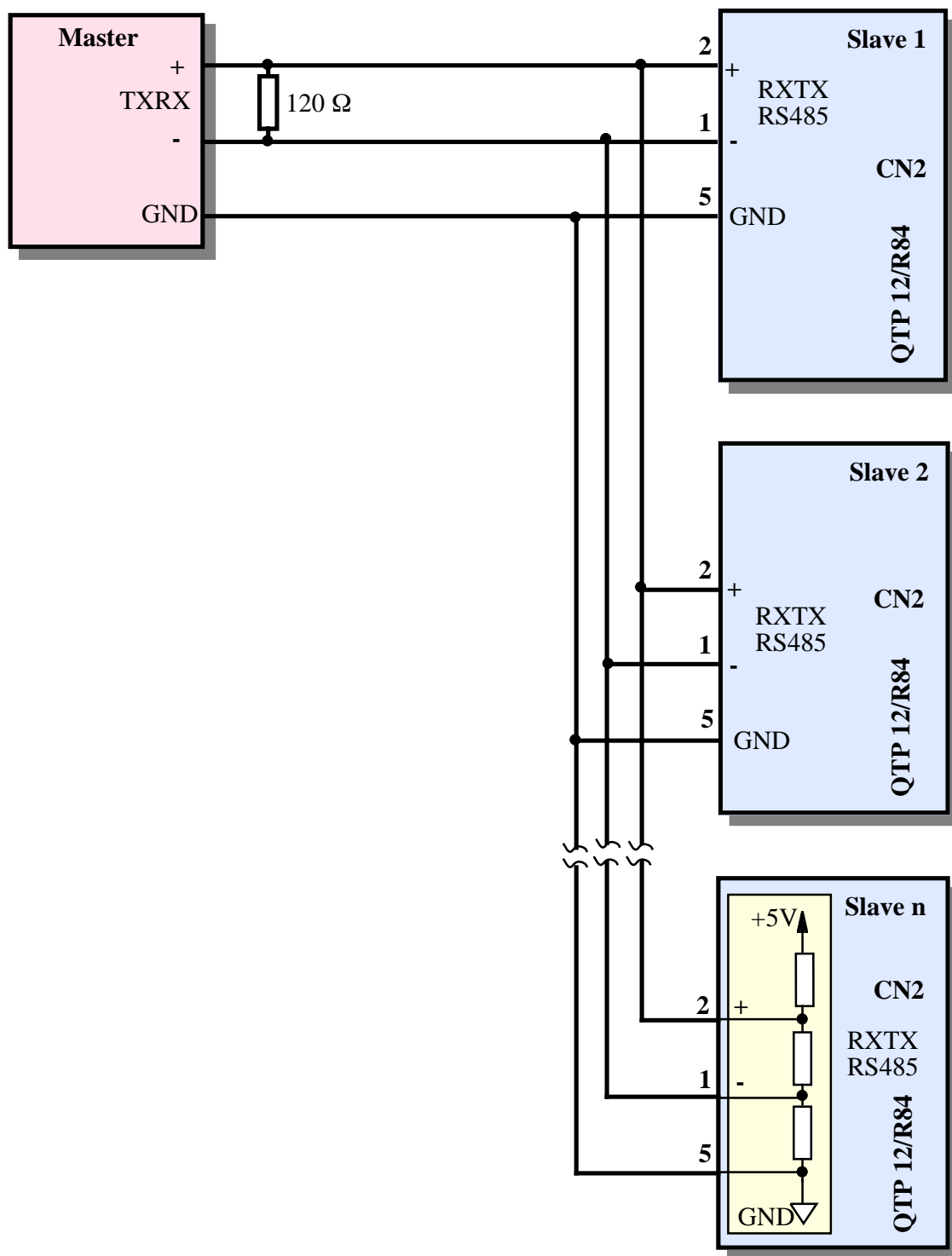


FIGURA 17: ESEMPIO COLLEGAMENTO IN RETE IN RS 485

Da notare che in una rete RS 485, devono essere presenti due resistenze di forzatura lungo la linea e due resistenze di terminazione ($120\ \Omega$), alle estremità della stessa, rispettivamente vicino all'unità Master ed all'ultima unità Slave.

A bordo della **QTP 12/R84** è presente la circuiteria di terminazione e forzatura, che può essere inserita o disinserita, tramite appositi jumpers, come illustrato in seguito.

In merito alla resistenza di terminazione dell'unità Master, provvedere a collegarla solo se questa non è già presente al suo interno (ad esempio molti convertitori RS232-RS485 ne sono già provvisti). Per maggiori informazioni consultare il Data-Book TEXAS INSTRUMENTS, "RS 422 and RS 485 Interface Circuits", nella parte introduttiva riguardante le reti RS 422-485.

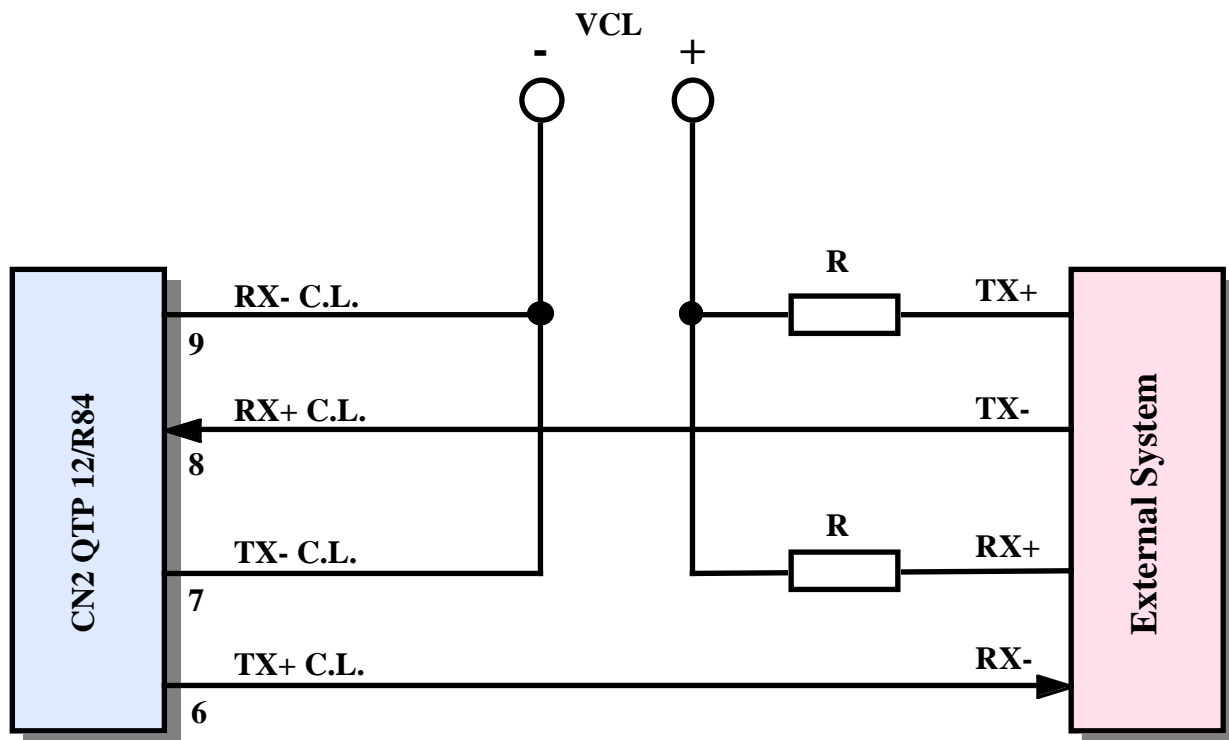


FIGURA 18: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI

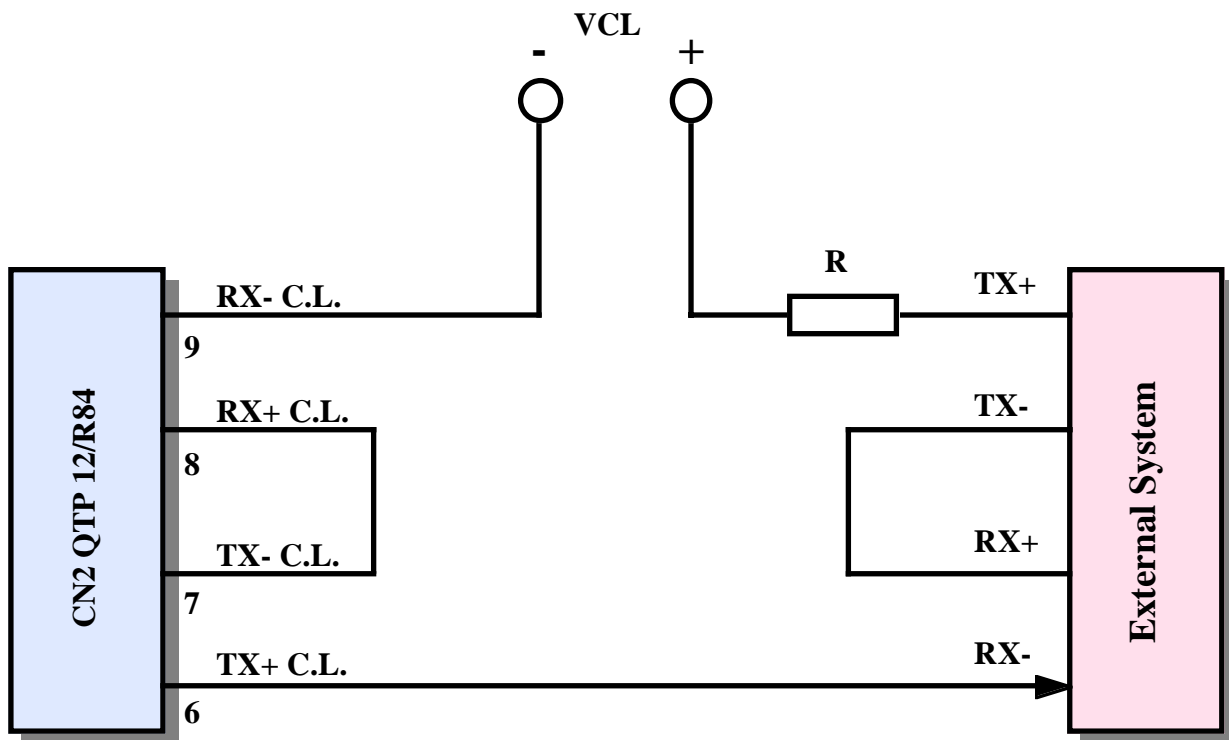


FIGURA 19: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI

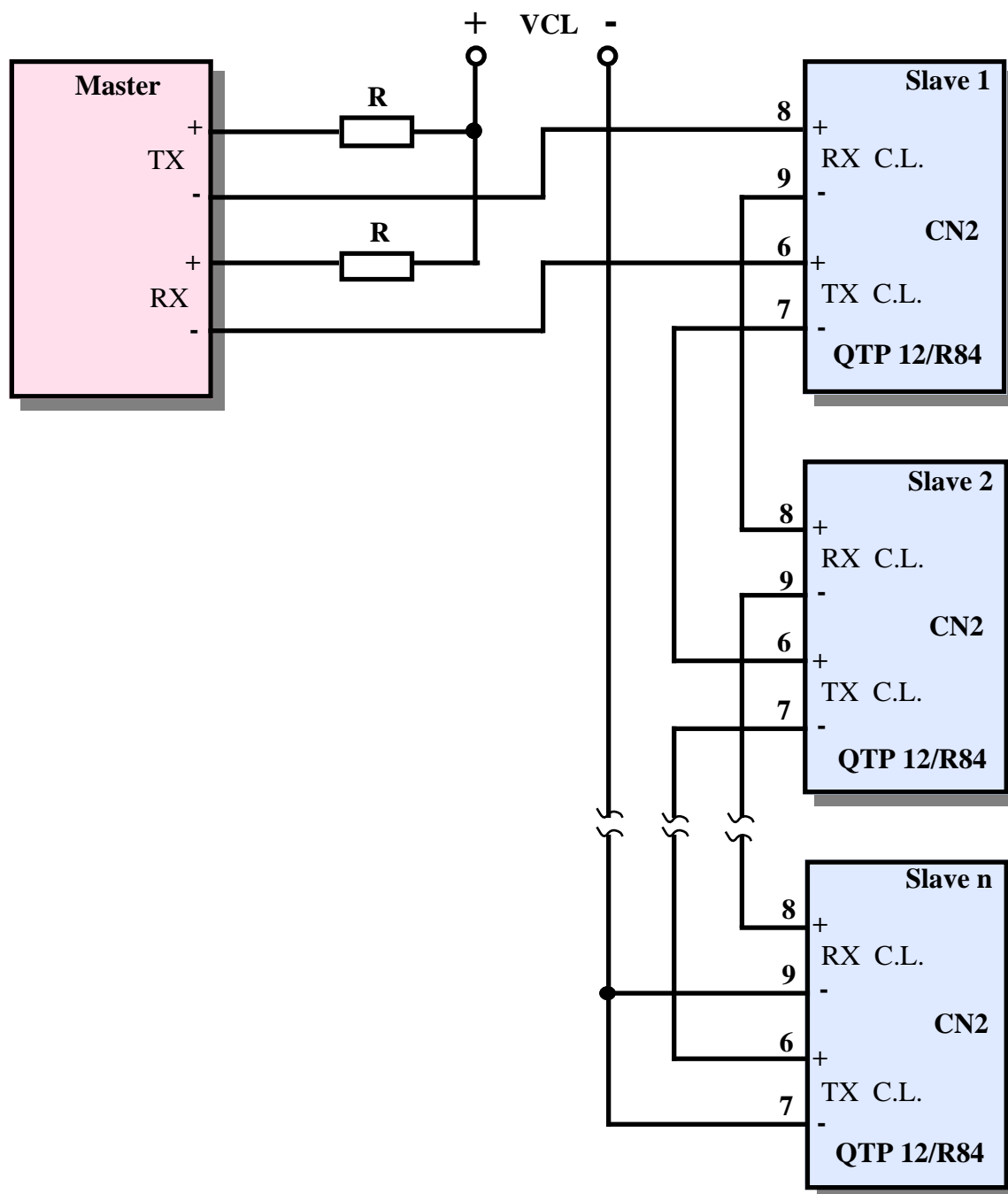


FIGURA 20: ESEMPIO DI COLLEGAMENTO IN RETE IN CURRENT LOOP

Per il collegamento in Current loop passivo sono possibili due diversi tipi di collegamento: a 2 fili ed a 4 fili. Tali connessioni sono riportate nelle figure 18÷20; in esse è indicata la tensione per alimentare l'anello (**VCL**) e le resistenze di limitazione della corrente (**R**). I valori di tali componenti variano in funzione del numero di dispositivi collegati e della caduta sul cavo di collegamento; bisogna quindi effettuare la scelta considerando che:

- si deve garantire la circolazione di una corrente di **20 mA**;
- su ogni trasmettitore cadono mediamente **2,35 V** con una corrente di 20 mA;
- su ogni ricevitore cadono mediamente **2,52 V** con una corrente di 20 mA;
- in caso di cortocircuito sulla rete ogni trasmettitore dissipa al massimo **125 mW**;
- in caso di cortocircuito sulla rete ogni ricevitore dissipa al massimo **90 mW**.

Per maggiori informazioni consultare il Data-Book HEWLETT-PACKARD, nella parte che riguarda gli opto accoppiatori per Current loop denominati **HCPL 4100** e **HCPL 4200**.

CN3 - CONNETTORE PER LINEA CAN

CN3 é un connettore a morsetteria, a rapida estrazione, verticale, con passo 3,5 mm a 3 vie. Tramite CN3 si può collegare la scheda ad una linea di comunicazione seriale CAN ottenendo un veloce, comodo ed efficiente nodo sul BUS di campo definito dallo stesso protocollo. La disposizione dei segnali é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo, seguendo le normative dello stesso standard.

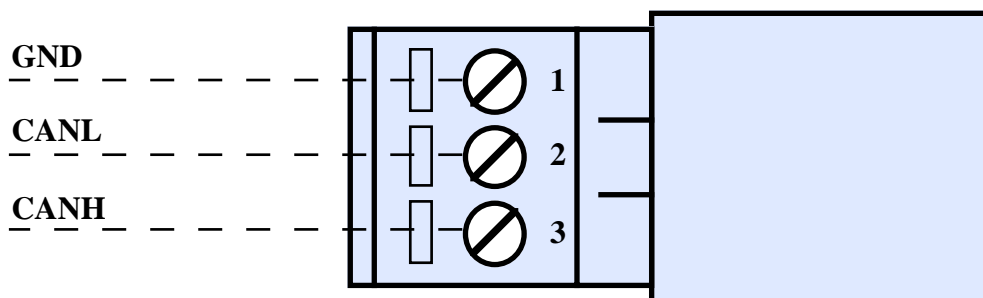


FIGURA 21: CN3 - CONNETTORE PER LINEA CAN

Legenda:

CANL = I/O - Linea differenziale low per linea CAN.
CANH = I/O - Linea differenziale high per linea CAN.
GND = - Linea di massa.

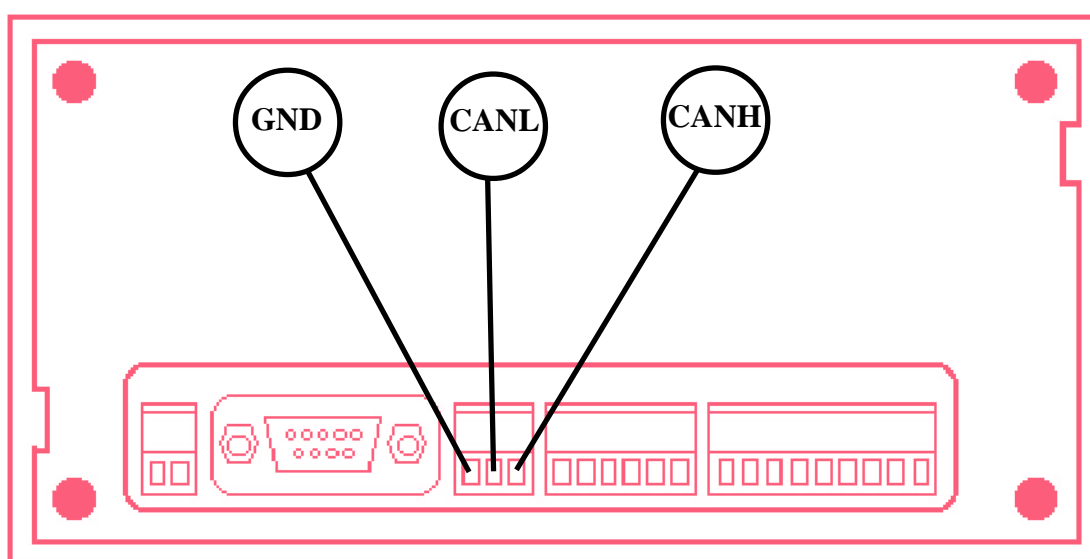


FIGURA 22: COLLEGAMENTO LINEA CAN

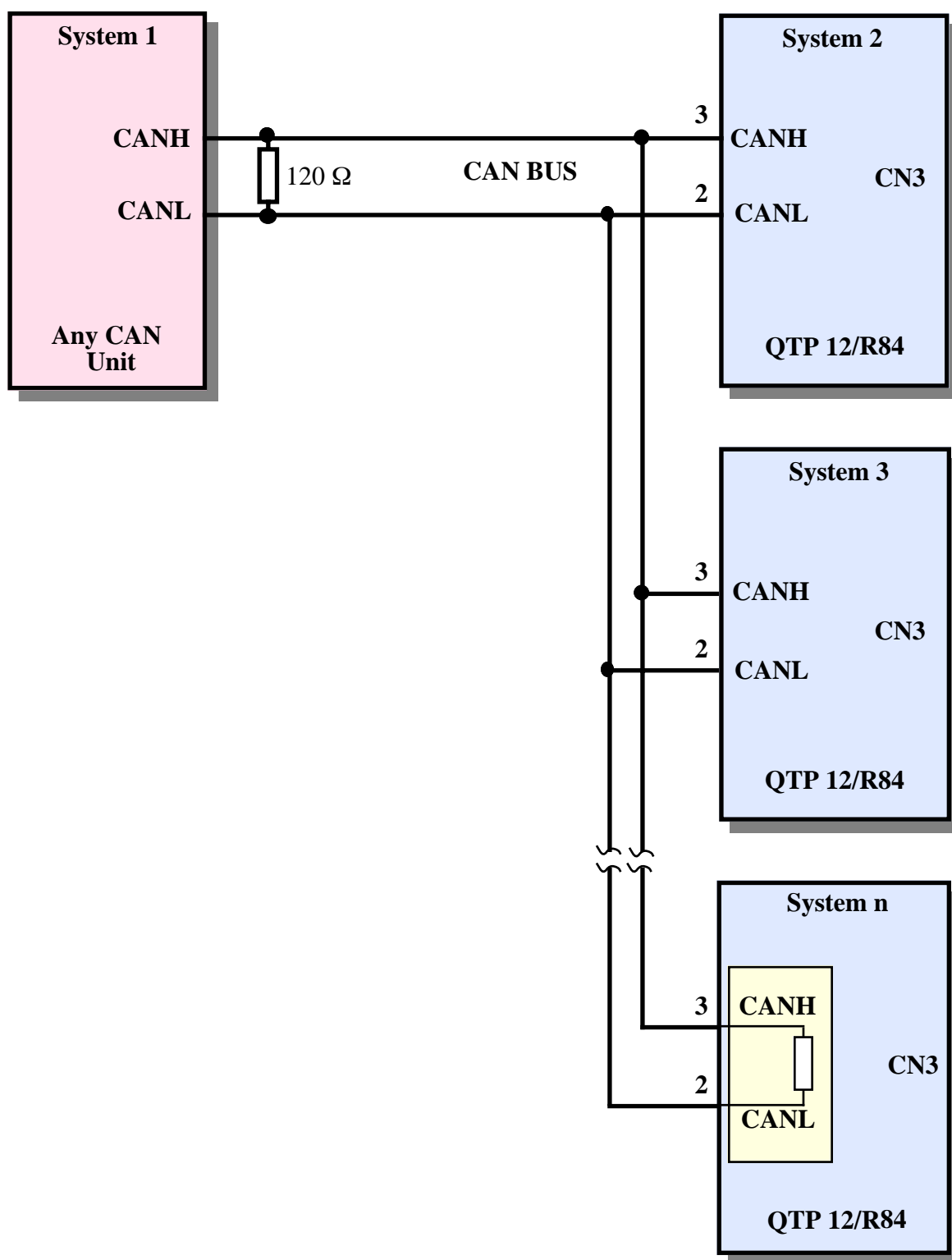


FIGURA 23: ESEMPIO COLLEGAMENTO IN RETE CAN

Da notare che una rete CAN, deve avere un'impedenza di linea di 60 Ω e per questa ragione lungo la linea possono essere presenti due resistenze di terminazione (120 Ω), alle estremità della stessa. A bordo della **QTP 12/R84** è presente la circuiteria di terminazione che può essere inserita o disinserita, tramite un apposito jumper, come illustrato in seguito.

Qualora i sistemi collegati sulla rete CAN risultino a differenze di potenziale elevate si può ovviare ad eventuali problemi di comunicazione e/o funzionamento, collegando anche le masse dei sistemi ovvero il pin 1 di CN3.

CN4 - CONNETTORE PER USCITE DIGITALI A RELÉ

CN4 é un connettore a morsettieria per rapida estrazione, a passo 3,5 mm, composto da 6 contatti. Tramite CN4 possono essere collegati i contatti normali aperti ed i relativi comuni delle 4 uscite a relé, presenti sulla **QTP 12/R84**. In fase di collegamento si deve ricordare che il carico massimo sopportato da ogni linea è di **5 A** (resistivi), con un tensione massima di **30 Vdc**.

La gestione di queste uscite avviene tramite una serie di segnali del microcontrollore, opportunamente bufferati, i quali sono stati accuratamente scelti, in modo da semplificare al massimo la gestione software (per maggiori informazioni vedere il capitolo **COMANDI**).

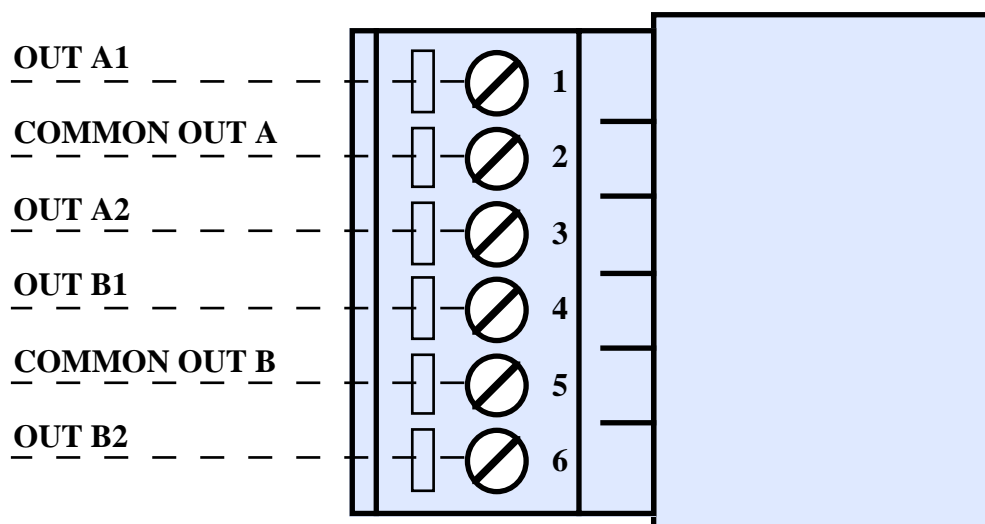


FIGURA 24: CN4 - CONNETTORE PER USCITE DIGITALI A RELÉ

Legenda:

- OUT An** = O - Contatto normale aperto del relé n, del gruppo A.
- COMMON OUT A** = - Contatto comune dei relé del gruppo A.
- OUT Bn** = O - Contatto normale aperto del relé n, del gruppo B.
- COMMON OUT B** = - Contatto comune dei relé del gruppo B.

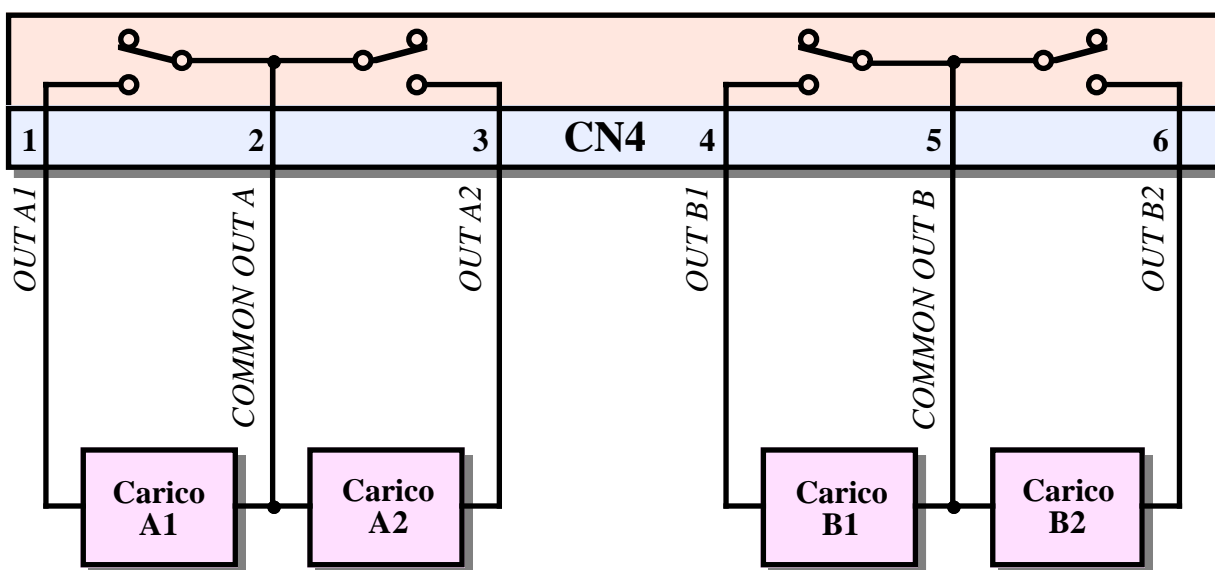


FIGURA 25: COLLEGAMENTO USCITE A RELÉ

I relè sono pilotati da 4 transistori PNP che a loro volta sono gestiti attraverso altrettanti pins di I/O del microcontrollore.

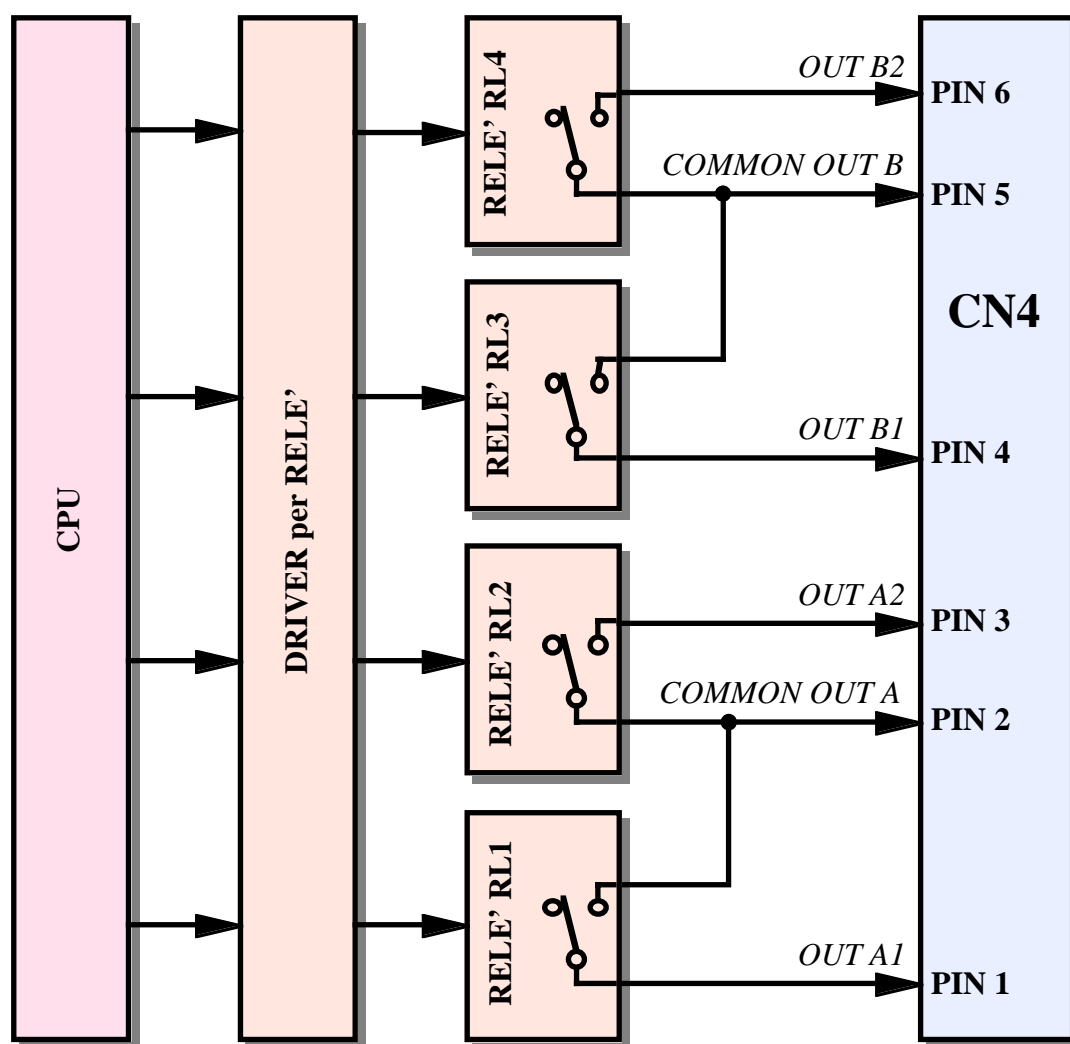


FIGURA 26: SCHEMA DELLE USCITE A RELÉ

Le precedenti figure 25 e 26 riportano rispettivamente il collegamento esterno ed a bordo scheda per le uscite digitali a relé. Come descritto in queste figure sono stati previsti due gruppi di relé denominati A e B provvisti di un proprio comune. In questo modo si possono collegare anche carichi esterni dotati di due distinte tensioni di alimentazione, ottenendo una notevole facilitazione nei cablaggi di tutto il sistema.

CN5 - CONNETTORE PER INGRESSI DIGITALI OPTOISOLATI

CN5 é un connettore a morsettieria per rapida estrazione, a passo 3,5 mm, composto da 9 contatti. Tramite CN5 possono essere collegati gli 8 ingressi optoisolati di tipo NPN o PNP, disponibili sulla scheda **QTP 12/R84**. Sul connettore oltre alle linee degli ingressi, é presente anche il segnale comune a cui collegare rispettivamente il positivo od il negativo della tensione galvanicamente isolata +V opto. Le linee del microcontrollore collegate agli ingressi di CN5 sono state scelte in modo da poter sfruttare appieno le periferiche interne, quindi gli stessi ingressi possono generare interrupts, essere contati dai contatori hardware, ecc, come descritto nel capitolo **COMANDI**.

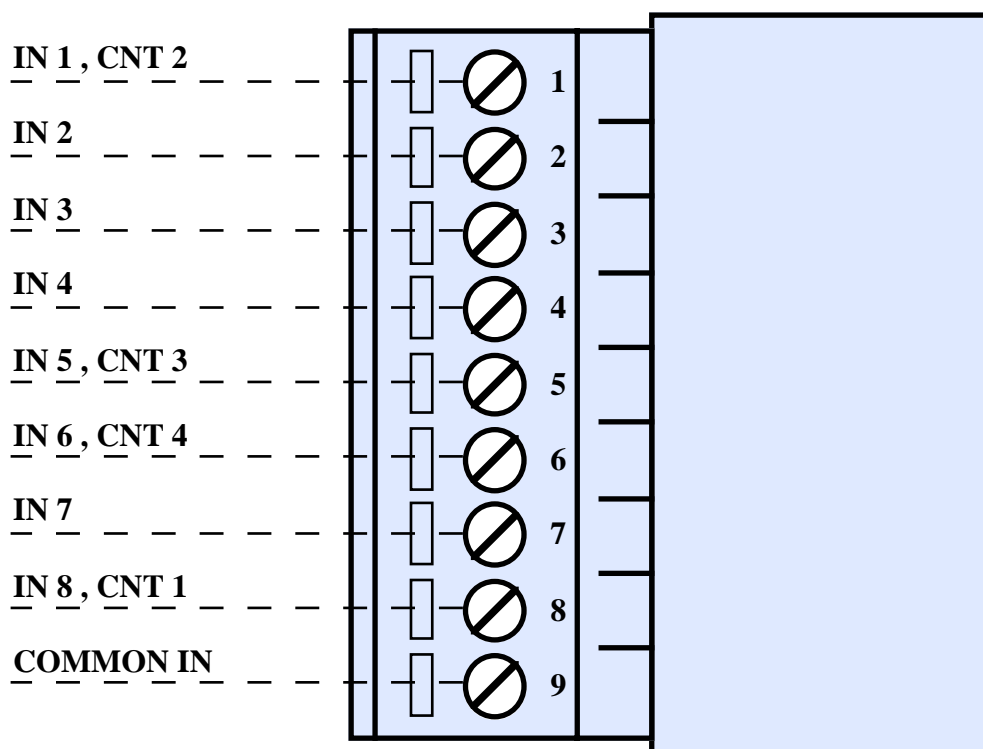


FIGURA 27: CN5 - CONNETTORE PER INGRESSI DIGITALI OPTOISOLATI

Legenda:

- IN n** = I - Ingresso n opto isolato di tipo NPN o PNP.
CNT n = I - Ingresso per contatore n opto isolato di tipo NPN o PNP.
COMMON IN = - Linea positiva (NPN) o negativa (PNP) della tensione di alimentazione esterna +V opto.

Le linee di input disponibili sulla scheda, sono del tipo optoisolato e sono dotati di filtro passa basso; in questo modo é garantita una certa protezione dell'elettronica interna, rispetto ai possibili disturbi provenienti dall'esterno.

L'uso di una circuiteria d'ingresso in cui la corrente può circolare in entrambi i sensi, rende le linee adatte sia a driver del tipo **NPN** che **PNP**.

La tensione di alimentazione degli optoisolatori +V opto deve essere fornita dall'esterno sempre sul connettore CN5 e deve soddisfare i requisiti descritti nel paragrafo **CARATTERISTICHE ELETTRICHE**.

Le successive figure 28 e 29 mostrano rispettivamente il collegamento degli ingressi digitali optoisolati a bordo scheda ed all'esterno.

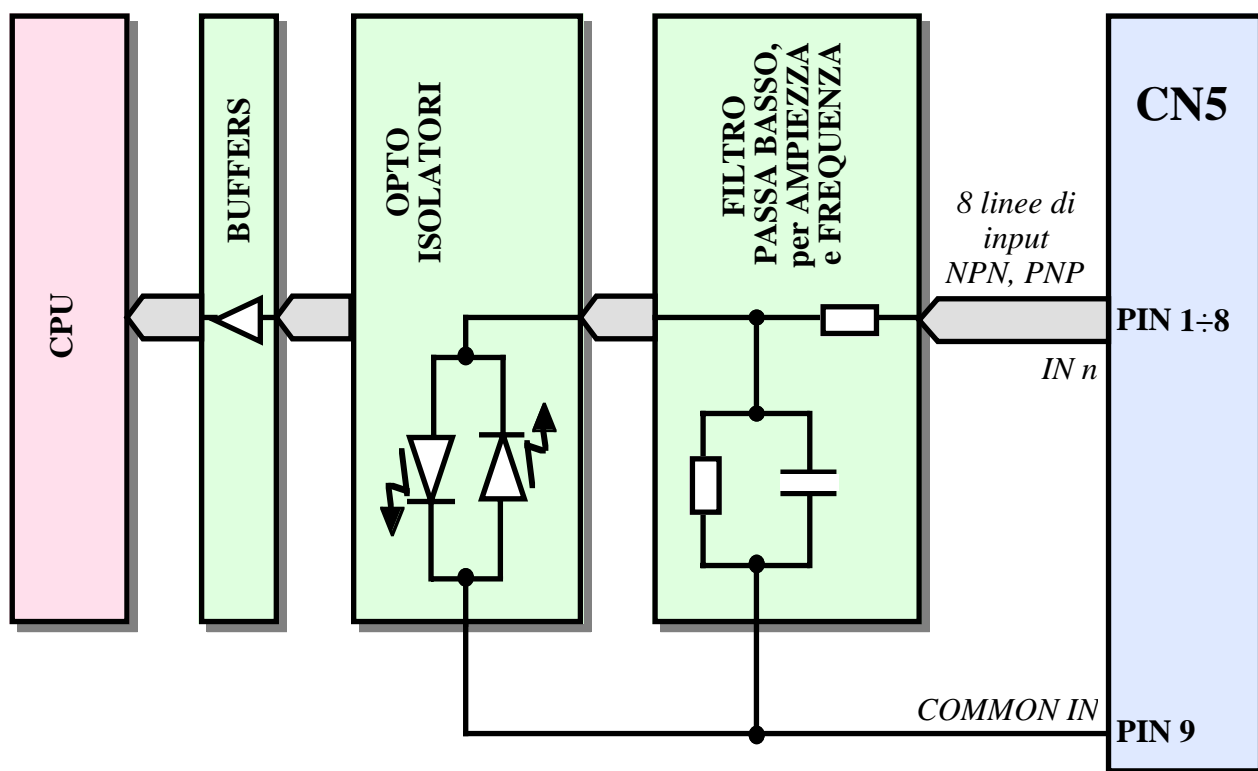


FIGURA 28: SCHEMA DEGLI INGRESSI OPTOISOLATI

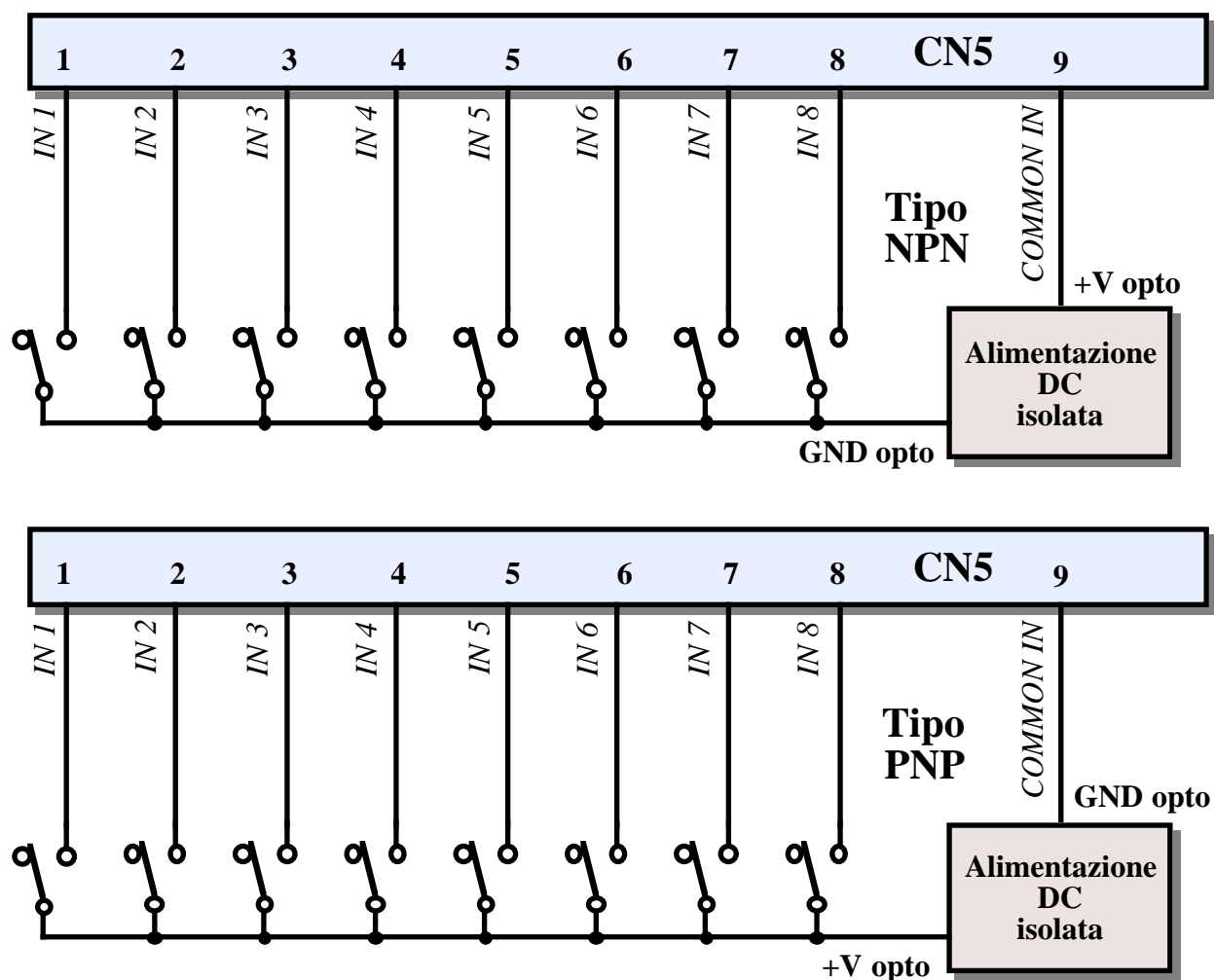


FIGURA 29: COLLEGAMENTO INGRESSI OPTOISOLATI

J6 - CONNETTORE PER I/O TTL, 1-WIRE, I2C BUS

J6 è un connettore strip maschio, verticale, passo 2.54 mm, composto da 4 vie.

Su J6 sono disponibili due linee di I/O digitali ed i terminali dell'alimentazione a +5 Vdc generata a bordo per poter alimentare comodamente i dispositivi o sistemi esterni alla scheda. Grazie a queste linee ed un'adeguata gestione firmware su J6 si possono collegare tutti i dispositivi periferici che seguono lo standard I2C BUS, 1-WIRE, SPI, ecc. I segnali sono a livello TTL, secondo le normative degli stessi standard, e sono disposti in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione.

J6 non è accessibile dallo scasso della parete posteriore del contenitore, quindi per utilizzarlo si deve aprire il mobile della **QTP 12/R84** ed inserire l'apposito connettore femmina facendo passare i fili sempre nello scasso; in alternativa si può montare anche l'elettronica su un piccolo stampato da montare in piggy back su J6, lasciandolo all'interno dello stesso contenitore.

Il connettore femmina per J6 può essere ordinato alla **grifo** con il codice **CKS.AMP4** (kit con 4 contatti a crimpare e connettore AMP Mod II femmina 4 vie) oppure **AMP4 CABLE** (connettore finito, con 4 fili lunghi 1 metro).

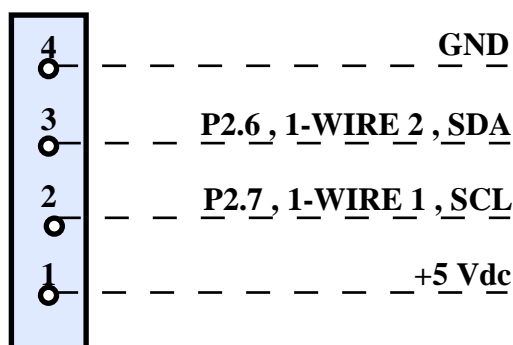


FIGURA 30: J6 - CONNETTORE PER I/O TTL, 1-WIRE, I2C BUS

Legenda:

P2.n	= I/O - Segnale collegato al Port 2.n del microcontrollore.
1-WIRE n	= I/O - Segnale gestione linea 1-WIRE n.
SDA	= I/O - Segnale di dati dell'I2C BUS.
SCL	= O - Segnale di clock dell'I2C BUS.
+5 Vdc	= O - Positivo della tensione di alimentazione a +5 Vdc.
GND	= - Linea di massa.

Si ricorda che elettricamente i pin 2 e 3 di J6 sono collegati ad una resistenza di pull up da 10KΩ, come indicato nella seguente figura:

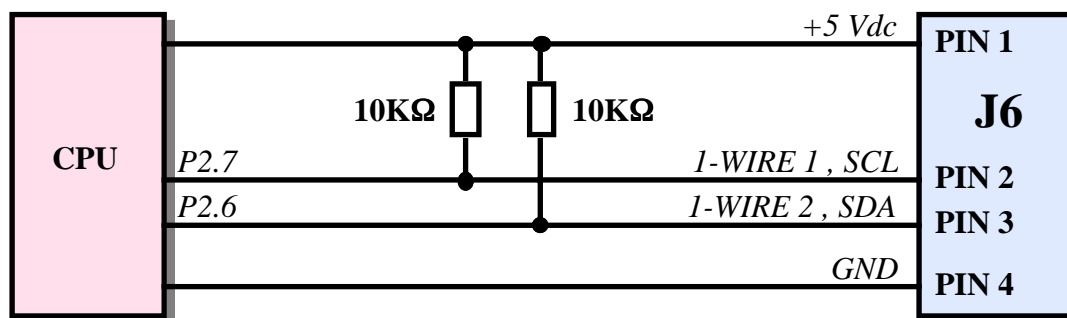


FIGURA 31: SCHEMA DEGLI I/O TTL, 1-WIRE, I2C BUS

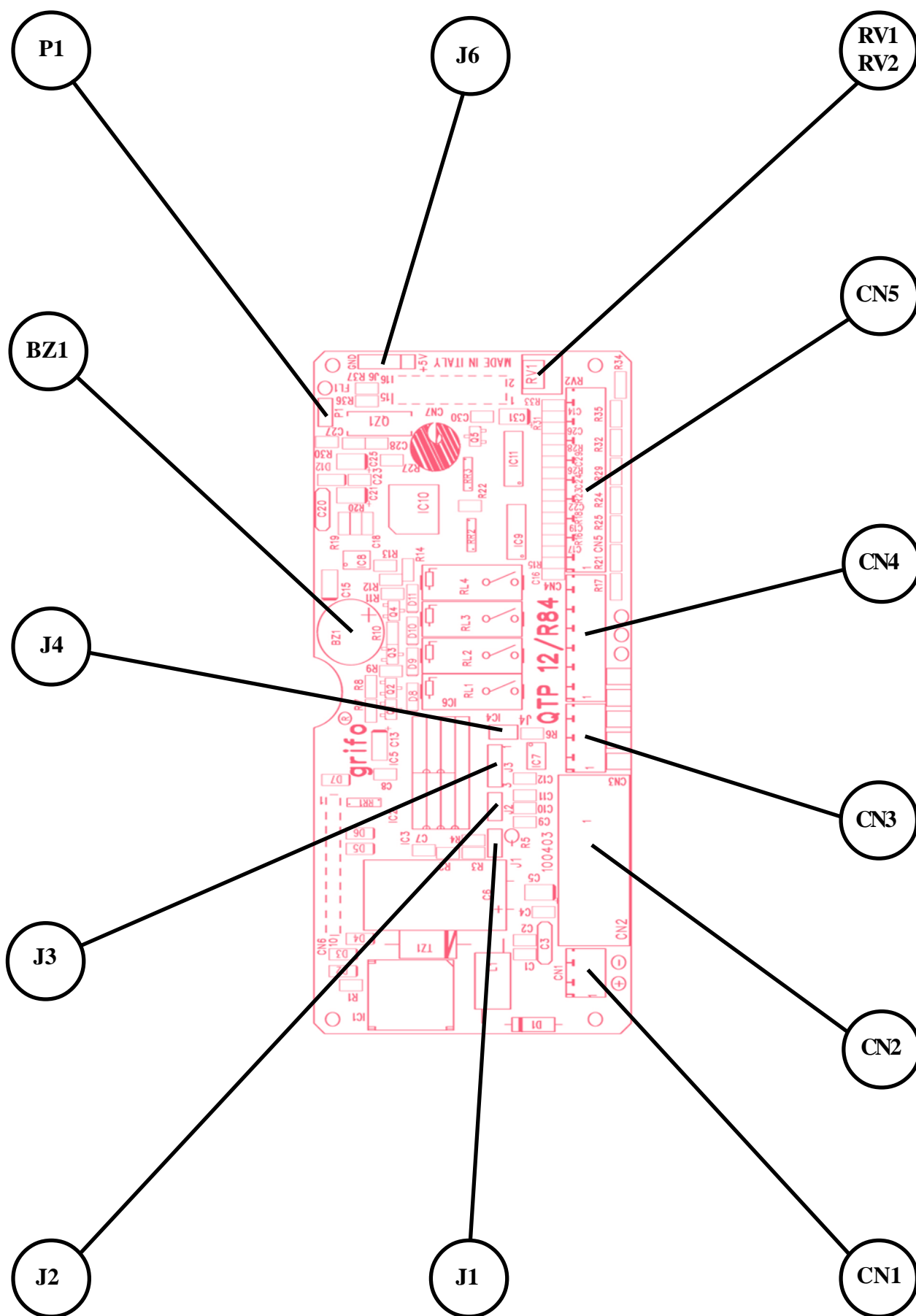


FIGURA 32: DISPOSIZIONE JUMPERS, CONNETTORI, TRIMMER, ECC.

TRIMMER REGOLAZIONE CONTRASTO

A bordo della **QTP 12/R84** é presente un trimmer che consente di definire il contrasto sui display LCD. Questo trimmer denominato RV1 o RV2 viene settato dalla **grifo®** in modo da ottenere la migliore visibilità del display in tutte le condizioni operative e l'utente normalmente non deve variarne la posizione. Nel caso di particolari esigenze, come condizioni di illuminazioni estremamente forti o deboli, si può intervenire sul trimmer effettuando minime variazioni nei due sensi di rotazione e verificando che la visibilità del display migliori. Per individuare il trimmer di regolazione contrasto sulla scheda, fare riferimento alla figura 30.

JUMPERS

Esistono a bordo della **QTP 12/R84** cinque jumpers a cavaliere, con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. Di seguito ne é riportato l'elenco e la loro funzione nelle varie modalità di connessione.

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J1 , J2	non connessi connessi	Non collegano la circuiteria di terminazione e forzatura alla linea seriale RS 422, RS 485. Collegano la circuiteria di terminazione e forzatura alla linea seriale RS 422, RS 485.	*
J3	posizione 1-2 posizione 2-3	Configura la linea seriale per lo standard elettrico RS 485 (half duplex a 2 fili). Configura la linea seriale per lo standard elettrico RS 422 (full duplex o half duplex a 4 fili).	*
J4	non connesso connesso	Non collega la resistenza di terminazione da 120 Ω alla linea CAN. Collega la resistenza di terminazione da 120 Ω alla linea CAN.	*
P1	non connesso connesso	Seleziona modalità RUN, eseguendo programma applicativo salvato in FLASH EPROM. Seleziona modalità DEBUG, eseguendo il boot loader (vedere paragrafo PROGRAMMAZIONE FLASH EPROM).	*

FIGURA 33: TABELLA JUMPERS

Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa o alla figura 33 di questo manuale, dove viene riportata la numerazione dei pin dei jumpers, che coincide con quella utilizzata nella precedente descrizione. Per l'individuazione dei jumpers a bordo della scheda, si utilizzi invece la figura 30.

Nella precedente tabella l'* (asterisco) indica la connessione di default, ovvero quella impostata in fase di collaudo, con cui la scheda viene fornita.

Ulteriori informazioni sulla funzione dei jumper della **QTP 12/R84** sono riportate nei seguenti paragrafi relativamente alla sezione su cui gli stessi jumper intervengono.

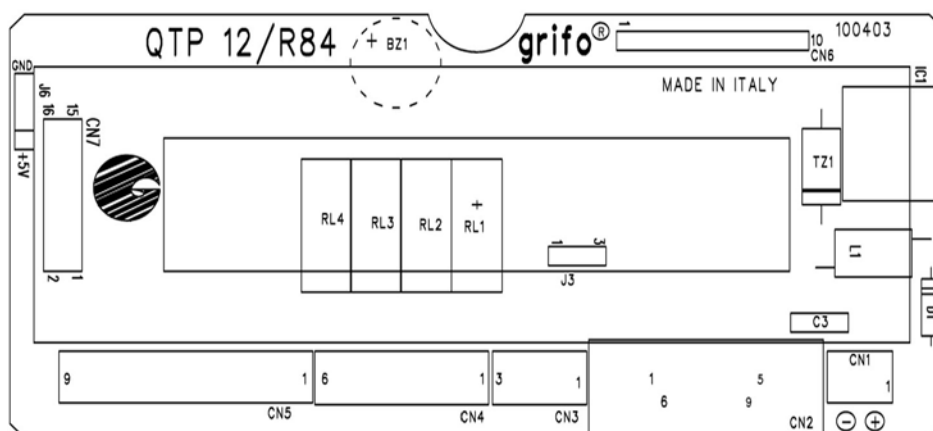


FIGURA 34: PIANTE COMPONENTI LATO STAGNATURE

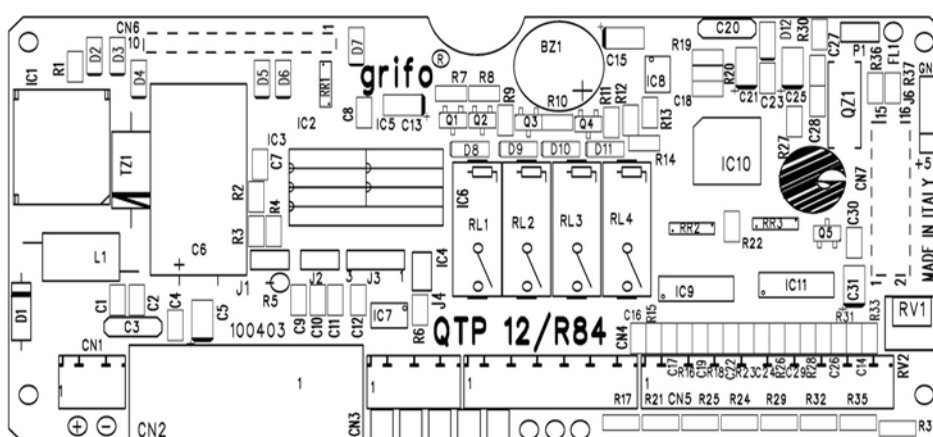


FIGURA 35: PIANTE COMPONENTI LATO COMPONENTI

CONFIGURAZIONE LINEA SERIALE

La linea di comunicazione seriale della **QTP 12/R84** può essere bufferata in RS 232, RS 422, RS 485 o Current loop. Dal punto di vista software sulla linea può essere definito il protocollo fisico e logico di comunicazione tramite le apposite modalità offerte dai firmware di gestione.

La selezione del protocollo elettrico avviene via hardware e richiede un'opportuna configurazione dei jumpers di bordo, come descritto nelle precedenti tabelle, e l'installazione di adeguati driver di comunicazione. Alcuni componenti necessari per le configurazioni RS 422, RS 485 e Current loop non sono montati e collaudati sulla scheda in configurazione di default; per questo la prima configurazione della seriale non in RS 232 deve essere sempre ordinata specificando la corrispondente opzione in modo che i tecnici **grifo®** la possono montare e collaudare. A questo punto l'utente può cambiare autonomamente la configurazione seguendo le informazioni sotto riportate:

- LINEA SERIALE SETTATA IN RS 232 (configurazione default)

J3	=	indifferente	IC4	=	driver MAX 202
J1 , J2	=	non connessi	IC2	=	nessun componente
			IC5	=	nessun componente
			IC3	=	nessun componente
			IC6	=	nessun componente

- LINEA SERIALE SETTATA IN CURRENT LOOP (opzione **.CLOOP**)

J3	=	indifferente	IC4	=	nessun componente
J1 , J2	=	non connessi	IC2	=	nessun componente
			IC5	=	nessun componente
			IC3	=	driver HP 4200
			IC6	=	driver HP 4100

Da ricordare che l'interfaccia seriale in Current loop è di tipo passivo e si deve quindi collegare una linea Current loop attiva, ovvero provvista di un proprio alimentatore come descritto nelle figure 18÷20. L'interfaccia Current loop può essere utilizzata per realizzare sia connessioni punto punto che reti multipunto con un collegamento a 4 o 2 fili.

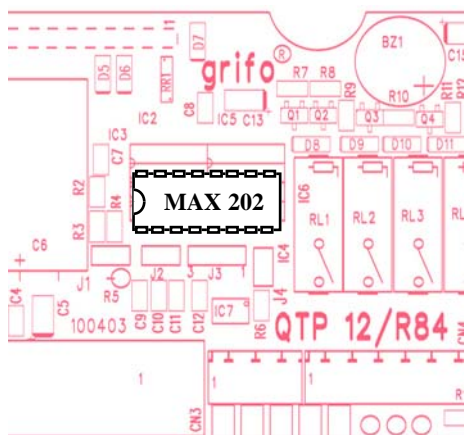
- LINEA SERIALE SETTATA IN RS 422 (opzione **.RS 422**)

J3	=	posizione 2-3	IC4	=	nessun componente
J1 , J2	=	(*)	IC2	=	driver SN 75176 o MAX 483
			IC5	=	driver SN 75176 o MAX 483
			IC3	=	nessun componente
			IC6	=	nessun componente

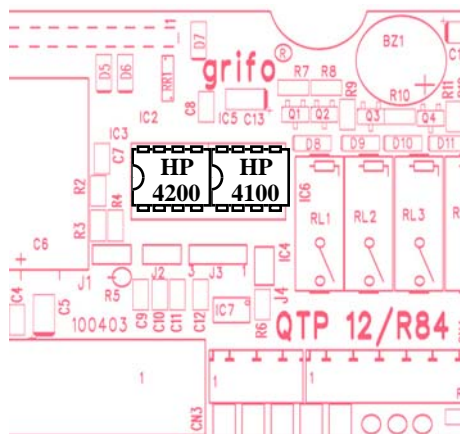
L'interfaccia RS 422 può essere utilizzata per realizzare connessioni full duplex sia punto punto che reti multipunto, con un collegamento a 4 fili. L'attivazione del trasmettitore RS 422, necessaria per realizzare reti, è gestita automaticamente dal *firmware seriale* (selezionando il protocollo logico master slave) oppure dallo stato del segnale P2.5 del microcontrollore, nel caso di *firmware di libreria*, come segue:

P2.5 = DIR = livello basso	=	stato logico 0	->	trasmettitore attivo
P2.5 = DIR = livello alto	=	stato logico 1	->	trasmettitore disattivo

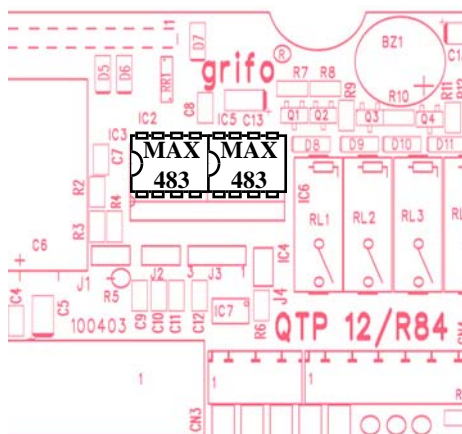
Per sistemi punto punto, la linea DIR può essere mantenuta sempre bassa (trasmettitore sempre attivo), mentre per reti multipunto si deve attivare il trasmettitore solo in corrispondenza della trasmissione.



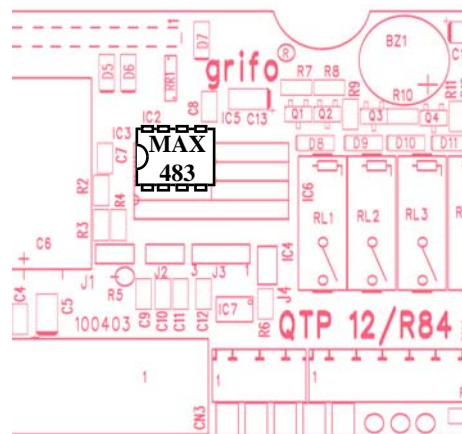
Seriale in RS 232



Seriale in Current loop



Seriale in RS 422



Seriale in RS 485

FIGURA 36: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE

- LINEA SERIALE SETTATA IN RS 485 (opzione **.RS 485**)

		IC4	= nessun componente	
		IC2	= driver SN 75176 o MAX 483	
J3	=	posizione 1-2	IC5	= nessun componente
J1 , J	=	(*)	IC3	= nessun componente
			IC6	= nessun componente

In questa modalità le linee da utilizzare sono i pin 1 e 2 di CN2, che quindi diventano le linee di trasmissione o ricezione a seconda dello stato definito dal firmware. L'interfaccia RS 485 può essere utilizzata per realizzare connessioni half duplex sia punto punto che reti multipunto, con un collegamento a 2 fili. La direzionalità della linea RS 485 è gestita automaticamente dal *firmware seriale* (selezionando il protocollo logico master slave) oppure dallo stato del segnale P2.5 del microcontrollore, nel caso di *firmware di libreria*, come segue:

P2.5 = DIR = livello basso	= stato logico 0	->	linea in trasmissione
P2.5 = DIR = livello alto	= stato logico 1	->	linea in ricezione

Sempre in questa modalità si riceve quanto trasmesso, in modo da fornire al sistema la possibilità di verificare autonomamente la riuscita della trasmissione; infatti in caso di conflitti sulla linea, quanto trasmesso non viene ricevuto correttamente e viceversa.

- (*) Nel caso si utilizzi la linea seriale in RS 422 o RS 485, con i jumpers J1 e J2 è possibile connettere la circuiteria di terminazione e forzatura sulla linea. Tale circuiteria deve essere sempre presente nel caso di sistemi punto punto, mentre nel caso di sistemi multipunto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione.

A seguito del power on, il driver RS 485 è in ricezione o il driver di trasmissione RS 422 è disattivo, in modo da eliminare eventuali conflittualità sulla linea di comunicazione.

Per ulteriori informazioni relative alla comunicazione seriale fare riferimento agli esempi di collegamento delle figure 14÷20.

COLLEGAMENTO LINEA CAN

Il jumper J4 ha il compito di collegare o meno l'apposita resistenza di terminazione della linea CAN come descritto nella tabella di figura 31. Il CAN BUS deve fisicamente coincidere con una linea differenziale con impedenza di 60 Ω e per questo le resistenze di terminazione devono essere collegate in modo da ricreare questa impedenza. In particolare tale collegamento deve essere sempre effettuato in caso di sistemi punto punto, mentre nel caso di sistemi multipunto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione CAN (vedere esempio di figura 23).

La corretta terminazione della linea CAN contribuisce notevolmente al funzionamento della comunicazione, infatti l'interfaccia di linea della **QTP 12/R84** è in grado di sopprimere i transienti e di essere immune ai disturbi di radio frequenza ed elettromagnetici, solo se il collegamento con il campo è effettuato correttamente.

Come descritto nel successivo paragrafo ALIMENTAZIONE la linea CAN non è galvanicamente isolata dalla tensione di alimentazione della scheda quindi la sua massa è collegata direttamente al segnale GND della scheda ed è inoltre riportato su un pin del connettore CN3. Quest'ultimo può essere utilizzato per equipotenziale i vari sistemi CAN ma anche per schermare il collegamento fisico, qualora si utilizzi del cavo con calza, ottenendo la massima protezione contro i disturbi esterni.

INTERRUPTS

Una caratteristica peculiare della **QTP 12/R84** è la notevole potenza nella gestione delle interruzioni. Di seguito viene riportata una breve descrizione di quali sono i dispositivi che possono generare interrupts e con quale modalità; per quanto riguarda la gestione di tali interrupts si faccia riferimento ai data sheets del microprocessore:

- Ingresso optoisolato IN 5 su CN5 -> Genera un /INT0 = P3.2 sul microcontrollore.
- Ingresso optoisolato IN 6 su CN5 -> Genera un /INT1 = P3.3 sul microcontrollore.
- Periferiche del microcontrollore -> Generano un interrupt interno. In particolare tali possibili sorgenti d'interrupt interno sono le sezioni: Timer 0, Timer 1, Timer 2, PCA e le sue modalità di comparazione e capture, UART, CAN, ecc.

Sulla scheda è presente un gestore d'interrupt (ICU) che consente di attivare, disattivare, mascherare le sorgenti d'interrupt e che regola l'attivazione contemporanea di più interrupts. In questo modo l'utente ha sempre la possibilità di rispondere in maniera efficace e veloce a qualsiasi evento esterno, stabilendo anche la priorità delle varie sorgenti.

I firmware sviluppati fanno uso di alcuni interrupt, come descritto nel capitolo DESCRIZIONE SOFTWARE.

INTERFACCIAMENTO DEGLI I/O CON IL CAMPO

Al fine di evitare eventuali problemi di collegamento della scheda con tutta l'elettronica del campo a cui le **QTP 12/R84** si devono interfacciare, si devono seguire le informazioni riportate nei precedenti paragrafi e le relative figure che illustrano le modalità interne di connessione.

- Tutti i segnali a livello TTL possono essere collegati a linee dello stesso tipo riferite alla massa digitale della scheda. Il livello 0V corrisponde allo stato logico 0, mentre il livello 5V corrisponde allo stato logico 1.
- Per i segnali optoisolati d'ingresso, devono essere collegati in serie sia i contatti da acquisire che la +V opto esterna. In dettaglio tali contatti (relé, fine-corsa, interruttori, ecc.) devono quindi effettuare il seguente collegamento tra i segnali presenti su CN5:

SEGNALE	NPN	PNP
IN x	GND opto	+V opto
COMMON IN	+V opto	GND opto

Al fine di evitare problemi di disturbi è preferibile mantenere galvanicamente separata l'alimentazione +V opto da quella di alimentazione ovvero non collegare il segnale GND della scheda al segnale GND opto.

- I segnali d'uscita a relé devono essere collegati direttamente al carico da pilotare (elettrovalvole, relé di potenza, teleruttori, ecc.). La scheda fornisce il contatto normalmente aperto, in grado di sopportare una corrente massima di **5A** con una tensione che può arrivare fino a **30 Vdc**. Per fornire la possibilità di pilotare anche carichi diversi, con alimentazioni distinte, sono previsti due diversi COMUNI relativi a due coppie di relé.
- Per i segnali che riguardano la comunicazione seriale asincrona con i protocolli RS 232, RS 422, RS 485, Current loop, CAN fare riferimento alle specifiche standard di ognuno di questi protocolli.
- Per i segnali che riguardano la comunicazione seriale sincrona con i protocolli I2C BUS, 1-WIRE fare riferimento alle specifiche standard di questi protocolli, ricordando che le linee sono già provviste di resistenza di pull up da 10KΩ.

ALIMENTAZIONE

Il terminale **QTP 12/R84** dispone di una efficiente circuiteria che si presta a risolvere in modo comodo ed efficace il problema dell'alimentazione della scheda in qualsiasi condizione di utilizzo. Tale sezione fornisce l'energia necessaria a tutte le sezioni della scheda ovvero: logica di controllo, display, retroilluminazione, interfacce seriali, interfaccia CAN, LED, buzzer, tastiera, ingressi digitali, uscite digitali, ecc.

Di seguito vengono riportate le tensioni richieste dalla scheda a seconda della configurazione:

Versione base: In questa configurazione a bordo è presente un alimentatore switching che richiede una tensione di $10 \div 38 \text{ Vdc} \pm 5\%$ oppure $8 \div 24 \text{ Vac} \pm 5\%$ che deve essere fornita tramite CN1 (in caso di tensione continua la polarità deve essere rispettata). In questo modo è possibile alimentare il terminale con dispositivi standard del settore industriale come trasformatori, batterie, celle solari, ecc. Per risolvere facilmente ed economicamente il problema dell'alimentazione nella configurazione base si possono utilizzare gli alimentatori **EXPS-1** ed **EXPS-2**, che partono dalla tensione di rete.

Da notare che l'alimentatore switching di bordo è dotato di raddrizzatore a singolo diodo, quindi in caso di alimentazione con una tensione continua, tutti i segnali di massa della scheda (GND) sono allo stesso potenziale.

La versione base, completa della sezione alimentatrice descritta, è quella normale di vendita.

Versione .5Vdc o .ALIM: In questa configurazione non è presente la sezione alimentatrice e quindi si deve fornire una tensione stabilizzata di $5 \text{ Vdc} \pm 5\%$ tramite CN1 (anche in questo caso la polarità deve essere rispettata). In questo modo è possibile alimentare il terminale con alimentatori, altre schede, ecc.

Si ricorda che questa configurazione senza sezione alimentatrice è una condizione particolare (OEM) da concordare con **grifo®**.

+V opto: Indipendentemente dal tipo di alimentazione prescelta la **QTP 12/R84** necessita sempre di una seconda tensione denominata +V opto che fornisce alimentazione agli optoisolatori della sezione di ingresso digitale della scheda; deve essere compresa nel range $8 \div 30 \text{ Vdc}$ e deve essere fornita sul connettore CN5.

La selezione del tipo di sezione alimentatrice della scheda, deve avvenire in fase di ordine infatti questa scelta implica una diversa configurazione hardware che deve essere effettuata dal personale della **grifo®**.

Per garantire la massima immunità ai disturbi e quindi un corretto funzionamento delle schede, è necessario che le due tensioni siano galvanicamente isolate tra di loro; a questo scopo può essere ordinato l'alimentatore **EXPS-2** che svolge questa funzione, partendo dalla tensione di rete.

In merito alla possibilità di alimentare carichi esterni con i segnali +5 Vdc e GND della scheda, ad esempio tramite il connettore J6, si ricorda che il loro consumo deve essere inferiore a:

400 mA - consumo max +5 Vdc	nel caso di QTP 12/R84-C2 e F2
1000 mA - consumo max +5 Vdc	nel caso di QTP 12/R84-GF2

che ad esempio nel caso di utilizzo di una **QTP 12/R84-F2** diventa:

400 mA - 360 mA = 40 mA.

La **QTP 12/R84** è sempre dotata di una circuiteria di protezione a **TransZorb™** per evitare danni dovuti a tensioni non corrette od a rotture della sezione alimentatrice e di un'efficace e distribuita circuiteria di filtro si preoccupa di proteggere la scheda dai disturbi o dal rumore del campo, in modo da migliorare il funzionamento di tutto il sistema.

Per ulteriori informazioni si faccia riferimento al paragrafo CARATTERISTICHE ELETTRICHE.



FIGURA 37: FOTO ALIMENTATORE EXPS-2

DESCRIZIONE SOFTWARE

Al fine di semplificare e velocizzare l'uso del modulo **QTP 12/R84**, sono stati sviluppati due firmware distinti che consentono all'utente di sviluppare l'applicativo ad alto livello senza dover preoccuparsi della gestione diretta dell'hardware. I due firmware in questione coincidono con delle opzioni definibili in fase di ordine, sono stati denominati *seriale* (**.SER**) e *di libreria* (**.LIB**) ed hanno molteplici caratteristiche comuni ma sostanziali differenze di utilizzo, come descritto di seguito. Si ricorda che in caso di nessuna indicazione d'ordine la configurazione base prevede il *firmware di libreria* e quindi solo quello *seriale* deve essere espressamente ordinato. Entrambi i firmware vengono forniti corredati di comodi programmi dimostrativi sia a livello sorgente che eseguibile; questi possono essere usati senza alcuna variazione per una prima prova del prodotto e poi modificati, o riutilizzati in parte, per realizzare il programma applicativo dell'utente. La documentazione dettagliata dei comandi disponibili in entrambi i firmware, utilizzabili per usufruire di tutte le caratteristiche della **QTP 12/R84**, è invece riportata nel capitolo successivo.

DATI IN EEPROM

Nella EEPROM di bordo i firmware della **QTP 12/R84** salvano una serie di dati che possono essere modificati e/o usati tramite gli appositi comandi. La scelta di usare una EEPROM è stata effettuata proprio per avere tutte le garanzie sulla validità e sul mantenimento dei dati, naturalmente anche in assenza di alimentazione. La descrizione dettagliata su ognuno di questi dati è riportata nel capitolo successivo in corrispondenza dei paragrafi relativi ai comandi che li trattano direttamente. In corrispondenza dell'acquisto ed eventuali riparazioni la EEPROM viene fornita già settata con i suoi valori di default, che sono:

byte di presenza	->	255 (FFH)
codici dei tasti	->	quelli riportati nella tabella di figura 38
modelli dei caratteri definibili da utente	->	255 (FFH)
messaggi	->	255 (FFH)
byte utente ad uso generico	->	255 (FFH)

Qualora l'utente desideri ripristinare la configurazione di default su tutti i dati in EEPROM, i firmware prevedono le modalità sotto descritte: l'utente deve fare molta attenzione alla scelta di inizializzare l'EEPROM infatti tutti i dati preventivamente salvati vengono definitivamente persi.

- *firmware seriale*: con il menù EEPROM DATA del setup locale l'utente può decidere se lasciare inalterati tali dati (opzione NOINI) oppure settarli al loro valore di default (opzione INIT). In dettaglio selezionando l'opzione INIT i dati in EEPROM vengono settati come sopra descritto ed uscendo dal setup locale viene rappresentata un'apposita stringa sul display assieme ad una barra a scorrimento di * (asterischi) che informa sullo stato di avanzamento dell'operazione. La durata della fase di inizializzazione è di circa 20 secondi mentre gli * rappresentati in questa fase devono essere 10.
- *firmware di libreria*: usando l'apposita funzione che inizializza prima la EEPROM e poi il firmware stesso, ovvero effettuando la chiamata descritta nel paragrafo INTEGRAZIONE ED USO DEL FIRMWARE DI LIBRERIA. Si ricorda che anche in questo caso la durata dell'esecuzione di questa funzione è di circa 20 secondi.

RAPPRESENTAZIONE DI CARATTERI SUL DISPLAY

I firmware della **QTP 12/R84** visualizzano sul display di bordo tutti i caratteri ricevuti aventi un codice compreso nel range **0÷255 (0÷FF Hex)** incluso quello che identifica le sequenze di comandi (**27 = 1BH**), come successivamente descritto. Il carattere viene visualizzato nella posizione attuale del cursore, e quest'ultimo avanza nella posizione successiva; se si trova nell'ultimo carattere del display (angolo in basso a destra), viene posizionato nella posizione di Home (angolo in alto a sinistra). Per quanto riguarda la corrispondenza codice e carattere rappresentato, valgono le seguenti informazioni:

Codici	Caratteri
0 ÷ 15 (00÷0F Hex)	Definibili da utente
16 ÷ 31 (10÷1F Hex)	Speciali e variabili in funzione del display installato
32 ÷ 127 (20÷7F Hex)	ASCII standard
128 ÷ 255 (80÷FF Hex)	Speciali e variabili in funzione del display installato

Per rappresentare i caratteri definibili da utente e quelli speciali, i cui codici coincidono con quelli di comandi ad un solo carattere, è stato previsto un apposito comando che seleziona la modalità operativa del firmware **QTP 12/R84** tra le due disponibili:

comandi	in cui i caratteri non sono rappresentati ma interpretati ed eseguiti i relativi comandi;
rappresentazione	in cui i caratteri sono sempre rappresentati.

A seguito di una accensione od inizializzazione è automaticamente selezionata la modalità comandi in modo da rendere subito disponibili tutte le funzionalità. I comandi a più di un carattere, che iniziano sempre con il carattere **ESC = 27 = 1BH**, sono invece sempre interpretati ed eseguiti indipendentemente dalla modalità scelta.

Grazie ai firmware tutti i modelli di **QTP 12/R84** dispongono di 8 caratteri definibili dall'utente che possono essere settati e/o salvati a seconda delle esigenze e quindi rappresentati sul display, come ampiamente descritto nel successivo paragrafo **COMANDI PER CARATTERI UTENTE**.

Per i caratteri speciali invece, si faccia riferimento all'APPENDICE B e si ricordi che è possibile avere altri modelli di display, con differenti caratteri speciali, ma il tutto deve essere direttamente concordato con la **grifo®**.



FIGURA 38: FOTO CARATTERI DISPONIBILI SU QTP 12/R84-GF2

BUFFER DI RICEZIONE

I firmware della **QTP 12/R84** sono dotati di un buffer di ricezione che rende la scheda più flessibile dal punto di vista del colloquio con il programma utente, riducendo i tempi di attesa di quest'ultimo. Ogni dato ricevuto viene immediatamente salvato in questo buffer (della dimensione di **30 bytes**) e quindi esaminato al termine del comando in corso.

- Nel caso di *firmware seriale* se il programma utente trasmette continuamente sequenze di comando che richiedono un lungo tempo di esecuzione (comandi di cancellazione, comandi di gestione della EEPROM, comandi grafici, ecc.) tale buffer é destinato ad riempirsi o addirittura traboccare. In caso di traboccamento l'ultima locazione del buffer viene sovrascritta dai caratteri sucessivamente ricevuti, con conseguente loro perdita. L'unità master deve quindi bloccare la trasmissione fino a quando il firmware seriale della **QTP 12/R84** non ha svuotato il buffer di ricezione, ed é nuovamente pronta a ricevere altri dati. Praticamente l'utente deve inserire dei ritardi nella comunicazione, calibrati sperimentalmente, tali da evitare il traboccamento del buffer di ricezione.
- Nel caso di *firmware di libreria* il buffer non può riempirsi eccessivamente e quindi il programma utente non necessita di alcun ritardo antitraboccamento.

ACQUISIZIONE DELLA TASTIERA

Quando i firmware riconoscono la pressione di un tasto, forniscono il codice relativo. E' inoltre implementata una funzione di **auto repeat** del tasto premuto, con cui se i firmware riconoscono la pressione di un determinato tasto per un tempo superiore ai **0.5 secondi**, iniziano a fornire il suo codice circa ogni **0.1 secondi**, fino a quando quel tasto viene rilasciato. Se la funzione di **keyclick** é attivata, in corrispondenza di ogni fornitura del codice del tasto premuto, viene inoltre emesso un beep dal buzzer di bordo in modo da segnalare acusticamente l'evento all'utente. Qualora il buzzer sia attivato o attivato ad intermittenza, la funzione di keyclick produrrà l'effetto opposto, ovvero spegnerà il buzzer per un piccolo intervallo di tempo. Un'altra caratteristica offerta dai firmware é la totale riconfigurabilità dei tasti da parte del programma utente, ovvero é possibile cambiare il codice fornito in corrispondenza della pressione del tasto o addirittura disattivarlo.

- Nel caso di *firmware seriale* i codici forniti sono trasmessi sulla seriale immediatamente se si utilizza la comunicazione normale mentre nel caso di comunicazione master slave, solo su specifica richiesta dell'unità master di comando, con il formato illustrato nei paragrafi successivi. Inoltre con questo firmware quattro tasti sono utilizzati per gestire il settaggio locale di alcuni parametri di funzionamento, come descritto nell'apposito paragrafo SET UP LOCALE.
- Con il *firmware di libreria* i codici forniti sono momentaneamente salvati in un apposito buffer di trasmissione e quindi restituiti al programma applicativo utente quando questo li richiede.

CODICI DEI TASTI

Di seguito é riportata una figura con la numerazione dei tasti ed una tabella con i codici di default che vengono forniti, dai firmware della **QTP 12/R84**, quando viene premuto un tasto. Come nel caso delle sequenze di comando, questi sono espressi in forma decimale, esadecimale e in formato mnemonico tramite i relativi caratteri della tabella ASCII.

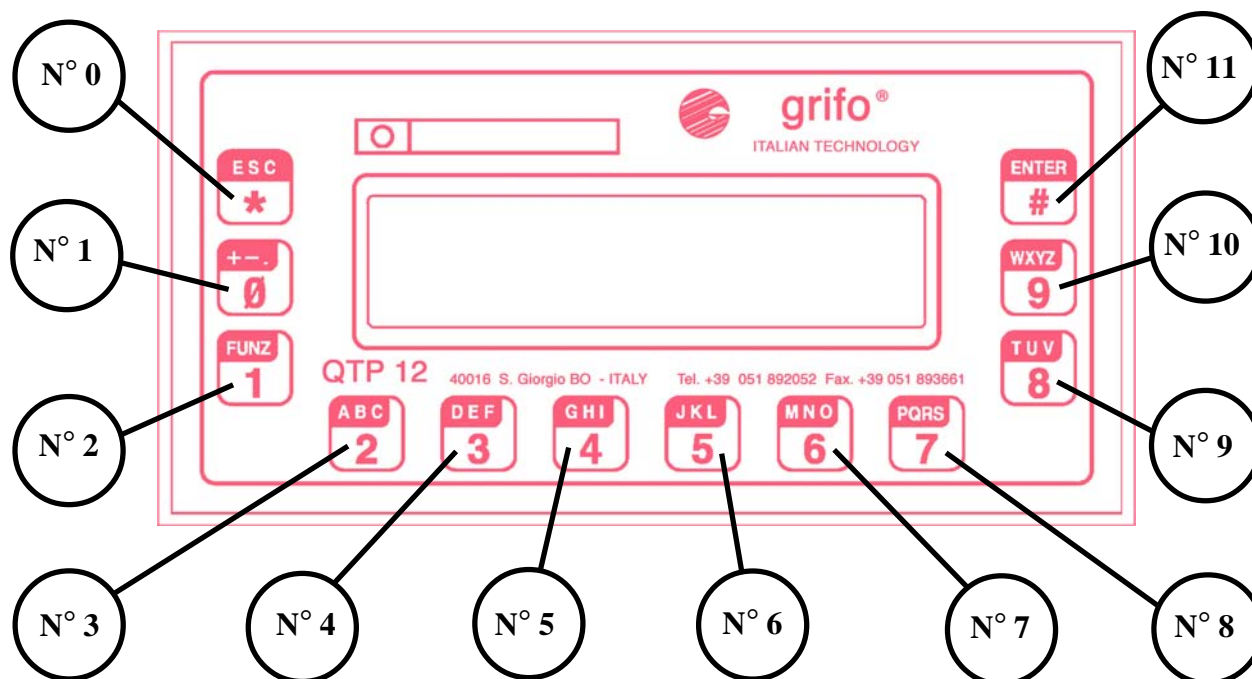


FIGURA 39: NUMERAZIONE E DISPOSIZIONE TASTI

N° TASTO	SERIGRAFIA	CODICE	CODICE HEX	MNEMONICO
0	* ESC	42	2A	*
1	0 +-.	48	30	0
2	1 FUNZ	49	31	1
3	2 ABC	50	32	2
4	3 DEF	51	33	3
5	4 GHI	52	34	4
6	5 JKL	53	35	5
7	6 MNO	54	36	6
8	7 PQRS	55	37	7
9	8 TUV	56	38	8
10	9 WXYZ	57	39	9
11	# ENTER	35	23	#

FIGURA 40: CODICI DI DEFAULT DEI TASTI

Tali codici sono quelli restituiti nella configurazione di default ricevuta all'acquisto, ma l'utente li può comodamente variare utilizzando gli appositi comandi; in questo modo viene notevolmente semplificata la realizzazione del programma di gestione che usa la **QTP 12/R84**.

FIRMWARE SERIALE .SER

Con questo firmware la **QTP 12/R84** opera come un completo terminale video in cui tutto quanto ricevuto dalla seriale, se non si tratta di un comando, viene visualizzato sul display, e tutti i codici dei tasti esterni premuti, vengono trasmessi in seriale all'unità master di controllo. L'utente non può intervenire sul codice eseguito dalla **QTP**, che viene quindi usata come un prodotto chiuso e finito, ma solo sul programma dell'unità master di controllo (P.C., PLC, scheda con CPU, ecc.) che comunica con la **QTP** tramite la linea seriale; su quest'ultima vengono naturalmente scambiati anche i parametri e risultati della ricca e completa serie di comandi.

Inoltre il *firmware seriale* può utilizzare due modalità di comunicazione definite normale e master slave; la prima si presta a collegamenti punto-punto mentre la seconda, quando abbinata ad un adeguato protocollo elettrico (RS 422, RS 485, Current loop), consente di collegare più **QTP** in rete e di colloquiare con unità dello stesso e/o diverso tipo, in una modalità comoda ed efficiente.

Visto che il firmware seriale è completamente basato sulla comunicazione seriale sul pannello operatore è disponibile un programma di set up locale, che permette di preimpostare il protocollo di comunicazione, utilizzando la tastiera ed il display della **QTP 12/R84.SER**.

Nei cinque paragrafi successivi sono riportate le caratteristiche fondamentali del *firmware seriale*.

SETUP LOCALE CON FIRMWARE SERIALE

Un'apposita modalità di setup locale consente di settare la funzione di keyclick, l'inizializzazione dei dati in EEPROM ed il protocollo fisico di comunicazione, utilizzando solo 4 tasti.

Per accedere a questa modalità di funzionamento è necessario fornire alimentazione al terminale, tenendo premuti contemporaneamente il tasto numero **0** e numero **1** (quelli serigrafati con * e 0), per un tempo di circa mezzo secondo.

A questo punto sul display viene visualizzato la stringa ***** Local Setup ***** e tramite i tasti numero **10** e **11** (quelli serigrafati con 9 e #) sarà possibile variare i vari parametri di configurazione, come di seguito riportato:

Tasto N° 11 (#): Permette di scorrere i vari menù, visualizzando i seguenti messaggi:

"COMMUNICATION"	variazione del tipo di comunicazione
"BAUD RATE"	variazione del Baud Rate di comunicazione
"STOP BIT"	variazione del numero di bit di Stop
"KEY-CLICK"	settaggio della funzione di keyclick
"NAME (Hex)"	prima cifra del Nome di identificazione in esadecimale
"NAME (Hex)"	seconda cifra del Nome di identificazione in esadecimale
"EEPROM DATA"	inizializza dati salvati in EEPROM
"SAVE and EXIT"	uscita dal setup

Tasto N° 10 (9): Permette di variare il valore relativo al menù visualizzato:

COMMUNICATION:	Norm. o M.-S. protocollo normale o master slave	(def=Norm.)
BAUD RATE:	38400, 19200, 9600, 4800, 2400 o 1200 Baud	(def=19200)
STOP BIT:	1 o 2 con protocollo normale	(def=1)
	1 con protocollo master slave	
KEYCLICK:	ON o OFF	(def=ON)
NAME (Hex):	Varia cifra indicata tra "> <" nel campo 0÷F Hex	(def=80H)
EEPROM DATA	NOINI o INIT	(def=NOINI)
SAVE and EXIT	Esce da setup e configura la QTP 12/R84 con i parametri selezionati	

Una volta usciti le opzioni selezionate sono salvate in EEPROM e mantenute fino alla successiva esecuzione del setup locale; subito dopo il terminale ritorna nella normale modalità di funzionamento. I valori di default riportati tra parentesi sono quelli settati alla fine della fase di collaudo, ovvero quelli impostati sulla **QTP 12/R84.SER** ricevuta dall'utente.

Le opzioni disponibili per i menù BAUD RATE e STOP BIT definiscono il protocollo fisico di comunicazione che ha gli altri due parametri fissi a nessuna parità ed 8 bit per carattere (o 9 bit se master slave). Le opzioni dei rimanenti menù sono invece descritte in alcuni paragrafi precedenti e successivi.

N.B.

Alla modalità di set up locale si può accedere solo all'atto dell'accensione qualora siano verificate le condizioni sopra descritte; durante il normale funzionamento della **QTP 12/R84.SER** con la pressione di questi tasti non si attiva il set up ma viene semplicemente trasmesso il relativo codice sulla seriale.

Il setup locale normalmente é eseguito solo una volta dopo la prima installazione, da parte di personale esperto, quindi non riguarda l'utente finale che usa la **QTP 12/R84.SER** solo come interfaccia operatore.

MODALITA' DI COMUNICAZIONE CON FIRMWARE SERIALE

La **QTP 12/R84.SER** prevede due diverse modalità di comunicazione tramite la sua linea seriale asincrona:

- Norm. la comunicazione avviene con 8 bit per carattere, nessuna parità e con stop bit e baud rate selezionabili dall'utente tramite il set up locale. Tale comunicazione é adatta a collegamenti punto punto in RS 232, RS 422 e Current loop.
- M.-S. la comunicazione avviene con 9 bit per carattere, nessuna parità, uno stop bit e baud rate selezionabile dall'utente tramite il set up locale. Tale comunicazione é adatta a collegamenti punto punto (con tutti i protocolli elettrici) o in rete (con i protocolli RS 422, RS 485 e Current loop). Per maggiori informazioni sulla modalità master slave fare riferimento al paragrafo successivo.

La selezione della modalità di comunicazione é gestita nel set up locale, come descritto nell'omonimo paragrafo, mentre l'interfaccia elettrica deve essere definita in fase di ordine del terminale oppure variata seguendo le indicazioni del paragrafo CONFIGURAZIONE LINEA SERIALE.

COMUNICAZIONE MASTER SLAVE CON FIRMWARE SERIALE

La modalità master slave, sfrutta la tecnica di comunicazione a 9 bit. In particolare oltre agli 8 bit di dati, viene gestito un nono bit che serve a distinguere una chiamata, da parte della apparecchiatura **master** ad una delle strutture **slave**, da un normale passaggio di informazioni tra il master e il dispositivo attualmente selezionato.

Quando il nono bit é posto a 1, gli 8 bit di dati dello stesso carattere devono contenere il Nome di identificazione del dispositivo con il quale si vuole comunicare, mentre ponendo questo particolare bit a 0 é possibile comunicare le informazioni al/dal dispositivo selezionato.

Nel caso particolare della comunicazione con la **QTP 12/R84.SER**, il Nome di identificazione deve essere quello settato tramite il programma di setup locale del terminale stesso, nei menù “NAME (Hex)”.

Quando viene inviato questo byte (con il nono bit posto a 1), la **QTP 12/R84.SER** si riconosce e si pone in attesa della stringa contenente caratteri, dati o comandi. In questa sequenza, può esserci solo un comando che comporta la restituzione di una risposta in seriale da parte della **QTP**; se ve ne è un numero superiore, le risposte ai restanti comandi verranno ignorate.

Con il protocollo master slave tra la trasmissione di un carattere ed il successivo, non deve trascorrere un tempo superiore al tempo di **Time Out**, in quanto, trascorso questo ritardo, la **QTP 12/R84.SER** considera finita la sequenza di comandi ed inizia la fase di risposta. I tempi di Time Out relativi ai vari Baud Rate sono indicati di seguito:

Baud Rate	Time Out	Tempo trasmissione carattere
38400 Baud	550 µsec	287 µsec
19200 Baud	990 µsec	573 µsec
9600 Baud	1540 µsec	1146 µsec
4800 Baud	3080 µsec	2292 µsec
2400 Baud	6105 µsec	4584 µsec
1200 Baud	12100 µsec	9167 µsec

Il master dopo aver completato la trasmissione dell'ultimo carattere della sequenza di comando, dovrà attendere un tempo di:

“tempo trasmissione carattere”+Time Out

prima che arrivi il primo carattere della sequenza di risposta, trasmessa dalla **QTP 12/R84.SER**. Tale risposta consiste in un carattere contenente il codice del tasto premuto (**FF Hex**, indica nessun tasto premuto), oppure una sequenza di uno o più caratteri che coincide con la risposta al comando inviato nella chiamata precedente. Da ricordare che la risposta viene restituita anche in caso di stringhe di comando con il solo Nome di identificazione, in modo da semplificare la verifica dei tasti premuti, o di comando non valido.

In abbinamento alla **QTP 12/R84.SER** vengono forniti dei programmi dimostrativi, codificati in diversi linguaggi, che implementano la comunicazione con protocollo master slave e che possono essere direttamente usati dall'utente oppure modificati a seconda delle proprie esigenze.

Quando il sistema master è un PC, l'utente può far uso anche di comode librerie **DLL** che consentono di gestire la comunicazione master slave ad alto livello, ovvero senza preoccuparsi della gestione del nono bit, delle tempistiche, di eventuali convertitori di protocollo elettrico, ecc. Anche queste librerie sono fornite in corrispondenza del primo acquisto, assieme alla relativa documentazione d'uso, salvate su un disco o su un CD.

NOTE:

- 1) Tra una chiamata e la successiva, per avere la certezza che il comando trasmesso sia correttamente eseguito, è necessario attendere un tempo che è funzione del numero di comandi inviati e del tipo di operazioni che questi comportano.
- 2) Se l'unità master di controllo non è in grado di dialogare a 9 bit, è possibile simulare questo tipo di comunicazione, sfruttando il bit di parità e programmando, prima di trasmettere ogni singolo byte, la parità pari o dispari, secondo quanto indicato di seguito:

Il Byte da trasmettere ha un numero PARI di bit a 1

Se il Bit 9 deve essere **1** -> Programmare la parità **DISPARI**

Se il Bit 9 deve essere **0** -> Programmare la parità **PARI**

Il Byte da trasmettere ha un numero DISPARI di bit a 1

Se il Bit 9 deve essere **1** -> Programmare la parità **PARI**

Se il Bit 9 deve essere **0** -> Programmare la parità **DISPARI**

- 3) Se é attivata la funzione di scorrimento dei messaggi, il tempo che deve trascorrere fra una chiamata e la successiva, oltre a quanto detto nella nota **1**, deve esseredi circa **12000 µsec**

Per chiarire meglio il protocollo master slave viene di seguito riportato un esempio di gestione in cui l'unità master fornisce tre comandi alla **QTP 12/R84.SER** (richiesta versione, rappresentazione stringa e verifica eventuali tasti premuti) con un Baud Rate di comunicazione di 38,4 KBAud e Nome di identificazione settato a 80H:

<i>Master</i>	<i>QTP 12/R84.SER</i>
Trasmette comando di lettura numero di versione, ovvero trasmette la sequenza di caratteri: 80H con nono bit a 1 1BH con nono bit a 0 56H con nono bit a 0 con ritardo tra i caratteri inferiore a 550 µsec	Riceve caratteri del comando e ne verifica il termine con il Time Out di 550 µ sec
Attende tempo di 837 µsec	Riconosce sequenza di comando, la esegue e salva risposta per successivo comando
Riceve risposta di un carattere	Trasmette risposta che coincide con il codice dell'eventuale tasto premuto, con nono bit a 0
Trasmette comando con stringa da rappresentare, ovvero trasmette la sequenza di caratteri: 80H con nono bit a 1 1° carattere stringa con nono bit a 0 2° carattere stringa con nono bit a 0 : : : : : : con ritardo tra i caratteri inferiore a 550 µsec	Riceve caratteri del comando e ne verifica il termine con il Time Out di 550 µ sec
Attende tempo di 837 µsec	Riconosce sequenza di comando e rappresenta i caratteri della stringa ricevuti
Riceve risposta di tre caratteri con il numero di versione richiesto nel comando precedente	Trasmette risposta salvata che coincide con il numero di versione richiesto nel comando precedente, con nono bit a 0
Trasmette comando per verifica pressione tasto, ovvero trasmette la sequenza di caratteri: 80H con nono bit a 1	Riceve caratteri del comando e ne verifica il termine con il Time Out di 550 µ sec
Attende tempo di 837 µsec	Riconosce sequenza senza comando e quindi non effettua operazioni
Riceve risposta di un carattere che coincide con il codice dell'eventuale tasto premuto	Trasmette risposta che coincide con il codice dell'eventuale tasto premuto, con nono bit a 0

FIGURA 41: ESEMPIO COMUNICAZIONE CON PROTOCOLLO MASTER SLAVE

COME INIZIARE CON FIRMWARE SERIALE

In questo paragrafo vengono illustrate le operazioni da effettuare per iniziare ad usare la **QTP 12/R84.SER** in maniera rapida e lineare, senza dover affrontare e risolvere alcun problema iniziale. Al fine di rendere fruibili queste istruzioni a qualsiasi utente si seleziona come unità master di controllo un normale Personal Computer provvisto di una linea seriale RS 232 libera ed un generico sistema operativo, fino a Windows 98.

A) Collegamento seriale tra **QTP12/R84.SER** e P.C.:

- A1) Realizzare il collegamento seriale descritto nella figura 40 ovvero collegare i due segnali di comunicazione (TX RS232, RX RS232) e la massa di riferimento (GND) ad una porta di comunicazione COMx libera, del P.C. Come si può notare tale cavo di collegamento é rovesciato e per praticità può essere ordinato alla **grifo®** specificando il codice CCR 9+9R.

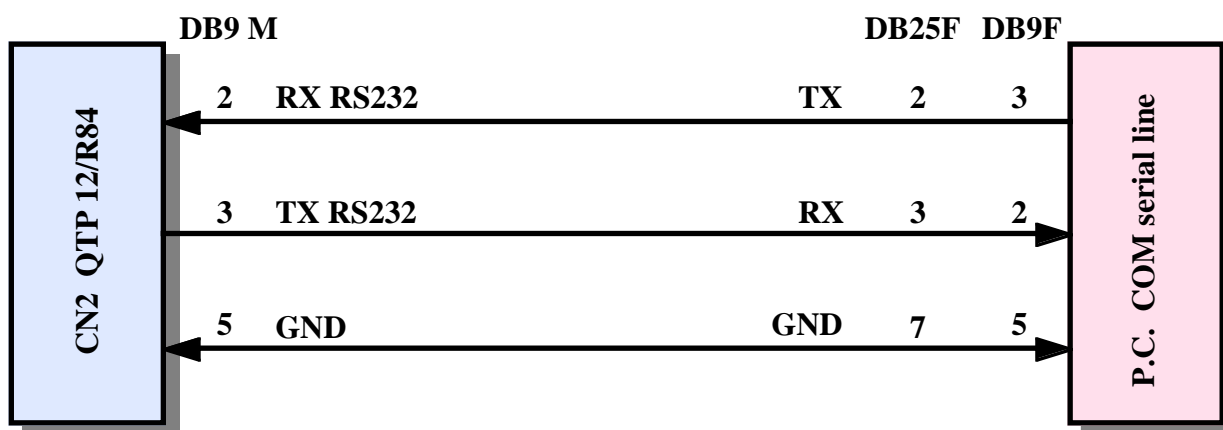


FIGURA 42: COLLEGAMENTO RS 232 CON P.C. E FIRMWARE SERIALE

- A2) Fornire alimentazione su CN1 e verificare che il buzzer si disattivi e che sul display compaia il cursore lampeggiante nell'angolo in alto a sinistra.
- A3) Premere alcuni pulsanti della **QTP** e verificare che venga emesso il relativo keyclick acustico dal buzzer interno.

B) Uso programma demo:

- B1) Su dischetti o sul CD **grifo®** ricevuto in caso di primo acquisto è disponibile il file PRQTP12R.EXE, che contiene il codice eseguibile del demo per il *firmware seriale*. Tale file, una volta localizzato, deve essere copiato assieme agli altri della cartella in una posizione comoda sul disco rigido del P.C. utilizzato.
- B2) Eseguire il programma copiato al punto B1, rispondere alle sue domande iniziali selezionando una comunicazione Normale ed il tipo di display montato. A questo punto premere un tasto per continuare senza eseguire il set up locale che infatti coincide con quello di default già impostato sulla **QTP 12/R84.SER** ricevuta.
- B3) Proseguire l'esecuzione del programma dimostrativo verificando che le indicazioni riportate sul monitor siano effettivamente eseguite sulla **QTP**; quando richiesto interagire con lo stesso programma in modo da provare tutti i comandi messi a disposizione, fino al raggiungimento della sua fine.

C) Uso emulazione terminale:

- C1) Individuare sul P.C. utilizzato il programma di comunicazione HYPERTERMINAL che normalmente si trova nel menù di Windows: "Start | Programmi | Accessori | Comunicazioni" e provvedere a lanciarlo.

- C2) Tramite le finestre di impostazione di HYPERTERMINAL impostare la comunicazione a:
- | | |
|---------------------|---|
| Connetti | direttamente a COM x (quella usata al punto A1) |
| Bit per secondo | 19200 |
| Bit di Dati | 8 |
| Parità | Nessuna |
| Bit di Stop | 1 |
| Controllo di flusso | Nessuno |
- ed attendere la comparsa della finestra di comunicazione.
- C3) A questo punto digitare qualcosa sulla tastiera del P.C. e verificare che quanto premuto compaia sul display della **QTP 12/R84.SER** e che premendo i tasti della **QTP** il relativo codice appaia sul monitor del P.C. Per completezza si possono provare gli effetti anche dei comandi provvedendo a digitare sempre sulla tastiera del P.C., le relative sequenze di codici (questa operazione é semplificata dalla pressione contemporanea del tasto ALT e del codice decimale sul tastierino numerico: ad esempio per trasmettere il codice 12 del comando di cancellazione pagina si deve premere il tasto ALT e contemporaneamente i tasti 1 e 2).

Se quanto descritto non avviene si deve riverificare la connessione seriale e, tramite il setup locale, assicurare che sia impostato il protocollo fisico di default.

PROGRAMMI DEMO PER FIRMWARE SERIALE

In caso di primo acquisto sul dischetto o sul CD **grifo®** ricevuto sono disponibili numerosi programmi dimostrativi che consentono di provare e valutare immediatamente il prodotto ricevuto. Tali programmi sono forniti in formato eseguibile e sorgente e sono disponibili per numerosi linguaggi di programmazione ad alto livello (C, PASCAL, BASIC, ecc.) sia per P.C. che per le schede a microprocessore **GPC®** della **grifo®**.

Come indicato nel paragrafo COME INIZIARE CON FIRMWARE SERIALE i programmi con il nome PRQTP12R.* utilizzano tutti i comandi disponibili con una semplice iterazione con l'utente ma ne vengono forniti numerosi altri in grado ad esempio di: comandare la **QTP** collegata su una rete seriale, gestire sensori 1-WIRE, gestire i messaggi, usare la modalità master slave con librerie DLL, ecc. L'utente può esaminare i commenti di tali esempi e decidere autonomamente se provarli.

Tutti i programmi dimostrativi possono essere usati direttamente oppure modificati od utilizzati in parte, a seconda delle proprie esigenze, senza alcuna autorizzazione o costo aggiuntivo. In caso di particolari esigenze o combinazioni d'uso possono essere anche richiesti dei demo specifici, previo accordo con la **grifo®**.

FIRMWARE DI LIBRERIA .LIB

Con questo firmware la **QTP 12/R84** opera come un potente controllore di processo completo di interfaccia operatore che può quindi funzionare sia autonomamente che abbinato ad altri sistemi. L'utente deve intervenire sul codice eseguito dalla **QTP**, in modo da realizzare un programma applicativo che soddisfi le sue esigenze, ed in questo è facilitato dalla ricca e completa serie di comandi che possono essere chiamati direttamente con i relativi parametri e risultati. Tali comandi soddisfano le normali richieste del settore industriale e sono dettagliatamente descritti nell'anonimo capitolo. La **QTP 12/R84.LIB** non è quindi un prodotto finito pronto per essere installato ma deve essere prima specializzato dall'utente. Questa specializzazione può essere effettuata con comodi ed economici ambienti di sviluppo, sia ad alto che a basso livello, e rendono la **QTP** un prodotto veramente flessibile e versatile. Infatti il programma applicativo dell'utente che la specializza consente di risolvere ogni problematica anche di alta complessità, ed allo stesso tempo consente di realizzare automazioni diverse, usando lo stesso hardware.

A differenza del *firmware seriale* quello di *libreria* non usa la linea seriale asincrona della **QTP**: in questo modo il programma applicativo può liberamente colloquiare con altri dispositivi sia in modalità punto punto che in rete, con qualsiasi protocollo logico. Tramite l'ambiente di sviluppo scelto sulla linea seriale si può inoltre effettuare il debug del programma applicativo, riducendo i tempi complessivi di preparazione.

Nei sei paragrafi successivi sono riportate le caratteristiche fondamentali del *firmware di libreria*.

INTEGRAZIONE ED USO DEL FIRMWARE DI LIBRERIA

Il *firmware di libreria* è stato progettato prefissando i seguenti obiettivi:

- poter essere abbinato a tutti i linguaggi di programmazione disponibili;
- ridurre al minimo le risorse hardware usate;
- mantenere la compatibilità d'uso con gli altri firmware delle **QTP**;
- prevedere un facile modalità di chiamata ai comandi ed interscambio parametri;
- coprire le normali e più frequenti necessità del settore industriale;

che hanno definito le modalità di integrazione ed uso dello stesso firmware, all'interno del programma applicativo dell'utente.

Per chiarezza si ricorda che l'utente del *firmware di libreria* deve avere una conoscenza base del microcontrollore usato e della realizzazione di software embedded in quanto la successiva documentazione non fornisce, ma usa, queste informazioni. Tali conoscenze possono essere acquisite dalla lettura del data sheet, riportato in APPENDICE D del manuale.

In questo paragrafo sono riportate tutte le informazioni generali sull'integrazione e l'uso del *firmware di libreria* che possono essere utilizzate da ogni utente, con qualsiasi ambiente di sviluppo fornito da **grifo®** o da terze parti.

Per integrare ed usare il *firmware di libreria* sono necessari alcuni strumenti hardware e software che vengono opportunamente indicati nella successiva descrizione. La documentazione completa di tali strumenti è disponibile all'interno degli stessi e non viene quindi riportata in questo manuale; tra questi si ricorda il P.C. di sviluppo che coincide con un normale Personal Computer provvisto di una linea seriale RS 232 libera ed un generico sistema operativo, a partire da Windows 95.

Concludendo le operazioni necessarie per integrare ed usare il *firmware di libreria* sono:

- a) Installare l'ambiente di sviluppo prescelto per realizzare il programma applicativo, sul P.C. di sviluppo. In generale la scheda può sfruttare tutti gli ambienti per il microprocessore montato, ovvero i numerosi pacchetti ideati per la famiglia 51, sia ad alto che a basso livello (assemblatori, compilatori, interpreti, ecc.). Tutti i pacchetti di sviluppo software forniti dalla **grifo®** sono sempre accompagnati dagli elementi che integrano completamente il firmware rendendolo pronto all'uso.
- b) Installare l'ambiente di programmazione ISP (In System Programming) sul P.C. di sviluppo, ovvero il programma FLIP che comunicando con il boot loader del microcontrollore, attraverso la porta seriale, permette di leggere, cancellare e riscrivere la memoria FLASH EPROM. La programmazione ISP riduce i costi ed i tempi di sviluppo dell'applicazione, infatti elimina la necessità di usare EPROM esterne, programmatori, cancellatori, ecc. Per ulteriori informazioni sulla programmazione ISP si prega di consultare la specifica documentazione tecnica rilasciata dalla ATMEL.
- c) Predisporre l'ambiente di sviluppo in modo che nel programma applicativo generato siano preservate le risorse hardware usate dal *firmware di libreria*; come illustrato nelle figure 41, 42 e nel successivo paragrafo RISORSE USATE DAL FIRMWARE DI LIBRERIA, il programma applicativo non deve usare l'area finale della FLASH, alcune aree di RAM interna, un timer counter, la EEPROM di bordo, ecc.

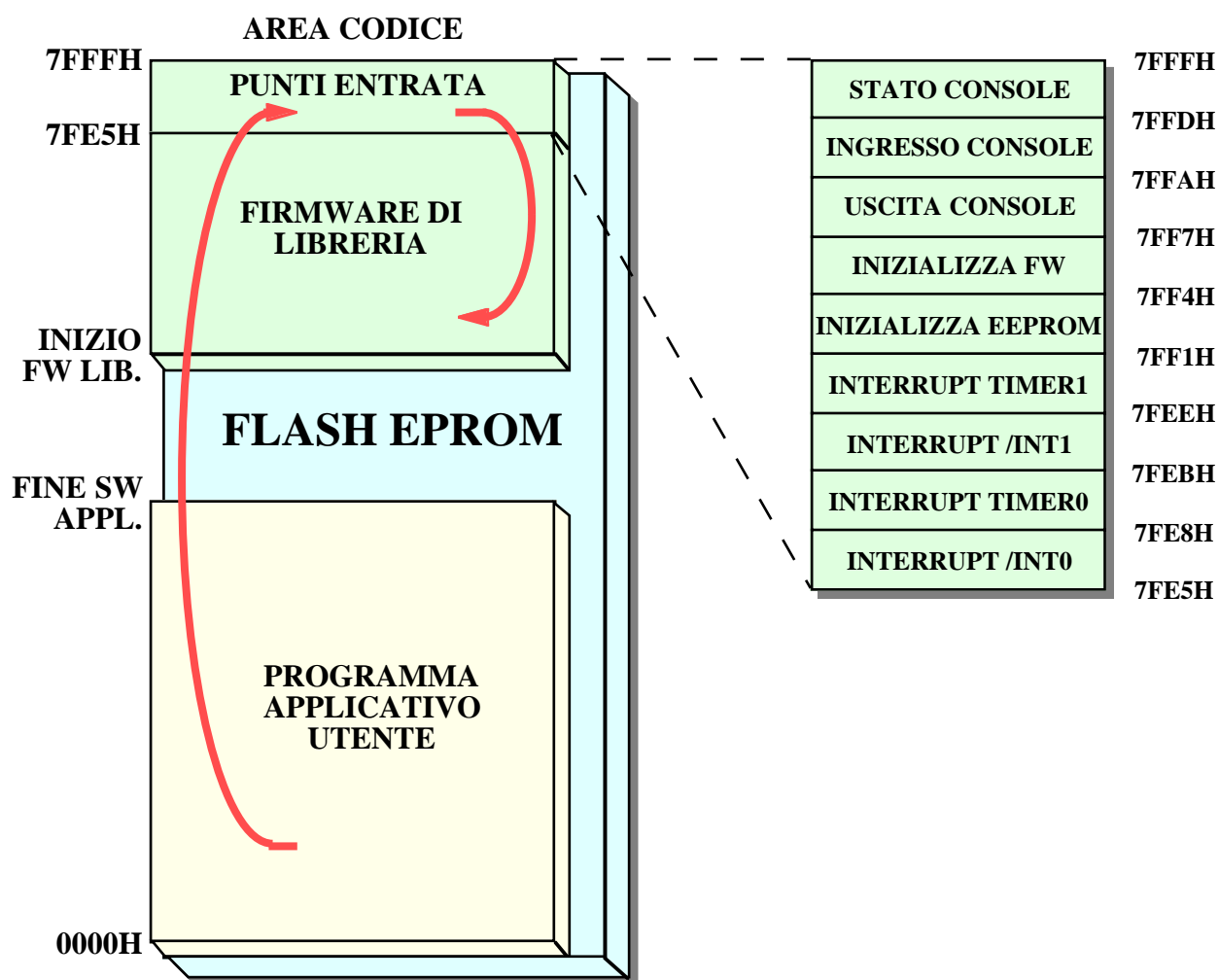


FIGURA 43: ORGANIZZAZIONE AREA CODICE CON FIRMWARE DI LIBRERIA

d) Fisicamente il *firmware di libreria* coincide con un codice eseguibile che deve essere salvato al termine della area codice del microcontrollore come indicato in figura 41. Tale codice viene fornito nel file QTP12Rxx.HEX che, grazie al suo formato HEX, può essere utilizzato direttamente per la programmazione della FLASH. In quest'ultima memoria oltre al firmware descritto si dovrà salvare anche il codice eseguibile del programma applicativo utente che deve essere naturalmente allocato all'inizio dell'area codice in modo che venga immediatamente eseguito a seguito di un'accensione o di un reset. Il passaggio dal programma applicativo al *firmware di libreria* é effettuato tramite un'opportuna tabella di entrata, allocata ad indirizzi fissi, che funge da ponte di collegamento tra i due codici presenti nell'unica area codice.

La scelta degli indirizzi di allocazione delle tre aree in FLASH é stata effettuata attentamente in modo da avere il massimo spazio libero per il programma applicativo e di avere dei punti di entrata che rimangono inalterati anche in caso di aggiornamenti ed ampliamenti del *firmware di libreria*. Così facendo l'utente può usare una nuova versione di firmware semplicemente riprogrammandolo nella FLASH, senza intervenire sul suo programma applicativo.

e) Il valore di INIZIO FW LIB. é stabilito dal *firmware di libreria* stesso e quindi varia al variare della sua versione; attualmente con la versione 1.3 é fissato a 6F00H e comunque può essere facilmente determinato caricando il file QTP12Rxx.HEX (dove xx corrisponde al numero di versione) ed esaminando il suo indirizzo di inizio.

L'utente a seguito di ogni realizzazione del programma applicativo deve sempre verificare che il relativo indirizzo di fine FINE SW APPL. sia inferiore all'INIZIO FW LIB. ovvero che i due codici non si sovrappongano. Questa verifica può essere effettuata facilmente infatti normalmente tutti gli ambienti di sviluppo (assemblatori, compilatori, linguaggi, ecc.) forniscono indicazioni sulle dimensioni del codice generato e sarà quindi sufficiente confrontare tali indicazioni con il valore di INIZIO FW LIB. descritto prima.

f) Ridirezionare i vettori delle procedure di risposta agli interrupts /INT0, TIMER0, /INT1, TIMER1 del microcontrollore, ai punti di entrata del *firmware di libreria* indicati in figura 41. Le ridirezioni descritte devono essere realizzate seguendo le regole dell'ambiente di sviluppo e normalmente coincidono con delle istruzioni di salto assoluto (ad esempio L JMP 7FE5H) all'indirizzo di entrata, poste nella procedura di risposta al relativo interrupt. Si ricorda che dei 4 interrupt da ridirezionare solo quello del TIMER0 é sempre necessario mentre i rimanenti 3 solo in caso d'uso dei relativi comandi, come descritto nel successivo paragrafo RISORSE USATE DAL FIRMWARE DI LIBRERIA.

g) Al fine di semplificare l'uso del firmware, di tutti i suoi comandi ed il passaggio parametri e risultati sono stati previsti tre procedure, con altrettanti punti di entrata, con le seguenti caratteristiche:

STATO CONSOLE: restituisce lo stato di presenza di un dato che il firmware deve fornire al programma applicativo; il dato può essere il codice di un tasto premuto oppure la risposta ad un comando inviato. La procedura non ha dati d'ingresso ed un solo dato d'uscita salvato nell'accumulatore che coincide con il numero di caratteri pronti per essere forniti al programma applicativo. Tale numero di caratteri funziona anche da stato infatti se azzerato non vi sono dati e viceversa.

INGRESSO CONSOLE: attende la disponibilità di un dato che il firmware fornisce al programma applicativo e lo restituisce; anche per questa procedura il dato può essere il codice di un tasto premuto oppure la risposta ad un comando inviato. La procedura non ha dati d'ingresso ed un solo dato d'uscita salvato nell'accumulatore che coincide con il dato descritto.

USCITA CONSOLE: invia un dato dal programma applicativo al firmware; il dato può essere un carattere da rappresentare sul display oppure un comando da inviare od i suoi eventuali parametri. La procedura ha un solo dato d'ingresso salvato nell'accumulatore che coincide con il dato descritto e non ha dati d'uscita.

Nel programma applicativo il *firmware di libreria* viene usato con le tre procedure descritte; se le procedure di console, che hanno già una struttura compatibile, sono ridirezionate su queste procedure si ottiene una notevole semplificazione ed una impareggiabile potenzialità d'uso (questo spiega il suffisso **CONSOLE** che é stato attribuito alle stesse procedure). Così le istruzioni ad alto livello PRINT, PRINTF, KBHIT, SCANF, INPUT, ecc. di un ambiente di sviluppo in C o BASIC, chiamano automaticamente le tre procedure di entrata del *firmware di libreria* consentendo di utilizzare tutte le loro possibilità.

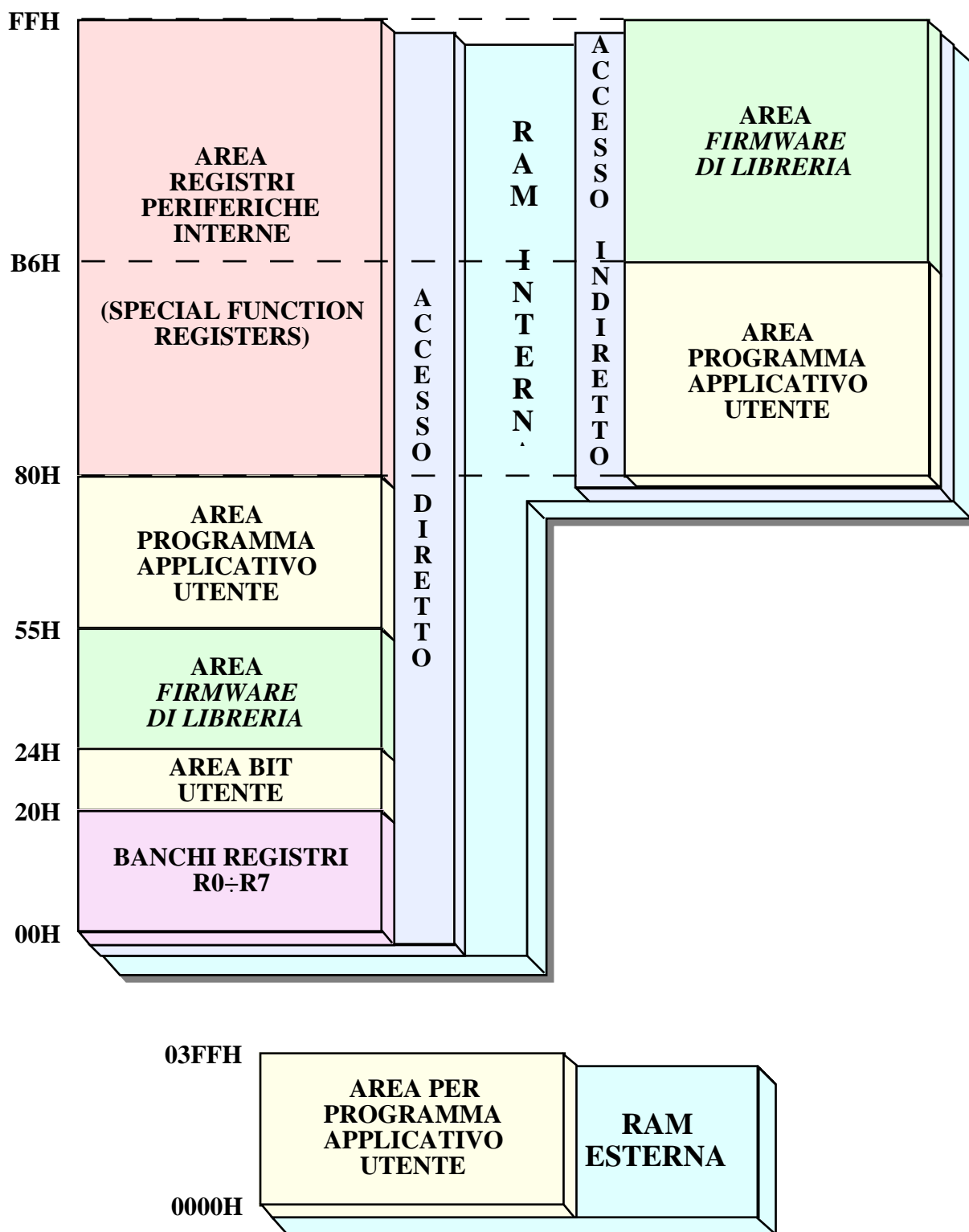


FIGURA 44: USO RAM CON FIRMWARE DI LIBRERIA

g) A seguito di un reset o di un'accensione il programma applicativo utente deve predisporre il *firmware di libreria* alle successive operazioni. Per queste inizializzazioni sono disponibili due procedure, con altrettanti punti di entrata, con le seguenti caratteristiche:

INIZIALIZZA FW: effettua tutte le operazioni di inizializzazione necessarie come: settaggio variabili; azzeramento buffer; disattivazione di buzzer, LED di stato, uscite digitali, contatori; inizializzazione e cancellazione display; settaggio cursore blocco lampeggiante, nella posizione di Home; caricamento dei caratteri utente; settaggio del keyclick impostato; attivazione scansione tastiera; attivazione funzioni temporali; ecc. La procedura non ha dati d'ingresso ed uscita.

INIZIALIZZA EEPROM: inizializza la EEPROM con i dati di default descritti nel paragrafo DATI IN EEPROM e poi effettua tutte le operazioni di inizializzazione descritte per la procedura INIZIALIZZA FW. La procedura ha un solo dato d'ingresso salvato nell'accumulatore che coincide con il tipo di display montato e non ha dati d'uscita. La codifica del dato d'ingresso è la seguente:

0	->	Display LCD alfanumerico 20x2	(QTP 12/R84-C2.LIB)
1	->	Display VFD alfanumerico 20x2	(QTP 12/R84-F2.LIB)
2	->	Display VFD grafico 140x16	(QTP 12/R84-GF2.LIB)

Si ricorda che la durata dell'esecuzione di questa procedura è di circa 20 secondi.

Normalmente queste procedure devono essere chiamate una sola volta all'inizio del programma applicativo, usando le modalità dell'ambiente di sviluppo scelto che normalmente coincidono con una istruzione di chiamata assoluta (ad esempio LCALL 7FF4H) all'indirizzo di entrata, preceduta da un eventuale settaggio dei dati d'ingresso. La procedura INIZIALIZZA EEPROM deve essere chiamata una sola volta per salvaguardare la durata dell'EEPROM su cui scrive: i suoi usi tipici sono quelli in corrispondenza della prima installazione oppure quando si devono ripristinare i settaggi iniziali a causa di modifiche indesiderate.

h) Una volta realizzato il programma utente che usa il *firmware di libreria* con le caratteristiche descritte nei punti precedenti lo si deve salvare nella FLASH EPROM della **QTP 12/R84.LIB** assieme allo stesso *firmware di libreria*, come descritto nel paragrafo PROGRAMMAZIONE FLASH EPROM.

i) A questo punto la **QTP 12/R84.LIB** è completa e pronta ad essere usata e testata all'interno dell'applicazione utente. Il debug del programma applicativo ottenuto può essere effettuato con gli strumenti messi a disposizione dall'ambiente di sviluppo usato, ricordando che la linea seriale (non usata dal *firmware di libreria*) è un ottimo candidato per questa funzione.

RISORSE USATE DA FIRMWARE DI LIBRERIA

Il *firmware di libreria* dispone di comandi che consentono di gestire facilmente la maggioranza delle risorse della scheda come display, tastiera, buzzer, LED, uscite digitali, ingressi digitali, memoria EEPROM, interfaccia 1-WIRE, ecc. Tali comandi però utilizzano una serie di risorse hardware aggiuntive della **QTP 12/R84** che sono brevemente descritte in questo paragrafo, assieme alle limitazioni d'uso, da parte del programma applicativo utente:

Area codice in FLASH EPROM del microcontrollore: coincide con la porzione finale della FLASH EPROM, già descritta nei punti (d), (e) del paragrafo precedente. Su tale area viene salvato il codice del *firmware di libreria* e non deve essere assolutamente usata dal programma applicativo, pena il malfunzionamento di tutto il sistema. Alcuni ambienti di sviluppo (come il BASCOM 8051) possono essere configurati per avvisare l'utente in caso di generazione di un codice che supera un limite impostato (FINE SW APPL. >= INIZIO FW LIB.).

Aree dati in RAM interna del microcontrollore:

coincide con due aree di RAM interna

di cui la prima allocata nell'area ad accesso diretto e la seconda ad accesso indiretto, in cui sono salvate tutti i flag, le variabili ed i buffer del *firmware di libreria*. Queste aree sono allocate agli indirizzi fissi descritti in figura 42 e non devono essere assolutamente usate dal programma applicativo, pena il malfunzionamento di tutto il sistema.

La salvaguardia d'uso di queste memorie é realizzata seguendo le regole dell'ambiente di sviluppo usato e normalmente avviene con direttive per il compilatore, settaggi dell'eventuale progetto, dichiarazione di variabili che vengono solo allocate ma mai usate, utilizzo di codici di partenza (start up) utente, ecc. Le finestre con i risultati della generazione del programma utente normalmente proposte dall'ambiente di sviluppo, consentono di verificare facilmente la salvaguardia di tali aree, evitando di scoprire i conseguenti malfunzionamenti in fase di prova.

Si ricorda che la scelta d'uso della memoria del *firmware di libreria* é stata effettuata cercando di lasciare comunque a disposizione tutti i tipi di memoria del microcontrollore, ovvero: 4 byte indirizzabili a bit per un totale di 32 bit utente, 43 byte ad accesso diretto, 48 byte ad accesso indiretto ed infine 1024 byte di RAM esterna. Inoltre la completezza dei comandi offerti dal firmware riduce drasticamente la necessità di memoria e variabili nel programma applicativo utente, tanto che spesso si riduce allo stack ed a poche variabili di lavoro.

Area di stack del microcontrollore:il *firmware di libreria* non ha un proprio

stack e quindi usa quello del programma applicativo. L'utente in fase di configurazione dell'ambiente di sviluppo scelto deve tenere conto della profondità di stack necessaria al firmware, che nella condizione peggiore può arrivare a 17 bytes.

Contatore temporizzatore TIMER0 del microcontrollore: per gestire tutti i processi temporali il *firmware di libreria* necessita di un interrupt periodico generato dal TIMER0 del microcontrollore.

Il programma applicativo utente non può usare questa risorsa, deve ridirezionare il relativo vettore d'interrupt del microcontrollore (000BH) al punto di entrata (7FE8H) e deve ricordare che una volta inizializzato, il firmware risponde a questo interrupt ogni 2,5 msec con un conseguente rallentamento nell'esecuzione. La riduzione delle prestazioni del programma applicativo dipende dai comandi usati e dai processi in corso ed in alcuni casi può arrivare a decine di msec, come opportunamente segnalato nella descrizione degli stessi comandi.

Contatore temporizzatore TIMER1 del microcontrollore: per gestire i comandi relativi al contatore 1 il *firmware di libreria* usa il TIMER1 del microcontrollore, programmato come contatore hardware con caricamento in interrupt. Se il programma applicativo utente usa il contatore 1 del firmware allora non può usare questa risorsa e deve ridirezionare il relativo vettore d'interrupt del microcontrollore (001BH) al corrispondente punto di entrata (7FEEH).Contatore temporizzatore TIMER2 del microcontrollore: per gestire i comandi relativi al contatore 2 il *firmware di libreria* usa il TIMER2 del microcontrollore, programmato come contatore hardware. Se il programma applicativo utente usa il contatore 2 del firmware allora non può usare questa risorsa.Segnale d'interrupt hardware /INT0 del microcontrollore: per gestire i comandi relativi al contatore 3 il *firmware di libreria* usa il segnale d'interrupt /INT0 del microcontrollore. Se il programma applicativo utente usa il contatore 3 del firmware allora non può usare questa risorsa e deve ridirezionare il relativo vettore d'interrupt del microcontrollore (0003H) al corrispondente punto di entrata (7FE5H).Segnale d'interrupt hardware /INT1 del microcontrollore: per gestire i comandi relativi al contatore 4 il *firmware di libreria* usa il segnale d'interrupt /INT1 del microcontrollore. Se il programma applicativo utente usa il contatore 4 del firmware allora non può usare questa risorsa e deve ridirezionare il relativo vettore d'interrupt del microcontrollore (0013H) al corrispondente punto di entrata (7FEBH).

PROGRAMMAZIONE FLASH EPROM

Come descritto nei paragrafi precedenti il *firmware di libreria* deve essere salvato nella FLASH EPROM del microcontrollore assieme al programma applicativo utente, come illustrato in figura 41. Questo salvataggio avviene tramite una programmazione ISP (In System Programming) che riduce i costi ed i tempi di sviluppo dell'applicazione, infatti elimina la necessità di usare EPROM esterne, programmatori, cancellatori, ecc.

La programmazione ISP avviene tramite un normale P.C. che eseguendo un apposito programma di gestione denominato FLIP (FLexible In system Programming), interagisce con un boot loader presente sul microcontrollore e consente di leggere, cancellare, verificare, programmare sia la memoria FLASH che quella EEPROM. Il tutto avviene tramite un collegamento tra il P.C. e la **QTP 12/R84.LIB** normalmente effettuato con linea seriale RS 232 oppure, in alternativa, con linea CAN (per quest'ultima possibilità contattare direttamente la **grifo®**).

Come descritto nella figura 31 il jumper P1 seleziona il modo operativo della **QTP 12/R84** tra i due disponibili:

P1		Modo operativo
Non connesso	->	Modalità RUN
Connesso	->	Modalità DEBUG

In modalità RUN a seguito di un'accensione parte sempre il programma applicativo salvato in FLASH indipendentemente dalle condizioni esterne, mentre in modalità DEBUG l'accensione provoca l'esecuzione del boot loader del microcontrollore e consente quindi la programmazione ISP. Per ulteriori informazioni sulla programmazione ISP e sull'uso del programma FLIP si prega di consultare la specifica documentazione tecnica rilasciata dalla ATMEL, mentre in questo paragrafo si ricorda solo che:

- nella finestra "Device | Select" selezionare il giusto dispositivo a seconda della **QTP** ordinata, ovvero T89C51AC2 in caso di **QTP 12/R84** e T89C51CC01 in caso di **QTP 12/R84.CAN**;
- nella finestra "Settings | Communication | RS232" selezionare il massimo baud rate disponibile (115200) e ridurlo solo in caso di problemi di comunicazione;
- nella finestra "Buffer | Options" settare a NO la voce "Reset Buffer Before Loading" e selezionare la voce "Whole buffer" in modo da garantire il corretto caricamento dei due file da programmare;
- con il FLIP caricare e quindi programmare sempre due file .HEX: quello con il programma applicativo utente generato dall'ambiente di sviluppo ed il *firmware di libreria* salvato nel file QTP12Rxx.HEX (dove xx corrisponde al numero di versione).

Nel paragrafo COME INIZIARE CON FIRMWARE DI LIBRERIA é disponibile un esempio completo di programmazione della FLASH EPROM, corredato delle fotografie sulle impostazioni cruciali del FLIP.

AMBIENTI DI SVILUPPO PER FIRMWARE DI LIBRERIA

Questa scheda ha la possibilità di usufruire di una ricca serie di strutture software che consentono di sviluppare al meglio il programma applicativo utente, più volte descritto nei paragrafi precedenti. In generale la **QTP 12/R84.LIB** può sfruttare tutte le risorse software per il microprocessore montato e tutti i pacchetti ideati per la famiglia 51, sia ad alto che a basso livello.

Gli ambienti di sviluppo software forniti dalla **grifo®** sono sempre accompagnati da esempi, librerie con console ridirezionabile, file di intestazione ed accessori che effettuano l'integrazione del *firmware di libreria* e lo rendono pronto all'uso per ogni comando e quindi ogni sezione della scheda. Tra questi ricordiamo:

BASCOM 8051: Cross compilatore a basso costo per files sorgenti scritti in BASIC, disponibile in ambiente WINDOWS con un comodo IDE che mette a disposizione un editor, il compilatore ed un simulatore molto potente per il debugger del sorgente. Comprende molti modelli di memoria, svariati tipi di dati ed istruzioni dedicate alle risorse hardware; è fornito su CD che include anche il relativo manuale d'uso.

µC/51: E' un comodo pacchetto software, a basso costo, che tramite un completo I.D.E. permette di utilizzare un editor, un compilatore ANSI C, un assembler, un linker e un remote debugger configurabile da utente a livello sorgente. Sono inclusi i sorgenti delle librerie fondamentali e del remote debugger, alcuni esempi di utilizzo e vari programmi di utility; è fornito su CD che include anche il relativo manuale d'uso.

LADDER WORK: E' un semplice sistema per creare programmi di automazione con la conosciuta e diffusa logica a contatti. Include un editor grafico che consente di posizionare e collegare i componenti hardware della scheda (input, output, contatori, A/D, ecc) come su uno schema elettrico e di definirne le proprietà, un efficiente compilatore che converte lo schema in codice eseguibile ed utility per il download di tale codice verso la scheda. Il tutto integrato in un comodo IDE per Windows. Viene fornito sotto forma di CD che comprende esempi e manuale d'uso e relativa chiave di abilitazione.

FIGURA 45: MODALITÀ SVILUPPO FIRMWARE DI LIBRERIA

COME INIZIARE CON FIRMWARE DI LIBRERIA

In questo paragrafo vengono illustrate le operazioni da effettuare per iniziare ad usare la **QTP 12/R84.LIB** in maniera rapida e lineare, senza dover affrontare e risolvere alcun problema iniziale. Al fine di rendere fruibili queste istruzioni a qualsiasi utente si seleziona come P.C. di sviluppo un normale Personal Computer provvisto di una linea seriale RS 232 libera ed un generico sistema operativo, superiore o uguale a Windows 98.

A) Collegamento seriale tra **QTP12/R84.LIB** e P.C.:

A1) Realizzare il collegamento seriale descritto nella figura 44 ovvero collegare i due segnali di comunicazione (TX RS232, RX RS232) e la massa di riferimento (GND) ad una porta di comunicazione COMx libera, del P.C. di sviluppo. Come si può notare tale cavo di collegamento é rovesciato e per praticità può essere ordinato alla **grifo®** specificando il codice CCR 9+9R.

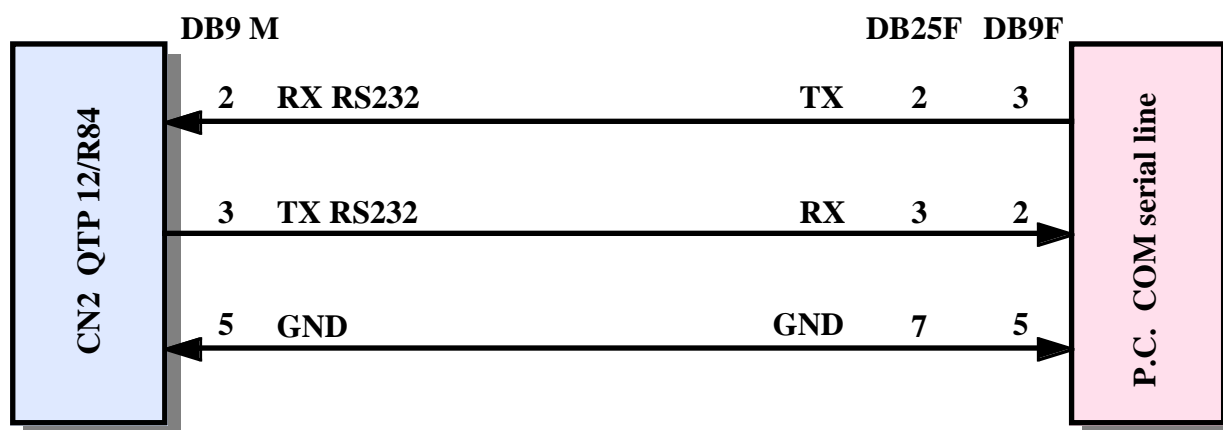


FIGURA 46: COLLEGAMENTO RS 232 CON P.C. E FIRMWARE DI LIBRERIA

A2) Avviare il programma di emulazione terminale **HYPERTERMINAL** sul P.C. di sviluppo ed impostare la comunicazione a:

Connetti	direttamente a COM x (quella usata al punto A1)
Bit per secondo	19200
Bit di Dati	8
Parità	Nessuna
Bit di Stop	1
Controllo di flusso	Nessuno

A3) Impostare la modalità **RUN**, ovvero non collegare il jumper **P1**.

A4) Fornire alimentazione su **CN1** e verificare che: il buzzer si disattivi, sul display compaia la stringa "Programma demo fw libreria QTP12/R84" con il cursore lampeggiante nell'angolo in alto a destra e che sul monitor del P.C. di sviluppo compaia la presentazione del programma dimostrativo. Ogni **QTP 12/R84.LIB**, in caso di primo acquisto, viene fornita con il rispettivo programma dimostrativo ed il *firmware di libreria* già programmati nella **FLASH** e configurata per farlo partire all'accensione: se non vedete comparire le schermate iniziali descritte, riverificate le connessioni seriali e controllate che il jumper **P1** sia correttamente posizionato.

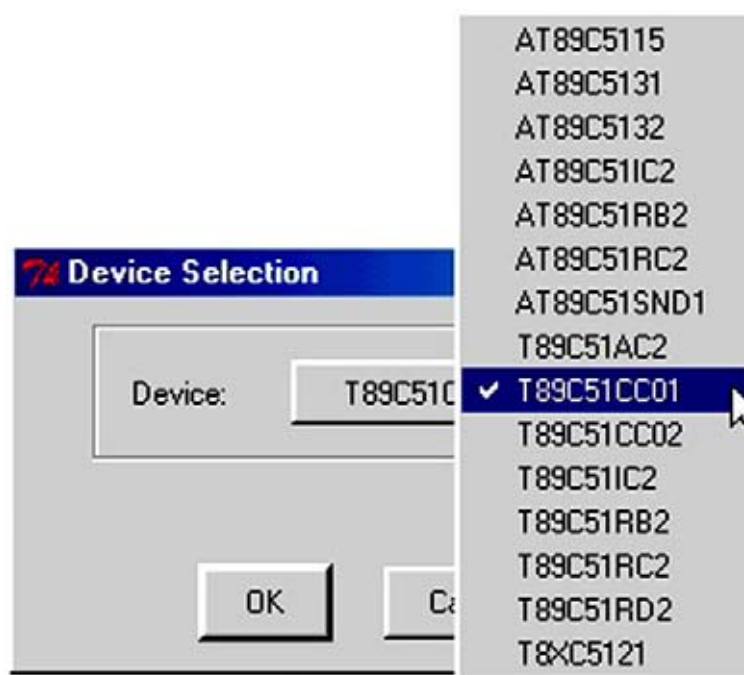
A5) Seguire le istruzioni del demo in modo da provare tutti i comandi del *firmware di libreria* e di verificarne gli effetti: l'utente deve interagire con il demo sia tramite la console seriale sul P.C. di sviluppo che tramite le risorse della stessa **QTP**.

A6) Terminata l'esecuzione del demo, togliere alimentazione alla **QTP**.

A7) Uscire dal programma **HYPERTERMINAL** sul P.C. di sviluppo.

B) Riprogrammazione della FLASH con programma demo:

- B1) Sul CD **grifo®** ricevuto sempre in caso di primo acquisto sono disponibili i file PRFWQ12R.HEX e QTP12Rxx.HEX (dove xx corrisponde al numero di versione), che contengono rispettivamente il codice eseguibile del demo ed il codice del *firmware di libreria*, già presenti in FLASH come descritto al punto A4. Tali file, una volta localizzati, devono essere copiati in una posizione comoda sul disco rigido del P.C. di sviluppo.
- B2) Sempre sul CD **grifo®** ricevuto, localizzare e quindi installare sul disco rigido del P.C. di sviluppo il programma di utility FLIP. Questo gestisce la programmazione ISP della FLASH EPROM sulla **QTP 12/R84.LIB** tramite il semplice collegamento seriale descritto al punto A1, come descritto nel precedente paragrafo PROGRAMMAZIONE FLASH EPROM.
- B3) Impostare modalità DEBUG, ovvero collegare il jumper P1.
- B4) Chiudere ogni programma che possa usare la linea seriale COMx del P.C. di sviluppo, come l'emulatore di terminale HYPERTERMINAL.
- B5) Fornire alimentazione e verificare che il buzzer si attivi e rimanga attivo durante l'esecuzione di tutti i successivi punti B6÷B17.
- B6) Lanciare software di gestione programmazione ISP: FLIP (versione >= 1.8.2) installato al punto B2.
- B7) Premere il primo pulsante in alto a sinistra, scegliere il microcontrollore da programmare nella finestra a scorrimento che appare e quindi premere OK. La selezione del dispositivo da programmare deve essere effettuata a seconda della **QTP** ordinata, ovvero T89C51AC2 in caso di **QTP 12/R84** e T89C51CC01 in caso di **QTP 12/R84.CAN**.

**FIGURA 47: FINESTRA SETTAGGIO FLIP (1 DI 4)**

- B8) Impostare le modalità di comunicazione per la programmazione ISP della **QTP** ovvero premere il secondo pulsante in alto da sinistra, scegliendo in sequenza: RS 232, la porta seriale del P.C. collegata al punto A1, 115200 Baud e quindi premendo il pulsante Connect. A questo punto il FLIP instaura la comunicazione con il boot loader del microcontrollore e compila una serie di dati nella sua finestra principale. Se la comunicazione non si instaura e dopo circa 20 secondi appare la finestra "Timeout Error", potete provare nell'ordine a: abbassare la velocità di comunicazione da 115200 a 19200 Baud; ripetere i punti da B3 fino a qui; verificare la corretta connessione seriale ripetendo i punti A1÷A5.



FIGURA 48: FINESTRA SETTAGGIO FLIP (2 DI 4)

- B9) Aprire la finestra "Buffer | Options" settare a NO la voce "Reset Buffer Before Loading" e selezionare la voce "Whole buffer" in modo da garantire il successivo corretto caricamento dei file da programmare.

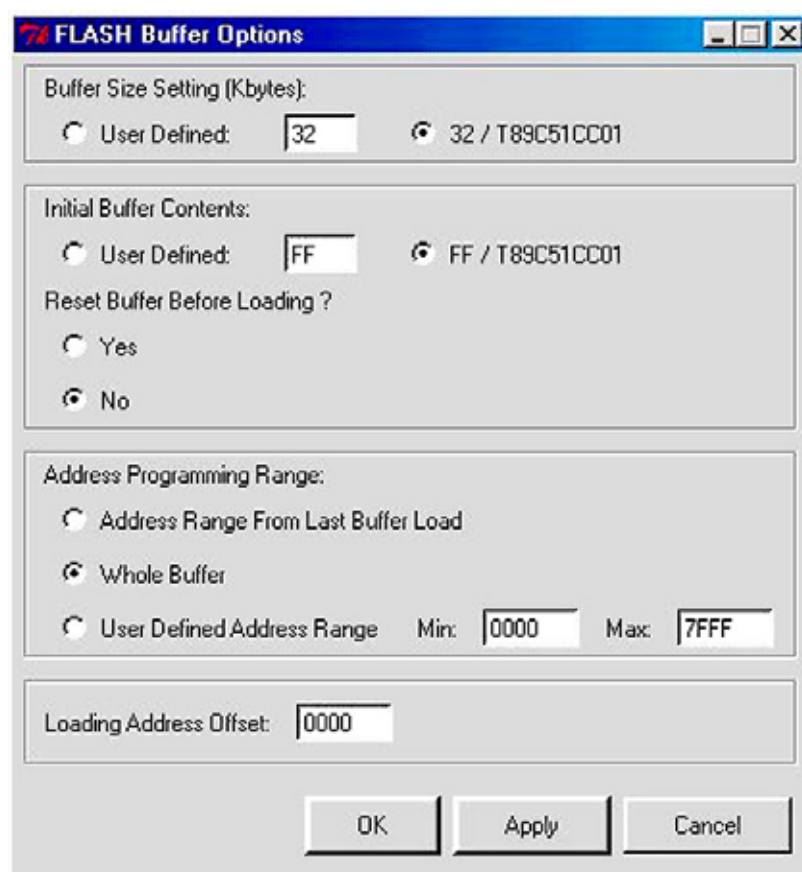


FIGURA 49: FINESTRA SETTAGGIO FLIP (3 DI 4)

- B10) Caricare i due file da salvare nella FLASH EPROM descritti al punto B1 (ovvero QTP12Rxx.HEX e PRFWQ12R.HEX) effettuando due volte le seguenti operazioni: premere il terzo pulsante da destra e selezionare il file tramite la finestra di dialogo che compare.
- B11) Spuntare tutte le caselle del riquadro "Operations Flow" come in figura 48, in modo che il FLIP esegua sequenzialmente le quattro operazioni di: cancellazione, verifica di cancellazione, programmazione e verifica di programmazione.
- B12) A questo punto assicurarsi che la finestra principale del FLIP si presenti come in figura 48, in particolare per le voci "Size:", "HEX File:", "Signature Bytes" e "BSB / EB / SBV".

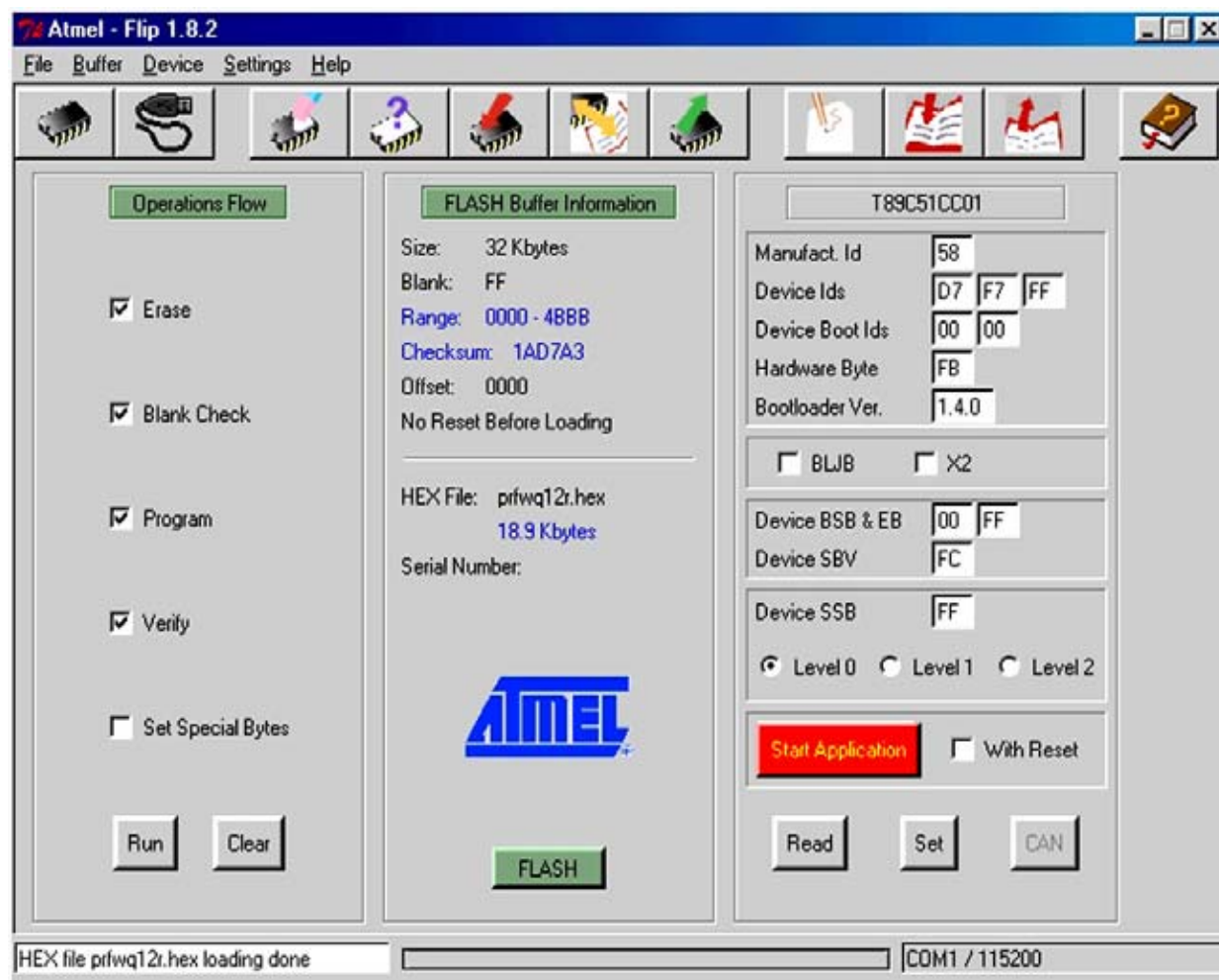


FIGURA 50: FINESTRA SETTAGGIO FLIP (4 DI 4)

- B13) Premere il pulsante "Run" nella finestra principale per avviare le operazioni ISP impostate.
- B14) Attendere la fine delle operazioni ISP. Nella barra di stato in basso si può vedere l'operazione in corso assieme ad una barra a scorrimento che indica il suo stato di avanzamento; le caselle di spunto diventano rosse durante l'esecuzione e poi verdi man mano che la rispettiva operazione viene completata. Si deve quindi aspettare che la casella "Verify" diventi verde.
- B15) Chiudere il FLIP.
- B16) Avviare l'emulatore di terminale configurato come nel punto A2. Il settaggio di HYPERTERMINAL può essere effettuato anche con un semplice doppio click sull'icona di un'apposito file di configurazione (file con estensione .HT) che può essere creato direttamente da HYPERTERMINAL, con l'opzione di salvataggio del menù "File".
- B17) Impostare la modalità RUN, ovvero non collegare il jumper P1.
- B18) Spegner e riaccendere la scheda; il programma demo, che usa il *firmware di libreria*, appena salvato in FLASH deve ripartire, come descritto nel punto A4.

C) Creazione del codice eseguibile del programma demo

- C1) Installare sul disco rigido del P.C. l'ambiente di sviluppo scelto per realizzare il programma applicativo. Come descritto nel paragrafo AMBIENTI DI SVILUPPO PER FIRMWARE DI LIBRERIA sono disponibili diversi ambienti in modo da soddisfare le richieste di ogni utente.
- C2) Sul CD grifo® oltre al file con il codice eseguibile del demo, descritto al punto B1, sono presenti anche il/i file sorgenti dello stesso. Questi hanno un'estensione che identifica

l'ambiente di sviluppo usato (ad esempio PRFWQ12R.BAS per il BASCOM 8051 oppure PRFWQ12R.C per il μ C/51) e sono opportunamente organizzati nelle tabelle degli esempi presenti sul CD, assieme agli eventuali file di definizione (8951CC01.DAT e FWQ12R.BAS per il BASCOM 8051, CANARY.H e FWQ12R.H per il μ C/51, ecc.). Una volta localizzati questi file devono essere salvati in una posizione comoda sul disco rigido del P.C. di sviluppo.

- C3) Ricompilare il sorgente usando l'ambiente di sviluppo scelto, in modo da ottenere il file PRFWQ12R.HEX identico a quello presente sul CD **grifo®** e già usato nei punti B.
- C4) Rieffettuare il salvataggio del file ottenuto nella FLASH della QTP, ripetendo i punti B3÷B18. In merito alle impostazioni del FLIP si ricorda che queste possono essere effettuate solo la prima volta, poi salvate in un apposito file di configurazione che una volta riaperto si preoccupa di riconfigurare automaticamente tutto il FLIP.

Se durante l'esecuzione dei passi sopra elencati si presenta un problema od un'anomalia si consiglia all'utente di rileggere e ripetere i passi con attenzione e qualora il malfunzionamento persista, di contattare direttamente la **grifo®**.

In caso di esecuzione corretta di tutte le fasi sopra descritte l'utente ha realizzato e salvato il suo primo programma applicativo coincidente con il demo della **QTP 12/R84.LIB**. A questo punto è possibile modificare il sorgente del/dei programmi demo in modo da soddisfare le richieste dell'applicazione da realizzare e provarla con i passi sopra elencati (da B3 a C3) in modo ciclico, fino a quando il programma applicativo realizzato è perfettamente funzionante. Raggiunto questo obiettivo si può eliminare il P.C. di sviluppo, ovvero:

D) *Preparazione definitiva dell'applicazione*

D1) Impostare modalità RUN (jumper P1 non connesso) e scollegare P.C. di sviluppo.

PROGRAMMI DEMO PER FIRMWARE DI LIBRERIA

In caso di primo acquisto sul dischetto o sul CD **grifo®** ricevuto sono disponibili numerosi programmi dimostrativi che consentono di provare e valutare immediatamente il prodotto ricevuto. Tali programmi sono forniti in formato eseguibile e sorgente e sono disponibili per gli ambienti di sviluppo proposti da **grifo®** ovvero compilatore C μ C/51, compilatore BASIC **BASCOM 8051**, logica a contatti **LADDER WORK**.

Come indicato nel paragrafo COME INIZIARE CON FIRMWARE DI LIBRERIA i programmi con il nome PRFWQ12R.* utilizzano tutti i comandi disponibili tramite una semplice iterazione con l'utente e consentono quindi di gestire il display in tutte le sue modalità di rappresentazione alfanumeriche e grafiche, la tastiera, la EEPROM, i messaggi, gli I/O digitali bufferati, i contatori hardware, alcuni sensori 1-WIRE, ecc. L'utente può esaminare i commenti di tali esempi e decidere autonomamente se provarli.

Tutti i programmi dimostrativi possono essere usati direttamente oppure modificati od utilizzati in parte, a seconda delle proprie esigenze, senza alcuna autorizzazione o costo aggiuntivo. In caso di particolari esigenze o combinazioni d'uso possono essere anche richiesti dei programmi specifici, oppure dei demo per diversi ambienti di sviluppo, previo accordo con la **grifo®**.

COMANDI

In questo capitolo sono descritti tutti i comandi disponibili nei firmware della **QTP 12/R84** assieme ai relativi parametri di ingresso ed uscita. I comandi sono divisi in sottogruppi concordemente con la loro funzione e per ogni codice, o sequenza di codici, viene riportata una doppia descrizione: quella mnemonica, tramite caratteri ASCII e quella numerica espressa in forma decimale ed esadecimale. I comandi seguono lo standard **ADDS View Point**, quindi tutte le sequenze iniziano con il carattere **ESC**, corrispondente al codice decimale **27 (1B Hex)**.

Una ricca serie di programmi dimostrativi (forniti in formato sorgente ed eseguibile) illustra le modalità pratiche di utilizzo dei comandi: si consiglia all'utente di integrare la presente documentazione con questi programmi che vengono forniti in caso di primo acquisto su CD o dischetto.

COMANDI PER IL POSIZIONAMENTO DEL CURSORE

Sono riportati di seguito, i comandi relativi alle varie modalità di posizionamento del cursore.

CURSORE A SINISTRA

Codice: **21**
Codice Hex: **15**
Mnemonico: **NACK**

Il cursore viene spostato di una posizione a sinistra senza alterare il contenuto del display. Se il cursore si trova nella posizione di Home, verrà posizionato nell'ultimo carattere in basso a destra del display.

CURSORE A DESTRA

Codice: **6**
Codice Hex: **6**
Mnemonico: **ACK**

Il cursore viene spostato di una posizione a destra. Se il cursore si trova nell'ultimo carattere, in basso a destra del display, verrà posizionato nella posizione di Home, in alto a sinistra.

CURSORE IN BASSO

Codice: **10**
Codice Hex: **A**
Mnemonico: **LF**

Il cursore viene posizionato nella riga successiva a quella in cui si trova, ma rimane nella stessa colonna. Se il cursore si trova nell'ultima riga del display, verrà posizionato nella prima riga del display.

CURSORE IN ALTO

Codice: 26
Codice Hex: 1A
Mnemonic: SUB

Il cursore viene posizionato nella riga precedente a quella in cui si trova, ma rimane nella stessa colonna. Se il cursore si trova nella prima riga del display, esso verrà posizionato nell'ultima riga.

CURSORE A INIZIO

Codice: 1
Codice Hex: 1
Mnemonic: SOH

Il cursore viene posto nella posizione di Home, corrispondente alla prima riga ed alla prima colonna del display, ovvero il carattere nell'angolo in alto a sinistra.

RITORNO A CAPO RIGA

Codice: 13
Codice Hex: D
Mnemonic: CR

Il cursore viene posizionato all'inizio della riga in cui si trova.

RITORNO A CAPO + NUOVA RIGA

Codice: 29
Codice Hex: 1D
Mnemonic: GS

Il cursore viene posizionato all'inizio della riga successiva a quella in cui si trovava. Se il cursore si trova nell'ultima riga del display, esso verrà posizionato all'inizio della prima riga, cioè nella posizione di Home.

POSIZIONAMENTO ASSOLUTO DEL CURSORE ALFANUMERICO

Codice: 27 89 r c
Codice Hex: 1B 59 r c
Mnemonic: ESC Y ASCII(r) ASCII(c)

Il cursore viene posizionato nel punto assoluto, indicato dai parametri **r** e **c**.

Questi parametri esprimono i valori di riga e colonna del display, riferiti alla posizione di Home, che ha coordinate 0, 0, a cui va aggiunto un offset di **32 (20 Hex)**. Tali coordinate sono riferite alla modalità alfanumerica del display e possono quindi rispettivamente variare nei range 32÷33 e 32÷51. Se ad esempio, si desidera posizionare il cursore sulla seconda linea, terza colonna (riga 1, colonna 2), sarà necessario inviare la sequenza:

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

Se i valori di riga e colonna non sono compatibili con il tipo di display installato, tale comando viene ignorato.

COMANDI PER LA CANCELLAZIONE DEI CARATTERI

Sono riportati di seguito, i comandi relativi alle modalità di cancellazione dei caratteri sul display.

SPAZIO INDIETRO

Codice: 8
Codice Hex: 8
Mnemonico: BS

Il cursore si sposta a sinistra di un carattere, cancellando il contenuto della cella raggiunta.

Se il cursore si trova nella posizione di Home, verrà cancellato il carattere che si trova nell'ultima cella in basso a destra del display.

CANCELLA PAGINA

Codice: 12
Codice Hex: C
Mnemonico: FF

Viene cancellato l'intero display ed il cursore viene posizionato in Home.

CANCELLA RIGA

Codice: 25
Codice Hex: 19
Mnemonico: EM

Viene cancellata l'intera linea in cui si trova il cursore, e questo viene posto all'inizio di tale riga.

CANCELLA FINO A FINE RIGA

Codice: 27 75
Codice Hex: 1B 4B
Mnemonico: ESC K

Vengono cancellati tutti i caratteri che si trovano nella riga in cui è posto il cursore, a partire dalla posizione del cursore stesso, fino al termine della riga. Il cursore rimane nella posizione in cui si trovava all'arrivo del codice di Cancella fino a fine riga.

Se ad esempio, il cursore si trova all'inizio di una riga del display, verrà cancellata l'intera linea.

CANCELLA FINO A FINE PAGINA

Codice: 27 107
Codice Hex: 1B 6B
Mnemonico: ESC k

Vengono cancellati tutti i caratteri dal punto in cui si trova il cursore, fino al termine del display. Il cursore rimane nella posizione in cui si trovava all'arrivo del codice di Cancella fino a fine pagina.

Se ad esempio, il cursore si trova nella posizione di Home, verrà cancellato l'intero display.

COMANDI PER LA GESTIONE DELLA EEPROM

Sono riportati di seguito alcuni dei comandi per la gestione dei dati memorizzati nella EEPROM di bordo del terminale **QTP 12/R84**; i rimanenti comandi che utilizzano indirettamente questa memoria sono invece descritti nei paragrafi successivi.

RICHIESTA DISPONIBILITA' AD USARE EEPROM

Codice: 27 51
Codice Hex: 1B 33
Mnemonico: ESC 3

Tramite questo comando é possibile richiedere al firmware se é possibile accedere alla EEPROM di bordo; esso deve essere eseguito prima di inviare qualsiasi comando che successivamente effettui una scrittura od una lettura su questo tipo di memoria.

Dopo aver inviato tale comando, il firmware della **QTP 12/R84** restituisce un codice che può essere:

6	(06 Hex)	(ACK)	->	QTP 12/R84 pronta
21	(15 Hex)	(NACK)	->	QTP 12/R84 non pronta

Se il firmware restituisce il codice NACK, non é ancora possibile memorizzare un nuovo dato in EEPROM oppure prelevarne uno già scritto.

SCRITTURA DEL BYTE DI PRESENZA

Codice: 27 33 78 byte
Codice Hex: 1B 21 4E byte
Mnemonico: ESC ! N ASCII(byte)

Il byte di presenza della scheda viene settato con il valore indicato nel parametro **byte** e può essere compreso nel range **0÷255**.

Questo byte é un'allocazione riservata nella EEPROM di bordo che, una volta settata con il valore desiderato, permette, ad esempio, di verificare che la **QTP 12/R84** funzioni correttamente, oppure che non ci siano conflitti di comunicazione in caso di *firmware seriale*.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando viene ignorato.

LETTURA DEL BYTE DI PRESENZA

Codice: 27 33 110
Codice Hex: 1B 21 6E
Mnemonico: ESC ! n

Il firmware restituisce il valore del proprio byte di presenza.

Tale comando può essere utile, ad esempio, nel caso si debba verificare la presenza, o il corretto funzionamento della scheda stessa e del suo firmware.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando non viene eseguito e viene restituito il codice **21 (15 Hex) = NAK**.

SCRITTURA BYTE SU EEPROM

Codice: 27 164 addl addh byte

Codice Hex: 1B A4 addl addh byte

Mnemonico: ESC ASCII(164) ASCII(addl) ASCII(addh) ASCII(byte)

Il valore passato nel parametro **byte**, compreso nel range **0÷255 (0÷FF Hex)**, viene scritto nella locazione della EEPROM utente, identificata dall'indirizzo **addh addl**.

L'EEPROM utente é un'area riservata nella EEPROM di bordo che può essere gestita direttamente a byte senza far uso degli altri comandi per messaggi, byte di presenza, ecc. Gli usi tipici di questa area sono il salvataggio di configurazioni, impostazioni, settaggi, ecc. che devono essere mantenuti anche in assenza di alimentazione. L'indirizzo che identifica la locazione usata é a 16 bit e di questi **addh** ed **addl** costituiscono rispettivamente la parte più e meno significativa. La EEPROM utente con i firmware della **QTP 12/R84** ha una dimensione di 40 bytes quindi il valore del parametro **addl** deve essere compreso nel range **0÷39 (0÷27H)** mentre il valore di **addh** deve essere sempre a 0. Questa scelta é stata effettuata per compatibilità con espansioni future ed altri terminali.

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dati non validi.

Se, per esempio, si vuole scrivere il valore 100 all'indirizzo 35 della EEPROM utente, sarà necessario inviare la sequenza:

27 164 35 0 100 oppure **1B A4 23 00 64 Hex** oppure **ESC ASCII(164) # NUL d**

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando viene ignorato.

LETTURA BYTE DA EEPROM

Codice: 27 165 addl addh

Codice Hex: 1B A5 addl addh

Mnemonico: ESC ASCII(165) ASCII(addl) ASCII(addh)

Il valore salvato nella locazione della EEPROM utente, identificata dall'indirizzo **addh addl**, viene letto e restituito. Come indicato nel comando precedente i valori del primo parametro deve essere compreso nel range **0÷39 (0÷27H)** ed il valore del secondo deve essere sempre 0; il dato restituito coincide con un singolo carattere compreso nel range **0÷255 (0÷FF Hex)**.

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dati non validi.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando non viene eseguito e viene restituito il codice **21 (15 Hex) = NAK**.

COMANDI PER FUNZIONI VARIE

Sono di seguito riportati i comandi per la gestione di varie funzioni dei firmware **QTP 12/R84** che non rientrano in nessuno degli altri gruppi e che sono quindi stati riuniti in questo paragrafo.

LETTURA DEL NUMERO DI VERSIONE

Codice: 27 86
Codice Hex: 1B 56
Mnemonic: ESC V

Viene restituita, una sequenza di 3 caratteri contenente il numero di versione del firmware di gestione, residente a bordo della **QTP 12/R84**.

Ad esempio con un firmware versione 1.3 i seguenti caratteri saranno restituiti:

49 46 51 oppure 31 2E 33 Hex oppure 1.3

LETTURA DEL CODICE SCHEDA

Codice: 27 160
Codice Hex: 1B A0
Mnemonic: ESC ASCII(160)

Viene restituita, un carattere contenente il codice scheda che nel caso della **QTP 12/R84** coincide con il valore **2 (02 Hex)**. Tale comando é utile soprattutto in caso di *firmware seriale* quando dispositivi diversi sono collegati sulla stessa rete seriale e l'unità master deve riconoscere tali dispositivi. Naturalmente il codice scheda identifica in modo univoco il prodotto associato.

SETTAGGIO LUMINOSITÀ DISPLAY FLUORESCENTE

Codice: 27 108 lum
Codice Hex: 1B 6C lum
Mnemonic: ESC l ASCII(lum)

Setta la luminosità sul display fluorescente ad uno dei quattro possibili valori definito dal valore del parametro **lum**, con la seguente corrispondenza:

0	(00 Hex)	->	luminosità al 100%
1	(01 Hex)	->	luminosità al 75%
2	(02 Hex)	->	luminosità al 50%
3	(03 Hex)	->	luminosità al 25%

Se il parametro con la luminosità non é valido, il comando viene ignorato.

N.B. Tale comando é utilizzabile solo quando si utilizzano i modelli **QTP 12/R84-F2** e **QTP 12/R84-GF2**, ovvero quando é presente un display fluorescente. In caso di **QTP 12/R84-C2** con display LCD, il comando non deve essere fornito infatti provoca la rappresentazione di un carattere spurio e lo sfasamento dei successivi.

SETTAGGIO MODALITÀ OPERATIVA

Codice: 27 65 *modo*
Codice Hex: 1B 41 *modo*
Mnemonico: ESC A ASCII(*modo*)

Definisce la modalità di gestione dei caratteri speciali bassi (quelli con codice inferiore a 32 = 20H) e quindi dei comandi ad un solo carattere. La modalità scelta é definita dal valore del byte **modo**, con la seguente corrispondenza:

0 (00 Hex)	-> Modalità comandi
255 (FF Hex)	-> Modalità rappresentazione

Se il parametro con il **modo** non é valido, il comando viene ignorato. Per ulteriori informazioni vedere paragrafo RAPPRESENTAZIONE DI CARATTERI SUL DISPLAY.

RESET DELLA COMUNICAZIONE

Codice: 27 163
Codice Hex: 1B A3
Mnemonico: ESC ASCII(163)

Ristabilisce uno stato azzerato sulla comunicazione, senza azzerare gli altri processi in corso. Naturalmente con comunicazione s'intende lo scambio di dati tra il firmware usato ed programma utente ed é quindi indipendente dal mezzo usato. In particolare il comando effettua le seguenti operazioni:

- svuota buffer di ricezione;
- elimina eventuali caratteri di risposte ancora da restituire;
- elimina eventuali tasti premuti ancora da restituire;
- termina la gestione dei comandi in corso che dirottano i caratteri forniti (vedi scrittura messaggi, modalità grafica, scrittura bytes su linea 1-WIRE, ecc.).

RESET GENERALE

Codice: 27 162
Codice Hex: 1B A2
Mnemonico: ESC ASCII(162)

Effettua un azzeramento generale del firmware in esecuzione reimpostando una condizione iniziale simile a quella impostata a seguito di un'accensione. In dettaglio il comando effettua le seguenti operazioni:

- azzerla la comunicazione come descritto nel comando precedente;
- disattiva tutte le uscite digitali e le eventuali temporizzazioni associate;
- azzerare tutti i contatori;
- disattiva tutte le funzioni alternative di tutti i segnali;
- cancella il display e disabilita l'eventuale shift di messaggi;
- disattiva LED di stato e buzzer e le eventuali intermittenze;
- ricarica i settaggi salvati in EEPROM ovvero keyclick, codici tasti, caratteri utente, nome di identificazione, protocollo di comunicazione, ecc.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo.

Il tempo di esecuzione di questo comando é dell'ordine di alcune decine di msec: in caso di *firmware seriale* .SER, se dopo la sua trasmissione devono essere mandati numerosi altri comandi é preferibile inserire un ritardo che eviti il riempimento del buffer di ricezione.

ATTIVAZIONE TEMPORIZZATA DEL BUZZER

Codice: 7
Codice Hex: 7
Mnemonic: BEL

Viene attivato il buzzer per un tempo di circa 1 decimo di secondo. Se il buzzer era già attivo allora viene disattivato per lo stesso intervallo di tempo, in modo da rendere sempre riconoscibile l'effetto del comando.

ATTIVAZIONE LED E BUZZER

Codice: 27 50 disp attr
Codice Hex: 1B 32 disp attr
Mnemonic: ESC 2 ASCII(disp) ASCII(attr)

Viene attivato il dispositivo indicato in **disp**, con l'attributo specificato in **attr**. In particolare i dispositivi hanno la seguente corrispondenza:

0	(00 Hex)	->	LED di stato
255	(FF Hex)	->	buzzer

mentre gli attributi disponibili sono i seguenti:

0	(00 Hex)	->	dispositivo disattivato
255	(FF Hex)	->	dispositivo attivato
85	(55 Hex)	->	dispositivo intermittente

Se il parametro con il dispositivo, o quello con l'attributo, non sono validi, il comando viene ignorato. La funzionalità di intermittenza é totalmente autonoma e non richiede nessun intervento da parte dell'utilizzatore. Se ad esempio, si vuole attivare il LED con l'attributo di lampeggio, sarà necessario inviare la sequenza:

27 50 0 85 oppure 1B 32 00 55 Hex oppure ESC 2 NUL U

COMANDI PER LA GESTIONE DEI MESSAGGI

Sono riportati di seguito, i comandi per la gestione dei messaggi, disponibili nei firmware della **QTP 12/R84**. I messaggi coincidono con delle sequenze di 20 caratteri che possono essere salvati nella EEPROM di bordo e successivamente prelevati o rappresentati sul display semplicemente fornendo al terminale il numero di identificazione del messaggio stesso. La funzione principale dei messaggi é quella di poter rappresentare sul display informazioni costanti (es. allarmi, stati dell'impianto, ecc.) senza dover fornire ogni volta i numerosi caratteri di tali informazioni ma solo i pochi caratteri del comando. Il firmware della **QTP 12/R84**, inoltre, gestisce la rappresentazione di tali messaggi anche in modalità a scorrimento; in questo modo é possibile visualizzare in un'unica riga del display delle informazioni che occupano uno spazio maggiore di quello normalmente rappresentabile.

In caso di firmware seriale **.SER**, un comodo programma per PC denominato **QTP EDIT** consente di editare i messaggi, salvandoli e prelevandoli dai dischi del PC, e di trasmettere e ricevere i messaggi direttamente dalla **QTP** collegata in seriale allo stesso PC.

La **QTP 12/R84** installa una EEPROM che ha una capacità di 2048 bytes che riesce a memorizzare fino ad un massimo di 95 messaggi numerati da 0 a 94.

LETTURA NUMERO DELL'ULTIMO MESSAGGIO MEMORIZZABILE

Codice: 27 110

Codice Hex: 1B 6E

Mnemonico: ESC n

Viene restituito, il numero dell'ultimo messaggio memorizzabile; questo coincide sempre con il valore **94 (5E Hex)** precedentemente descritto. Tale comando é particolarmente importante sulle altre **QTP** in cui le dimensioni della EEPROM sono variabili mentre sulla **QTP 12/R84** é stato previsto in modo da mantenere appunto la compatibilità e l'intercambiabilità con gli altri pannelli operatore della **grifo®**.

MEMORIZZAZIONE DI UN MESSAGGIO

Codice: 27 33 67 n.mess. car. 0 ÷ car. 19

Codice Hex: 1B 21 43 n.mess. car. 0 ÷ car. 19

Mnemonico: ESC ! C ASCII(n.mess.) ASCII(car. 0) ÷ ASCII(car. 19)

Il messaggio, di 20 caratteri, identificato da **n.mess.** viene memorizzato nella EEPROM di bordo. I codici dei 20 caratteri che compongono il messaggio devono essere visualizzabili sul display, quindi compresi nel range **0÷255 (0÷FF Hex)**. Il numero del messaggio deve essere compreso nel range **0÷94**, ovvero selezionare uno dei possibili messaggi disponibili.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando viene ignorato.

LETTURA DI UN MESSAGGIO

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

Il messaggio di 20 caratteri, corrispondente a **n.mess.**, viene letto dall'EEPROM e restituito, partendo dal primo carattere della stringa in questione.

Il numero del messaggio deve essere compreso nel range **0÷94**, ovvero selezionare uno dei possibili messaggi disponibili. Se tale numero non é valido il comando viene ignorato.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se cosí non é, il comando non viene eseguito e viene restituito il codice **21 (15 Hex) = NAK**.

VISUALIZZAZIONE DI MESSAGGI

Codice: 27 33 68 *n. mess. n*
Codice Hex: 1B 21 44 *n. mess n*
Mnemonic: ESC ! D ASCII(*n. mess.*) ASCII(*n*)

Vengono visualizzati sul display **n** messaggi di 20 caratteri, a partire dalla posizione attuale del cursore.

Il primo degli **n** messaggi, é quello identificato da **n.mess.**, mentre gli altri sono quelli immediatamente successivi in EEPROM.

Il numero **n.mess.** ed il numero dei messaggi successivi definito da **n**, devono essere compresi nel range **0÷94**, ovvero selezionare i possibili messaggi disponibili. Se tali numeri non sono validi il comando viene ignorato.

Invece la quantità **n** di messaggi da visualizzare, dipende dal modello di display installato e, visto che in modalità alfanumerica per tutti i tre diversi display il numero massimo di caratteri rappresentabili é 40, si ricava che il numero massimo di messaggi é 2. In altre parole il parametro **n** deve essere compreso nel range **1÷2** e se il suo valore non é valido, il comando viene ignorato.

Una volta eseguito il comando il cursore viene posizionato nella cella successiva all'ultimo messaggio visualizzato; se l'ultimo carattere del messaggio in questione, occupa l'ultima posizione del display, il cursore verrà posizionato nella posizione di Home.

Se, per esempio, si vogliono visualizzare i messaggi 10 e 11, sarà necessario inviare al firmware la sequenza:

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

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se cosí non é, il comando viene ritardato fino al completamento dell'operazione in corso.

VISUALIZZAZIONE DI MESSAGGI A SCORRIMENTO

Codice: 27 33 83 *n. mess. n.car.*

Codice Hex: 1B 21 53 *n. mess n.car.*

Mnemonico: ESC ! S ASCII(*n. mess.*) ASCII(*n.car*)

Visualizza, sulla prima riga del display, una stringa a scorrimento di **n.car.** caratteri; la stringa composta da questi caratteri viene shift-ata da destra verso sinistra, in modo da visualizzare, in un'unica riga del display (la prima), delle informazioni che occupano uno spazio maggiore di quello normalmente rappresentabile.

La stringa in questione, di lunghezza **n.car.**, inizia dal primo carattere del messaggio memorizzato in EEPROM, con codice corrispondente ad **n.mess.** e prosegue con tutti i caratteri seguenti (memorizzati nei successivi messaggi della EEPROM).

Il parametro **n.mess.** deve essere compreso nel range **0÷94**, ovvero selezionare uno dei possibili messaggi disponibili. Se tale numero non è valido il comando viene ignorato.

Il dato **n.car.**, invece, deve essere compreso nei seguenti valori:

0	Interrompe lo scorrimento in corso (il valore di n.mess. non è significativo)
20÷200	Attiva lo scorrimento dei caratteri specificati

Se **n.car.** contiene un valore al di fuori di tali specifiche, il comando viene ignorato; questo avviene anche se il numero di caratteri in questione, si estende oltre l'ultimo messaggio memorizzato nella EEPROM.

La rappresentazione del messaggio a scorrimento avviene nella prima riga del display, senza variare la posizione del cursore ed i relativi attributi.

Se, per esempio, si vuole visualizzare, a scorrimento, la stringa di 35 caratteri, composta dal messaggio 10 (20 caratteri) e dai primi 15 caratteri del messaggio 11, sarà necessario inviare al firmware la sequenza:

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

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo è meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non è, il comando viene ritardato fino al completamento dell'operazione in corso.

L'operazione di scorrimento di una stringa, comporta un aggiornamento periodico del messaggio sul display: questa operazione incrementa il tempo di esecuzione del firmware e quindi un rallentamento nell'esecuzione del programma applicativo utente in caso di *firmware di libreria* **.LIB**, o nell'interpretazione dei dati ricevuti in caso di *firmware seriale* **.SER**. Con quest'ultimo firmware se deve essere inviata una quantità notevole di informazioni alla **QTP 12/R84**, ed è in corso lo scorrimento di un messaggio, è consigliabile attendere alcuni **msec**, fra la trasmissione di un blocco di 20÷30 byte ed il successivo, per avere la certezza che il terminale interpreti correttamente i dati ricevuti, senza riempire il buffer di ricezione.

COMANDI PER GESTIONE DELLA TASTIERA

Sono riportati di seguito i comandi per la gestione dei tasti presenti sulla **QTP 12/R84**. Per informazioni sulla gestione e sui codici che il firmware restituisce in corrispondenza della pressione dei vari tasti, fare riferimento al paragrafo **ACQUISIZIONE DELLA TASTIERA**.

RICONFIGURAZIONE DI UN TASTO

Codice: 27 55 *n.tasto codice*

Codice Hex: 1B 37 *n.tasto codice*

Mnemonic: ESC 7 ASCII(*n.tasto*) ASCII(*codice*)

Il codice del tasto corrispondente a **n.tasto** viene riconfigurato, ed ogni volta che questo verrà premuto, il firmware restituirà il nuovo **codice** specificato.

Il parametro **n.tasto** può essere stabilito facendo riferimento alla figura 37 e deve essere compreso nel range **0÷11 (0÷B Hex)**, se questo non avviene il comando viene ignorato.

Se **codice** è compreso nel range **0÷254 (00÷FE Hex)**, a seguito della pressione del tasto in questione, il firmware restituisce il valore specificato; mentre se tale parametro viene impostato a **255 (FF Hex)**, il tasto sarà disattivato e la sua pressione non produrrà alcun risultato.

La figura 38 riporta i codici dei tasti attribuiti di default ed il paragrafo **DATI IN EEPROM** indica come ripristinare questi codici in caso di modifiche indesiderate.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo è meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non è, il comando viene ignorato.

ATTIVAZIONE KEYCLICK SENZA MEMORIZZAZIONE

Codice: 27 53

Codice Hex: 1B 35

Mnemonic: ESC 5

Viene attivata la funzione di **keyclick**, ovvero l'attivazione temporizzata del buzzer quando viene premuto un tasto (qualora il buzzer sia già attivo viene disattivato per lo stesso intervallo di tempo, in modo da rendere sempre riconoscibile la pressione del tasto). Con questo comando tale configurazione non viene memorizzata nella EEPROM di bordo, quindi a seguito di una riaccensione od inizializzazione, si ritorna alla condizione precedentemente salvata su questa memoria.

DISATTIVAZIONE KEYCLICK SENZA MEMORIZZAZIONE

Codice: 27 54

Codice Hex: 1B 36

Mnemonic: ESC 6

Viene disattivata la funzione di **keyclick**, non si ha quindi un feedback sonoro quando viene premuto un tasto. Con questo comando la configurazione non viene memorizzata nella EEPROM di bordo, quindi a seguito di una riaccensione od inizializzazione, si ritorna alla condizione precedentemente salvata su questa memoria.

ATTIVAZIONE KEYCLICK CON MEMORIZZAZIONE

Codice: 27 33 53

Codice Hex: 1B 21 35

Mnemonic: ESC ! 5

Viene attivata la funzione di **keyclick**, cioè l'attivazione temporizzata del buzzer quando viene premuto un tasto (qualora il buzzer sia già attivo viene disattivato per lo stesso intervallo di tempo, in modo da rendere sempre riconoscibile la pressione del tasto). Tale settaggio viene memorizzato nella EEPROM di bordo, e quindi mantenuto anche in assenza della tensione di alimentazione.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando viene ignorato.

DISATTIVAZIONE KEYCLICK CON MEMORIZZAZIONE

Codice: 27 33 54

Codice Hex: 1B 21 36

Mnemonic: ESC ! 6

Viene disattivata la funzione di **keyclick**, eliminando quindi il feedback sonoro quando viene premuto un tasto; tale parametro viene memorizzato nella EEPROM di bordo, e quindi mantenuto anche in assenza della tensione di alimentazione.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando viene ignorato.

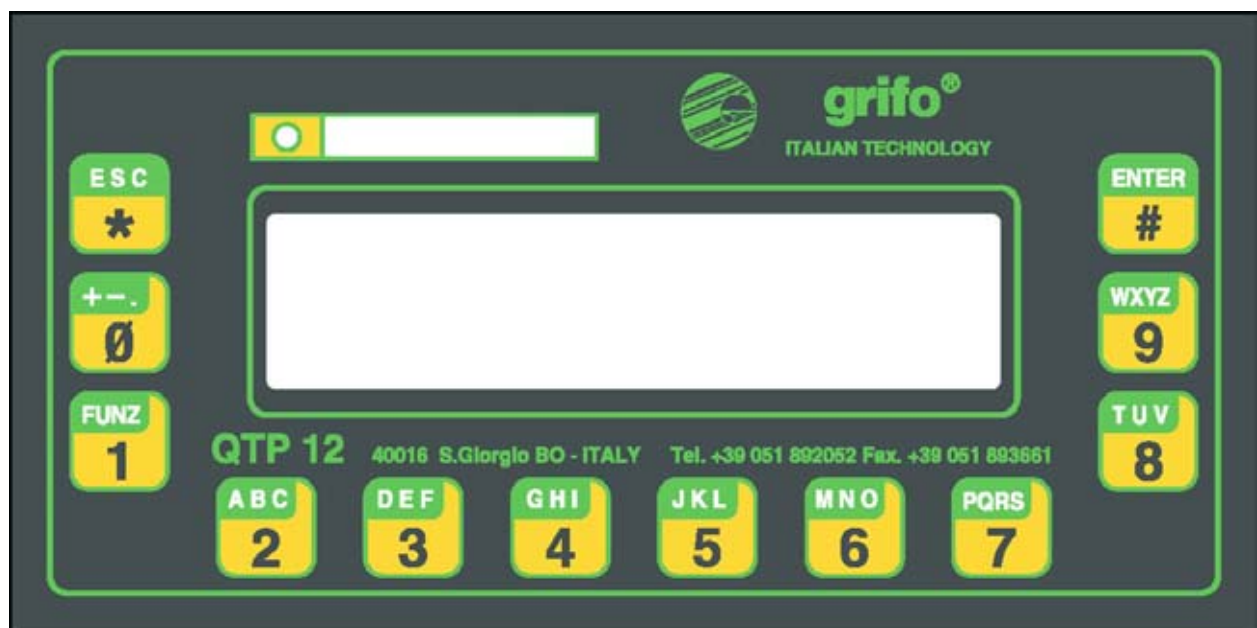


FIGURA 51: FOTO DEL FRONTALE CON TASTIERA

COMANDI PER CARATTERI UTENTE

Il firmware della **QTP 12/R84** fornisce all'utente la possibilità di definire e rappresentare fino ad 8 caratteri utente; tali caratteri possono essere usati per rappresentare sul display caratteri speciali, caratteri pseudografici, simboli speciali, ecc. che non sono già disponibili nello stesso display (vedere tabelle nell'APPENDICE B). I caratteri utente possono essere definiti e salvati con un modello corrispondente ad una matrice di 5 x 8 punti, così organizzati:

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

FIGURA 52: MODELLO DEI CARATTERI UTENTE

Per la loro rappresentazione é invece sufficiente fornire il relativo codice (da 0 a 7 equivalente da 8 a 15) eventualmente usando prima il comando di **SETTAGGIO MODALITA' OPERATIVA** per selezionare appunto la modalità rappresentazione.

Quando i caratteri utente sono salvati i loro modelli di punti vengono salvati sulla EEPROM di bordo e successivamente ricaricati sul display, in corrispondenza di ogni accensione od inizializzazione, del terminale.

N.B. Si ricorda che sul modello **QTP 12/R84-F2** il modello dei caratteri utente é una matrice di 5x7 punti (Pat 0÷Pat 6) e che lo stato del punto Pat 7.4 definisce lo stato di tutti i cinque punti Pat 7.4÷Pat 7.0, ovvero dell'attributo di sottolineatura del carattere.

DEFINIZIONE DI UN CARATTERE UTENTE

Codice: 27 66 ncar Pat 0 ÷ Pat 7

Codice Hex: 1B 42 ncar Pat 0 ÷ Pat 7

Mnemonico: ESC B ASCII(ncar) ASCII(Pat 0) ÷ ASCII(Pat 7)

Dopo i due codici iniziali, bisogna fornire altri 9 bytes che hanno il seguente significato:

ncar	(0÷7)	(00÷7 Hex)	->	Numero del carattere utente da definire.
Pat 0	(0÷31)	(00÷1F Hex)	->	Primo byte del modello equivalente alla prima riga alta del carattere.
:	:		->	:
Pat 7	(0÷31)	(00÷1F Hex)	->	Settimo byte del modello equivalente all'ultima riga bassa del carattere.

Con tale comando viene impostato sul display il modello del carattere definibile da utente **ncar** con i valori passati negli otto byte **Pat 0 ÷ Pat 7**, come descritto nella figura 50; tale modello viene solo definito ma non salvato, quindi in caso di spegnimento e riaccensione od inizializzazione, il carattere utente **ncar** non manterrà il modello fornito.

Se, per esempio, si vuole definire il carattere utente 5 con un rettangolo vuoto delle massime dimensioni, sarà necessario inviare al firmware la sequenza:

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

DEFINIZIONE E SALVATAGGIO DI UN CARATTERE UTENTE

Codice: 27 33 66 ncar Pat 0 ... Pat 7

Codice Hex: 1B 21 42 ncar Pat 0 ... Pat 7

Mnemonico: ESC ! B ASCII(ncar) ASCII(Pat 0) ... ASCII(Pat 7)

Dopo i tre codici iniziali, bisogna fornire altri 9 bytes che hanno il seguente significato:

ncar	(0÷7)	(00÷7 Hex)	->	Numero del carattere utente da definire.
Pat 0	(0÷31)	(00÷1F Hex)	->	Primo byte del modello equivalente alla prima riga alta del carattere.
:	:		->	:
Pat 7	(0÷31)	(00÷1F Hex)	->	Settimo byte del modello equivalente all'ultima riga bassa del carattere.

Con tale comando viene impostato sul display il modello del carattere definibile da utente **ncar** con i valori passati negli otto byte **Pat 0 ÷ Pat 7**, come descritto nella figura 50; tale modello viene inoltre salvato in EEPROM, quindi in caso di spegnimento e riaccensione od inizializzazione, il carattere utente **ncar** manterrà il modello fornito.

N.B. Tale comando comporta un accesso alla EEPROM di bordo, quindi prima di eseguirlo é meglio assicurarsi che la scheda sia pronta per una nuova operazione su tale dispositivo; se così non é, il comando viene ignorato.

Il tempo di esecuzione di questo comando é dell'ordine di 80 msec. In caso di *firmware seriale .SER*, se dopo la sua trasmissione devono essere mandati numerosi altri comandi é preferibile inserire un ritardo che eviti il riempimento del buffer di ricezione.

COMANDI PER GESTIONE DEGLI ATTRIBUTI DEL CURSORE

Sono riportati di seguito, i comandi relativi ai vari attributi del cursore. Si ricorda che il cursore é disponibile solo nella modalit  alfanumerica, mentre in quella grafica non   mai rappresentato, anche se gestito. E' comunque possibile definire la posizione e lo stile del cursore anche in grafica, grazie all'uso alternato dei comandi grafici ed alfanumerici.

DISATTIVAZIONE DEL CURSORE

Codice: 27 80

Codice Hex: 1B 50

Mnemonic: ESC P

Il cursore viene disattivato e non   pi  visibile.

ATTIVAZIONE DEL CURSORE FISSO

Codice: 27 79

Codice Hex: 1B 4F

Mnemonic: ESC O

Il cursore viene attivato, quindi reso visibile, e rappresentato sotto forma di linea non lampeggiante posizionata al di sotto del carattere.

N.B. Tale comando   utilizzabile solo quando si utilizzano i modelli **QTP 12/R84-C2** e **QTP 12/R84-F2**, ovvero quando   presente un display alfanumerico. In caso di **QTP 12/R84-GF2** con display grafico, il comando non ha alcun effetto.

ATTIVAZIONE DEL CURSORE "BLOCCO" LAMPEGGIANTE

Codice: 27 81

Codice Hex: 1B 51

Mnemonic: ESC Q

Il cursore viene attivato, quindi reso visibile, e rappresentato sotto forma di rettangolo lampeggiante, visualizzato alternativamente con il carattere sovrapposto ad esso.

COMANDI PER GESTIONE I/O DIGITALI BUFFERATI

Sono riportati di seguito i comandi relativi alla gestione dei segnali di I/O digitale bufferato disponibili nei firmware della **QTP 12/R84**. Come descritto in tutti i successivi paragrafi per individuare i segnali di I/O si usano gli stessi nomi usati nella descrizione dei relativi connettori (vedi figure 24, 26) e quando necessario, un anumerazione sequenziale che parte da 1.

SCRITTURA DI TUTTE LE USCITE DIGITALI

Codice: 27 166 out
Codice Hex: 1B A6 out
Mnemonico: ESC ASCII(166) ASCII(out)

Vengono settate tutte le uscite digitali bufferate a relé con il valore **out**, secondo la seguente corrispondenza:

(MSB) 0 0 0 0 OUT B2 OUT B1 OUT A2 OUT A1 (LSB)

Dove **OUT Nn** indica lo stato logico, **0** (output disattivo) o **1** (output attivo), che devono assumere le relative uscite a relé della scheda, su CN4.

Se sulle linee di output sono in corso delle temporizzazioni, queste vengono interrotte.

Qualora la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Se, per esempio, si vogliono attivare solo le uscite OUT A2 ed OUT B2 sarà necessario inviare la sequenza:

27 166 10 oppure 1B A6 0A Hex oppure ESC ASCII(166) LF

LETTURA DI TUTTI GLI INGRESSI DIGITALI

Codice: 27 167
Codice Hex: 1B A7
Mnemonico: ESC ASCII(167)

Viene acquisito e quindi restituito lo stato di tutti gli ingressi digitali optoisolati IN0÷7 disponibili su CN5. Tale stato viene restituito sotto forma di un valore con la seguente corrispondenza:

(MSB) IN 8 IN 7 IN 6 IN 5 IN 4 IN 3 IN 2 IN 1 (LSB)

Dove i bit **IN n**, indicano gli stati logici **0** (ON = ingresso chiuso) o **1** (OFF = ingresso aperto), in cui si trovano le relative linee di input.

Qualora la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Se, per esempio, su CN5 sono chiusi solo gli ingressi IN 6 ed IN 5 sarà restituito il seguente dato:

48 oppure 30 Hex oppure 0

ATTIVAZIONE SINGOLA USCITA DIGITALE

Codice: 27 168 bit
Codice Hex: 1B A8 bit
Mnemonic: ESC ASCII(168) ASCII(bit)

Viene posta allo stato logico **1** (uscita attiva), la linea di uscita digitale a relé identificata da **bit** che ha la seguente corrispondenza con i segnali su CN4:

1 -> OUT A1 **2** -> OUT A2 **3** -> OUT B1 **4** -> OUT B2

Se sulla linea di output in questione sono in corso delle temporizzazioni, queste vengono interrotte. Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vuole attivare l'uscita OUT A2 senza alterare le rimanenti uscite sarà necessario inviare la sequenza:

27 168 2 oppure **1B A8 02 Hex** oppure **ESC ASCII(168) STX**

ATTIVAZIONE SINGOLA USCITA DIGITALE TEMPORIZZATA

Codice: 27 169 bit tmp
Codice Hex: 1B A9 bit tmp
Mnemonic: ESC ASCII(169) ASCII(bit) ASCII(tmp)

Viene posta allo stato logico **1** (uscita attiva), per un determinato tempo, la linea di uscita digitale a relé identificata da **bit** che ha la seguente corrispondenza con i segnali su CN4:

1 -> OUT A1 **2** -> OUT A2 **3** -> OUT B1 **4** -> OUT B2

L'uscita selezionata rimane in questo stato per un periodo di tempo determinato dal parametro **tmp** e quindi ritorna allo stato logico **0** (uscita disattiva); il periodo di tempo è espresso in decine di msec e deve essere compreso nel range **1÷255**.

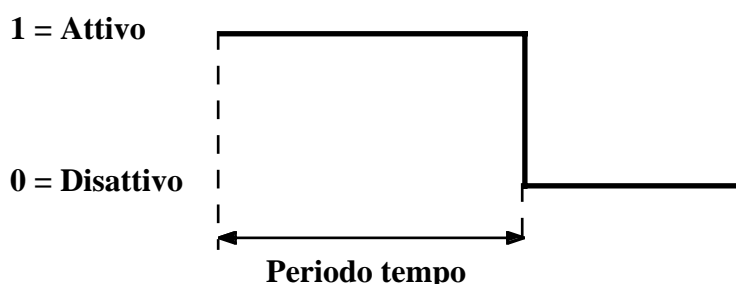


FIGURA 53: COMANDO ATTIVAZIONE SINGOLA USCITA TEMPORIZZATA

Se sulla linea di output in questione sono in corso altre temporizzazioni, queste vengono interrotte. Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vuole attivare l'uscita OUT A2 per un tempo di 500 msec, senza alterare le rimanenti uscite sarà necessario inviare la sequenza:

27 169 2 50 oppure **1B A9 02 32 Hex** oppure **ESC ASCII(169) STX 2**

DISATTIVAZIONE SINGOLA USCITA DIGITALE

Codice: 27 170 bit

Codice Hex: 1B AA bit

Mnemonico: ESC ASCII(170) ASCII(bit)

Viene posta allo stato logico **0** (uscita disattiva), la linea di uscita digitale a relé identificata da **bit** che ha la seguente corrispondenza con i segnali su CN4:

1 -> OUT A1 **2** -> OUT A2 **3** -> OUT B1 **4** -> OUT B2

Se sulla linea di output in questione sono in corso delle temporizzazioni, queste vengono interrotte. Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vuole disattivare l'uscita OUT A1 senza alterare le rimanenti uscite sarà necessario inviare la sequenza:

27 170 1 oppure **1B AA 01 Hex** oppure **ESC ASCII(170) SOH**

DISATTIVAZIONE SINGOLA USCITA DIGITALE TEMPORIZZATA

Codice: 27 171 bit tmp

Codice Hex: 1B AB bit tmp

Mnemonico: ESC ASCII(171) ASCII(bit) ASCII(tmp)

Viene posta allo stato logico **0** (uscita disattiva), per un determinato tempo, la linea di uscita digitale a relé identificata da **bit** che ha la seguente corrispondenza con i segnali su CN4:

1 -> OUT A1 **2** -> OUT A2 **3** -> OUT B1 **4** -> OUT B2

L'uscita selezionata rimane in questo stato per un periodo di tempo determinato dal parametro **tmp** e quindi ritorna allo stato logico **1** (uscita attiva); il periodo di tempo è espresso in decine di msec e deve essere compreso nel range **1÷255**.

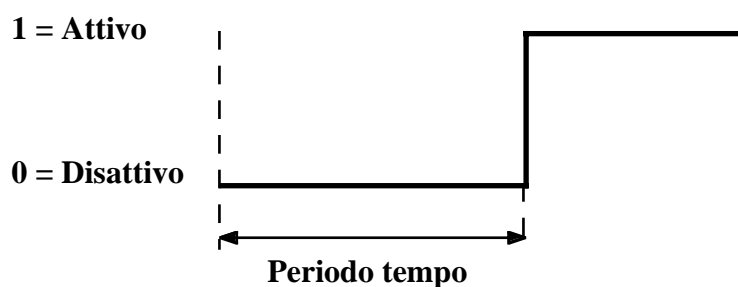


FIGURA 54: COMANDO DISATTIVAZIONE SINGOLA USCITA TEMPORIZZATA

Se sulla linea di output in questione sono in corso altre temporizzazioni, queste vengono interrotte. Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vuole attivare l'uscita OUT A1 per un tempo di 550 msec, senza alterare le rimanenti uscite sarà necessario inviare la sequenza:

27 171 1 55 oppure **1B AB 01 37 Hex** oppure **ESC ASCII(171) SOH 7**

ACQUISIZIONE SINGOLO INGRESSO DIGITALE

Codice: 27 172 *bit*
Codice Hex: 1B AC *bit*
Mnemonic: ESC ASCII(172) ASCII(*bit*)

Viene acquisito e restituito lo stato dell'ingresso digitale bufferato indicato dal parametro **bit** che ha la seguente corrispondenza con i segnali su CN5:

1 -> IN 1 2 -> IN 2 ÷ 7 -> IN 7 8 -> IN 8

Il valore restituito coincide con lo stato logico dell'input indicato; in particolare viene restituito il valore **0** (ingresso chiuso) oppure **1** (ingresso aperto).

Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vuole acquisire lo stato dell'ingresso IN 7 sarà necessario inviare la sequenza:

27 172 7 oppure 1B AC 07 Hex oppure ESC ASCII(172) BEL

ACQUISIZIONE SINGOLO INGRESSO DIGITALE CON DEBOUNCING

Codice: 27 173 *bit tmp*
Codice Hex: 1B AD *bit tmp*
Mnemonic: ESC ASCII(173) ASCII(*bit*) ASCII(*tmp*)

Viene acquisito e restituito lo stato con debouncing dell'ingresso digitale bufferato indicato dal parametro **bit** che ha la seguente corrispondenza con i segnali su CN5:

1 -> IN 1 2 -> IN 2 ÷ 7 -> IN 7 8 -> IN 8

A differenza del comando precedente l'acquisizione viene eseguita con un processo di debouncing, la cui durata, espressa in decine di msec, è stabilita dal parametro **tmp** che deve essere compreso nel range **1÷255**.

Il dato che viene restituito può assumere i seguenti valori:

0 -> *Ingresso chiuso durante tutto il tempo di debouncing*
1 -> *Ingresso aperto durante tutto il tempo di debouncing*
7 -> *Lo stato dell'ingresso è variato durante il tempo di debouncing*

Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vuole acquisire lo stato dell'ingresso IN 7, con un tempo di debouncing di 50 msec, sarà necessario inviare la sequenza:

27 173 7 5 oppure 1B AD 07 05 Hex oppure ESC ASCII(173) BEL ENQ

N.B. Il tempo di esecuzione di questo comando coincide con il tempo di debouncing passato quindi la risposta viene fornita una volta trascorso questo tempo. In caso di *firmware seriale* **.SER**, se dopo la sua trasmissione devono essere mandati numerosi altri comandi è preferibile inserire un ritardo che eviti il riempimento del buffer di ricezione.

ONDA QUADRA TEMPORIZZATA DA 1 SU SINGOLA USCITA DIGITALE

Codice: 27 174 bit tmp nper

Codice Hex: 1B AE bit tmp nper

Mnemonico: ESC ASCII(174) ASCII(bit) ASCII(tmp) ASCII(nper)

Viene generata un'onda quadra temporizzata, **con stato logico iniziale 1** e simmetria del 50% (uscita attiva e disattiva alternativamente ad intervalli uguali), sulla linea di uscita digitale a relé identificata da **bit** che ha la seguente corrispondenza con i segnali su CN4:

1 -> OUT A1 2 -> OUT A2 3 -> OUT B1 4 -> OUT B2

La durata del periodo di attivazione e disattivazione é stabilita dal parametro **tmp**, espresso in decine di msec e deve essere compreso nel range **1÷255**.

Il parametro **nper** invece indica il numero di periodi ovvero il numero di cambiamenti di stato che devono avvenire sull'uscita indicata; facendo riferimento alla figura seguente si nota che vengono eseguiti **nper** periodi equivalenti a **nper+1** variazioni. Anche questo dato deve essere compreso nel range **1÷255**.

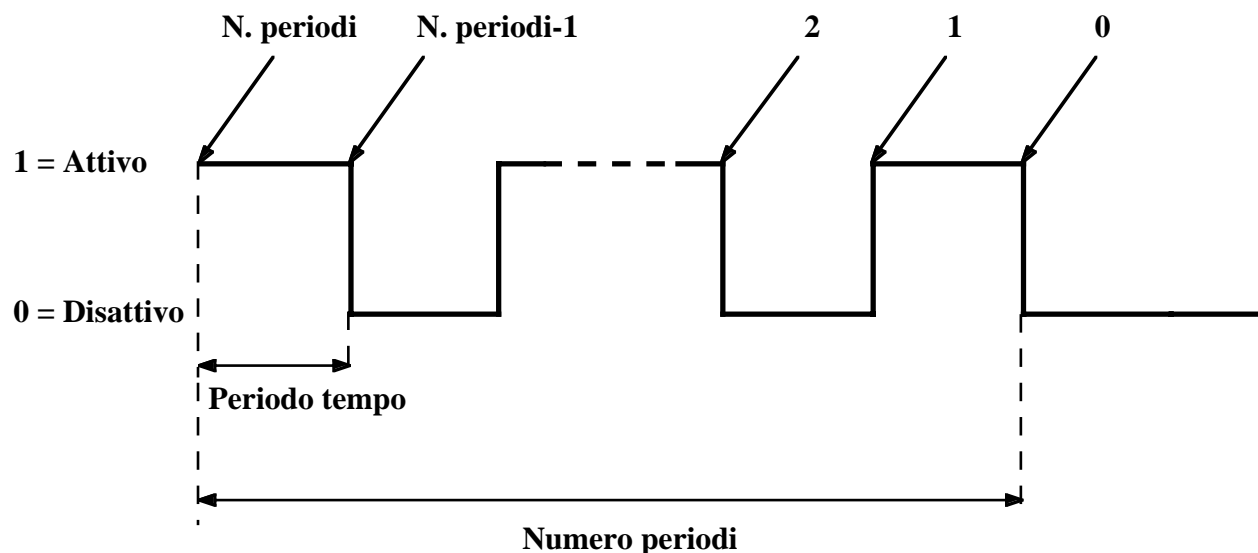


FIGURA 55: COMANDO ONDA QUADRA TEMPORIZZATA DA 1

Se sulla linea di output in questione sono in corso altre temporizzazioni, queste vengono interrotte. Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vogliono generare 10 periodi, ognuno di durata 250 msec, sull'uscita OUT B2 senza alterare le rimanenti uscite, sarà necessario inviare la sequenza:

27 174 4 25 10 oppure **1B AE 04 19 0A Hex** oppure **ESC ASCII(174) EOT EM LF**

N.B. Facendo riferimento alla figura 55 si può notare che lo stato che assumerà l'uscita al termine dell'esecuzione del comando dipende dal numero di periodi eseguiti; in particolare un numero pari di periodi comporterà uno stato finale **1** (output attivo) e viceversa.

ONDA QUADRA TEMPORIZZATA DA 0 SU SINGOLA USCITA DIGITALE

Codice: 27 175 bit tmp nper

Codice Hex: 1B AF bit tmp nper

Mnemonico: ESC ASCII(175) ASCII(bit) ASCII(tmp) ASCII(nper)

Viene generata un'onda quadra temporizzata, **con stato logico iniziale 0** e simmetria del 50% (uscita attiva e disattiva alternativamente ad intervalli uguali), sulla linea di uscita digitale a relé identificata da **bit** che ha la seguente corrispondenza con i segnali su CN4:

1 -> OUT A1 2 -> OUT A2 3 -> OUT B1 4 -> OUT B2

La durata del periodo di attivazione e disattivazione é stabilita dal parametro **tmp**, espresso in decine di msec e deve essere compreso nel range **1÷255**.

Il parametro **nper** invece indica il numero di periodi ovvero il numero di cambiamenti di stato che devono avvenire sull'uscita indicata; facendo riferimento alla figura seguente si nota che vengono eseguiti **nper** periodi equivalenti a **nper+1** variazioni. Anche questo dato deve essere compreso nel range **1÷255**.

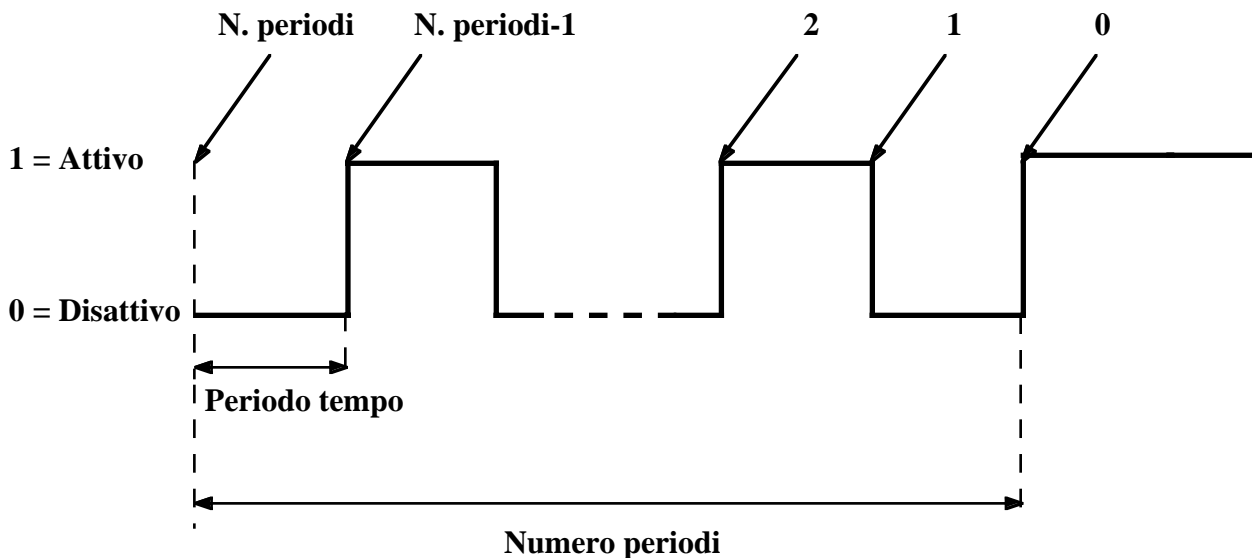


FIGURA 56: COMANDO ONDA QUADRA TEMPORIZZATA DA 0

Se sulla linea di output in questione sono in corso altre temporizzazioni, queste vengono interrotte. Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vogliono generare 40 periodi, ognuno di durata 200 msec, sull'uscita OUT B2 senza alterare le rimanenti uscite, sarà necessario inviare la sequenza:

27 175 4 20 40 oppure **1B AE 04 14 28 Hex** oppure **ESC ASCII(175) EOT DC4 (**

N.B. Facendo riferimento alla figura 56 si può notare che lo stato che assumerà l'uscita al termine dell'esecuzione del comando dipende dal numero di periodi eseguiti; in particolare un numero pari di periodi comporterà uno stato finale **0** (output disattivo) e viceversa.

ONDA QUADRA SU SINGOLA USCITA DIGITALE

Codice: 27 176 bit tmp

Codice Hex: 1B B0 bit tmp

Mnemonico: ESC ASCII(176) ASCII(bit) ASCII(tmp)

Viene generata un'onda quadra con simmetria del 50% (uscita attiva e disattiva alternativamente ad intervalli uguali), sulla linea di uscita digitale a relé identificata da **bit** che ha la seguente corrispondenza con i segnali su CN4:

1 -> OUT A1

2 -> OUT A2

3 -> OUT B1

4 -> OUT B2

La durata del periodo di attivazione e disattivazione é stabilita dal parametro **tmp**, espresso in decine di msec e deve essere compreso nel range **1÷255**.

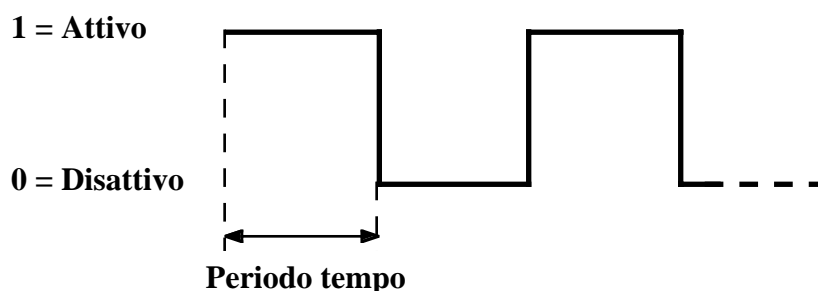


FIGURA 57: COMANDO ONDA QUADRA

Se sulla linea di output in questione sono in corso altre temporizzazioni, queste vengono interrotte. Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

Se, per esempio, si vuole generare un onda quadra con periodo di tempo 500 msec (pari ad un periodo dell'onda di 1000 msec e quindi ad una frequenza di 1 Hz), sull'uscita OUT B1 senza alterare le rimanenti uscite, sarà necessario inviare la sequenza:

27 176 3 50 oppure **1B B0 03 32 Hex** oppure **ESC ASCII(176) ETX 2**

N.B. La generazione dell'onda quadra iniziata da questo comando può essere terminata da un qualsiasi altro comando che riguarda le singole uscite digitali; i più adatti sono sicuramente quelli di ATTIVAZIONE e DISATTIVAZIONE SINGOLA USCITA DIGITALE che inoltre consentono di lasciare l'uscita nello stato desiderato.

Tra gli usi più frequenti di questo comando, e di quelli precedenti, si ricorda la generazione di basi temporali richieste da sistemi esterni, il comando di automatismi ripetitivi e periodici (ad esempio la piegatura di un pezzo), il comando di segnalazioni di stato (sirene, semafori, ecc.), lo spostamento programmato di pezzi, ecc.

COMANDI PER GESTIONI ALTERNATIVE DEGLI I/O DIGITALI

In aggiunta alle gestioni dei segnali di I/O digitale bufferato descritte nei precedenti paragrafi i firmware della **QTP 12/R84** offrono ulteriori gestioni definite alternative, che hanno sempre come obiettivo la soluzione dei problemi di automazione industriale. Per individuare i segnali di I/O usati in questi comandi si usano gli stessi nomi usati nella descrizione dei connettori (vedi figure 24÷29).

SETTAGGIO DEI PIN MULTIFUNZIONE

Codice: 27 179 bit cnf

Codice Hex: 1B B3 bit cnf

Mnemonic: ESC ASCII(179) ASCII(bit) ASCII(cnf)

Definisce la configurazione del segnale di I/O digitale bufferato identificato da **bit** che ha la seguente corrispondenza con i segnali su CN4 e CN5:

1 -> IN 1 2 -> IN 2 ÷ 7 -> IN 7 8 -> IN 8
9 -> OUT A1 10 -> OUT A2 11 -> OUT B1 12 -> OUT B2

La configurazione del pin, passata nel parametro **cnf**, definisce la sua funzionalità e per fornire flessibilità ed espandibilità al comando sono state previste 4 diverse configurazioni numerate da **0÷3**. Nell'attuale versione dei firmware sono previste solo alcune funzioni alternative su alcuni dei pin, come indicato nella seguente descrizione:

bit	cnf	Funzione
1÷12	0	-> Linea di ingresso uscita normale, senza funzioni alternative
1	1	-> Contatore hw a 16 bit associato a Timer 2 del microcontrollore
5	1	-> Contatore hw/fw a 16 bit associato ad /INT0 del microcontrollore
6	1	-> Contatore hw/fw a 16 bit associato ad /INT1 del microcontrollore
8	1	-> Contatore hw a 16 bit associato a Timer 1 del microcontrollore

Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

A seguito di un'accensione od inizializzazione, sono automaticamente disattivate tutte le funzioni alternative e tutti i segnali di I/O sono configurati come semplici ingressi ed uscite semplici.

Se, per esempio, si vuole configurare l'ingresso digitale IN 1 come contatore a 16 bit, sarà necessario inviare la sequenza:

27 179 1 1 oppure 1B B3 01 01 Hex oppure ESC ASCII(179) SOH SOH

LETTURA SETTAGGIO DEI PIN MULTIFUNZIONE

Codice: 27 180 bit

Codice Hex: 1B B4 bit

Mnemonic: ESC ASCII(180) ASCII(bit)

Restituisce la configurazione del segnale di I/O digitale bufferato identificato da **bit** con le modalità descritte nel precedente paragrafo.

Se, per esempio, si vuole leggere la configurazione dell'ingresso digitale IN 6, sarà necessario inviare la sequenza:

27 180 6 oppure 1B B4 06 Hex oppure ESC ASCII(180) ACK

e verrà restituito il valore **1** se l'ingresso è configurato come contatore a 16 bit oppure il valore **0** se l'ingresso non svolge funzioni alternative.

LETTURA CONTATORE

Codice: 27 177 cnt
Codice Hex: 1B B1 cnt
Mnemonico: ESC ASCII(177) ASCII(cnt)

Restituisce il valore attuale del contatore a 16 bits, identificato dal parametro **cnt** con la seguente corrispondenza:

cnt	segnale	Contatore
1	IN 8	-> Contatore hw a 16 bit associato a Timer 1 del microcontrollore
2	IN 1	-> Contatore hw a 16 bit associato a Timer 2 del microcontrollore
3	IN 5	-> Contatore hw/fw a 16 bit associato ad /INT0 del microcontrollore
4	IN 6	-> Contatore hw/fw a 16 bit associato ad /INT1 del microcontrollore

A seguito del comando sono restituiti due valori che costituiscono la combinazione a 16 bit attualmente memorizzata all'interno del contatore: di questi il primo coincide con il byte basso (LSB) ed il secondo il byte alto (MSB).

Indipendentemente dal contatore utilizzato il firmware riconosce come impulsi da contare il fronte di discesa del segnale, equivalente alla chiusura dell'ingresso optoisolato. Quando il contatore raggiunge il suo massimo valore, corrispondente a 65535 (FFFF Hex), ad un successivo impulso sul segnale d'ingresso, il nuovo valore della combinazione passa a 0.

Se il pin relativo al segnale d'ingresso non é configurato come contatore (tramite l'apposito comando SETTAGGIO PIN MULTIFUNZIONE), a seguito di questo comando, sarà sempre restituita la combinazione 0.

Qualora la sequenza inviata contiene dati non validi, il comando viene ignorato.

A seguito di un'accensione od inizializzazione tutti i contatori sono azzerati.

Se, per esempio, si vuole leggere il contatore 2 associato all'ingresso digitale IN 1, sarà necessario inviare la sequenza:

27 177 2 oppure **1B B1 02 Hex** oppure **ESC ASCII(177) STX**

e supponendo che vengano restituiti i due valori 100 e 15 (64 e 0F Hex) il contatore ha raggiunto la combinazione 3940 (0F64 Hex).

AZZERAMENTO CONTATORE

Codice: 27 178 cnt
Codice Hex: 1B B2 cnt
Mnemonico: ESC ASCII(178) ASCII(cnt)

Azzerà il valore attuale del contatore a 16 bits, identificato dal parametro **cnt** con la corrispondenza riportata nel precedente comando.

Tra gli usi più frequenti dei comandi relativi ai contatori si ricorda il conteggio di pezzi prodotti, l'acquisizione di sensori di posizione (anche encoder bidirezionali), il conteggio di transiti nei sistemi di controllo accesso, ecc.

COMANDI PER GRAFICA

La **QTP 12/R84-GF2**, con display grafico da 140 x 16 punti, fornisce all'utente la possibilità di rappresentare immagini grafiche, istogrammi, caratteri con diversi font e diverse dimensioni, grafici, ecc. grazie ad una serie di facili comandi. Tutti i comandi grafici sono basati sulle unità minime del display definite punti o pixel, ed in particolare sulle loro coordinate che sono organizzate come illustrato nella seguente figura:

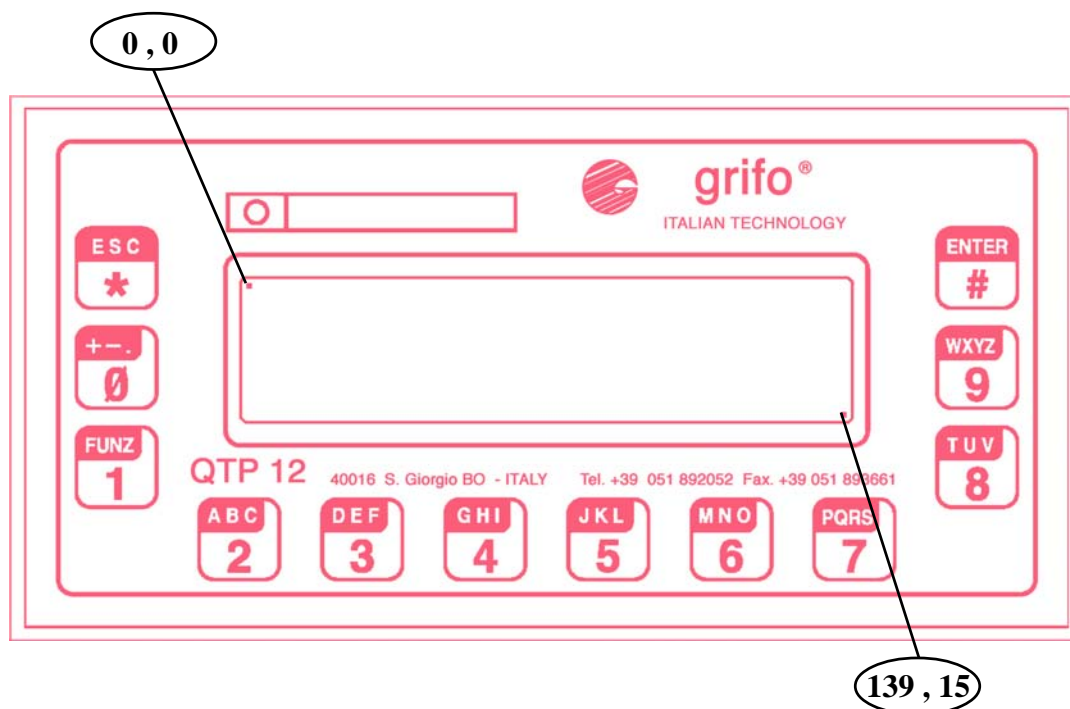


FIGURA 58: COORDINATE PUNTI DISPLAY GRAFICO

N.B. Si ricorda che i comandi grafici successivamente descritti sono utilizzabili solo sul modello **QTP 12/R84-GF2** mentre nei rimanenti modelli i caratteri dei comandi non vengono riconosciuti come comandi e quindi rappresentati sul display.
Si ricorda inoltre che il modello grafico **QTP 12/R84-GF2** riconosce tutti i comandi grafici ed anche tutti i comandi alfanumerici descritti nei paragrafi precedenti.

SETTAGGIO MODALITÀ ALFANUMERICA

Codice: 27 208
Codice Hex: 1B D0
Mnemonic: ESC ASCII(208)

Questo comando seleziona la modalità di rappresentazione alfanumerica con cui possono essere utilizzati tutti i comandi alfanumerici relativi alla rappresentazione, descritti nei paragrafi precedenti. Una volta eseguito il comando viene ripresentato il cursore (se attivo) nell'ultima posizione stabilita dai comandi precedentemente usati.

A seguito di un'accensione od inizializzazione, è automaticamente selezionata la modalità alfanumerica in modo da rendere subito disponibili tutte le funzionalità.



FIGURA 59: PRIMO ESEMPIO DI RAPPRESENTAZIONE GRAFICA



FIGURA 60: SECONDO ESEMPIO DI RAPPRESENTAZIONE GRAFICA

SETTAGGIO MODALITÀ GRAFICA

Codice: 27 209

Codice Hex: 1B D1

Mnemonic: ESC ASCII(209)

Questo comando seleziona la modalità di rappresentazione grafica con cui possono essere successivamente trasmessi i dati grafici al terminale senza che questi vengano interpretati come comandi. Visivamente questo comando non ha alcun effetto ma una volta eseguito, i caratteri ricevuti non vengono più verificati per i comandi ad un solo carattere bensì direttamente forniti al display grafico. Una volta settata la modalità grafica i comandi ad un solo carattere non hanno più effetto fino a quando non viene rileszionata la modalità alfanumerica.

A seguito di una accensione od inizializzazione, è automaticamente selezionata la modalità alfanumerica in modo da rendere subito disponibili tutte le funzionalità. I comandi a più di un carattere, che iniziano sempre con il carattere ESC = 27 = 1BH, sono invece sempre interpretati ed eseguiti indipendentemente dalla modalità scelta.

POSIZIONAMENTO ASSOLUTO DEL CURSORE GRAFICO

Codice: 27 206 y x 0

Codice Hex: 1B CE y x 00

Mnemonic: ESC ASCII(206) ASCII(y) ASCII(x) NUL

Sposta il cursore in corrispondenza del punto individuato dalle coordinate **x** e **y**; la posizione indicata da questi due parametri è assoluta, quindi indipendente da tutti gli altri parametri impostati e va al di fuori del normale posizionamento alfanumerico. I caratteri successivamente ricevuti, saranno visualizzati partendo dal punto indicato, sviluppandosi verso l'alto e verso destra.

I valori delle coordinate **y** e **x** devono essere rispettivamente compresi nei range **0÷15** e **0÷139**, ovvero nelle dimensioni del display utilizzato.

Se ad esempio, si desidera posizionare il cursore sul punto di coordinate (10,100), sarà necessario inviare la sequenza:

27 206 100 10 0 oppure **1B CE 64 0A 00** Hex oppure **ESC ASCII(206) d LF NUL**

N.B. Il codice **0** (NUL) presente al termine della sequenza, è stato inserito per compatibilità con espansioni future ed altri terminali: deve sempre essere fornito per una corretta esecuzione del comando.

SETTAGGIO AREA GRAFICA

Codice: 27 241 x1 y1 x2 y2 cmd

Codice Hex: 1B F1 x1 y1 x2 y2 cmd

Mnemonic: ESC ASCII(241) ASCII(x1) ASCII(y1) ASCII(x2) ASCII(y2) ASCII(cmd)

Definisce l'area grafica di lavoro e l'azione da utilizzare su quest'ultima. L'area grafica ha come vertice in alto a sinistra il punto con coordinate **x1** e **y1** e come vertice in basso a destra quello individuato da **x2** e **y2**. I valori di **y1**, **y2** ed **x1**, **x2** devono essere rispettivamente compresi nei range **0÷15** e **0÷139**, ovvero nelle dimensioni del display utilizzato.

Il byte **cmd** seleziona invece l'azione da effettuare sull'area grafica scelta e quindi anche la funzione dei successivi byte ricevuti dal firmware, come illustrato nel seguente elenco:

cmd	=	67	(43 Hex)	C	-> Cancella l'area grafica selezionata.
		70	(46 Hex)	F	-> Riempie l'area grafica selezionata.
		72	(48 Hex)	H	-> Disegna l'area grafica selezionata con i seguenti dati grafici orizzontali, con spostamento orizzontale.
		73	(49 Hex)	I	-> Inverte l'area grafica selezionata.
		79	(4F Hex)	O	-> Rappresenta cornice dell'area grafica selezionata.
		86	(56 Hex)	V	-> Disegna l'area grafica selezionata con i seguenti dati grafici verticali, con spostamento orizzontale.
		104	(68 Hex)	h	-> Disegna l'area grafica selezionata con i seguenti dati grafici orizzontali, con spostamento verticale.
		111	(6F Hex)	o	-> Cancella cornice dell'area grafica selezionata.
		118	(76 Hex)	v	-> Disegna l'area grafica selezionata con i seguenti dati grafici verticali, con spostamento verticale.

Nel caso dei comandi che disegnano l'area grafica (H,h,V,v) i successivi byte trasmessi al terminale saranno utilizzati come dati grafici che quindi determinano l'attivazione dei corrispondenti punti del display. La corrispondenza tra i bit di tali byte ed i punti del display é chiaramente riportata nelle successive figure 60÷63 in cui vengono descritte tutte le quattro possibili modalità di organizzazione e spostamento degli stessi dati. Inoltre si ricorda che lo stato logico **1** del bit corrisponde all'**attivazione** del relativo punto e viceversa lo stato logico **0** del bit corrisponde alla **disattivazione** del punto.

Il completamento del comando di settaggio dell'area grafica con disegno può avvenire in due modalità: o per riempimento dell'area grafica selezionata o per trasmissione di un altro comando; quest'ultimo naturalmente interrompe il primo e provoca la rappresentazione dei soli punti ricevuti sino a quel momento.

Se, ad esempio, si vuole disegnare una freccia partendo dall'angolo in alto a sinistra del display come quella illustrata nella seguente figura

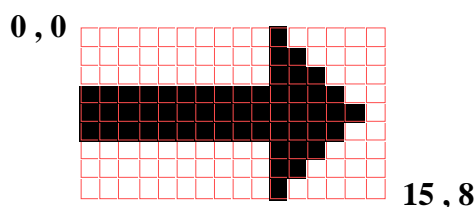


FIGURA 61: ESEMPIO DISEGNO AREA GRAFICA

sarà necessario inviare al firmware il comando:

27 241 0 0 15 8 72 oppure **1B F1 00 00 0F 08 48 Hex**

e poi la seguente sequenza di dati grafici:

0 0 0 255 255 255 0 0 0 32 48 56 252 254 252 56 48 32 oppure
00 00 00 FF FF FF 00 00 00 20 30 38 FC FE FC 38 30 20 Hex

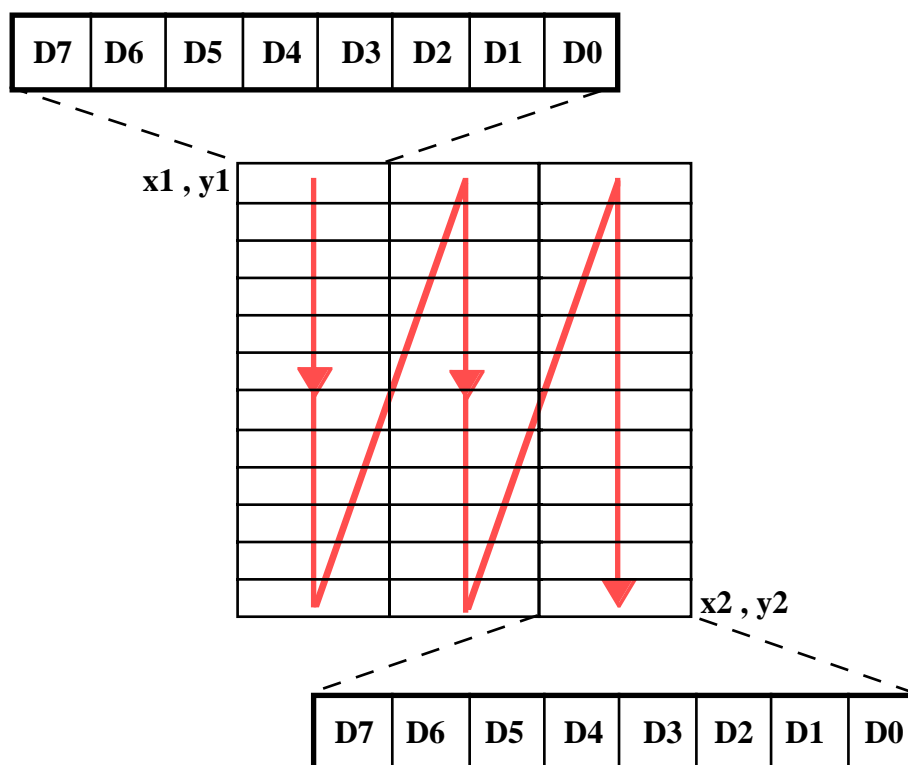


FIGURA 62: DATI GRAFICI ORIZZONTALI CON SPOSTAMENTO ORIZZONTALE

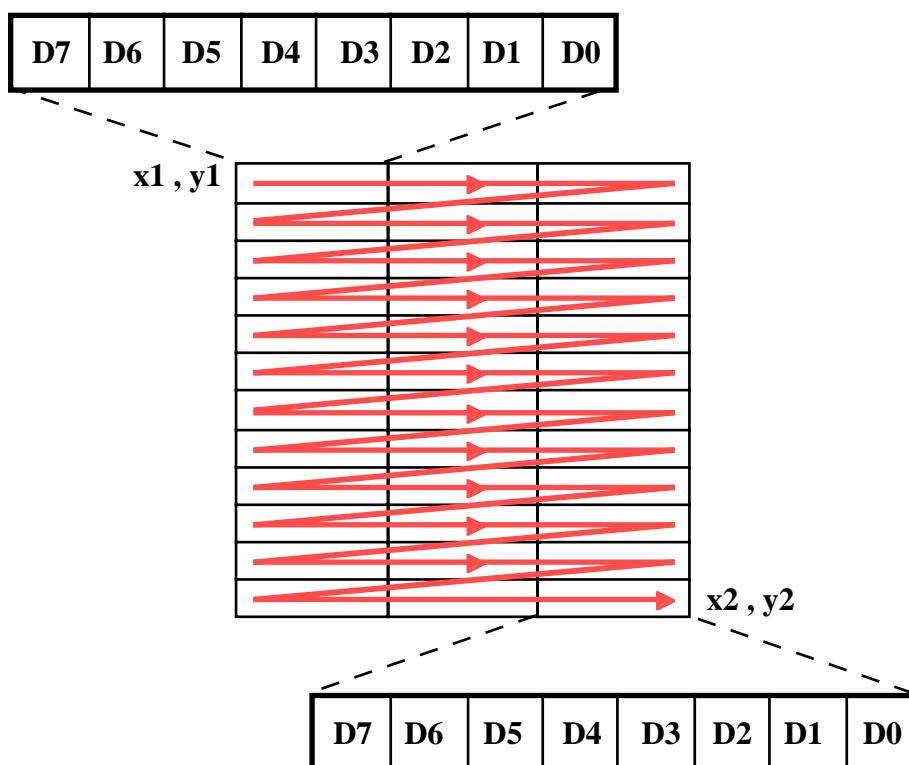


FIGURA 63: DATI GRAFICI ORIZZONTALI CON SPOSTAMENTO VERTICALE

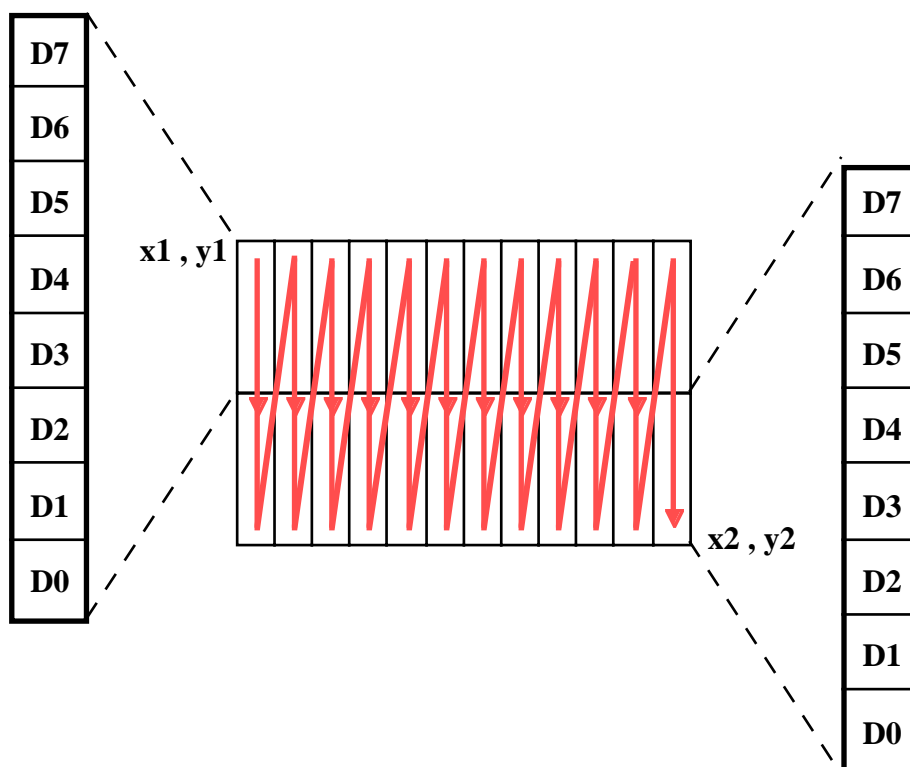


FIGURA 64: DATI GRAFICI VERTICALI CON SPOSTAMENTO ORIZZONTALE

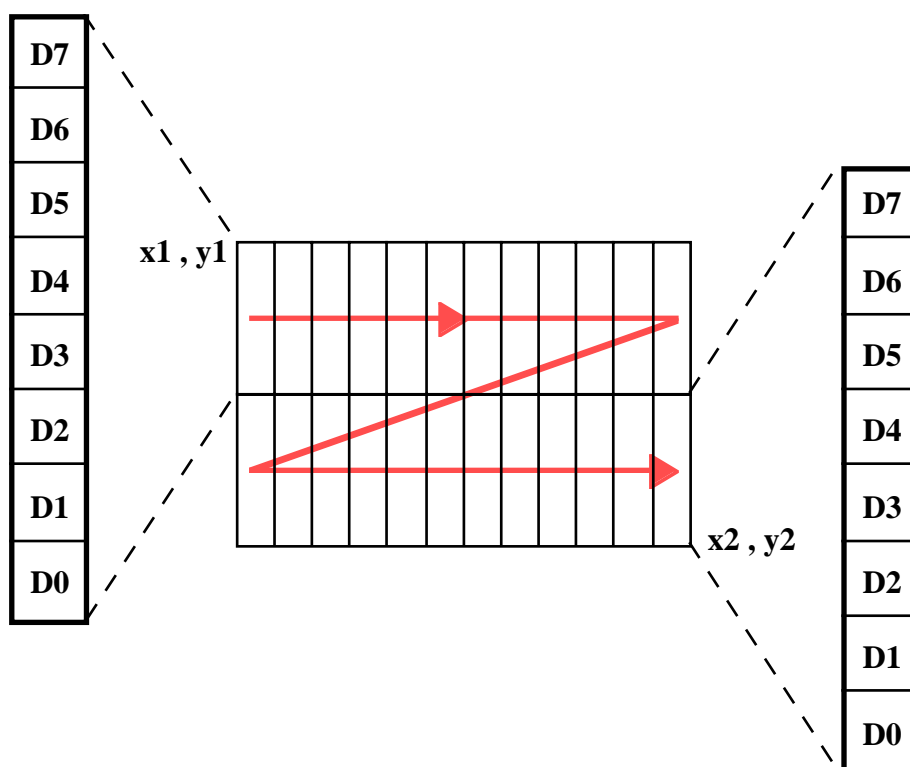


FIGURA 65: DATI GRAFICI VERTICALI CON SPOSTAMENTO VERTICALE

SETTAGGIO FONT GRAFICO

Codice: 27 242 font

Codice Hex: 1B F2 font

Mnemonic: ESC ASCII(242) ASCII(font)

Imposta il font utilizzato per le successive rappresentazioni alfanumeriche in modalità grafica. Quando è selezionata quest'ultima modalità e non è in corso un comando di disegno area grafica (che come già detto usa i byte ricevuti come dati grafici), i byte ricevuti vengono comunque rappresentati sul display come caratteri. Per quest'ultima condizione è definibile il font di rappresentazione che quindi si differenzia da quello utilizzato in modalità puramente alfanumerica. Il font disponibile è scelto dal parametro **font** con la seguente corrispondenza:

font =	65	(41 Hex)	A	-> Minifont proporzionale da 3x5÷5x5 punti.
	97	(61 Hex)	a	-> Minifont proporzionale da 3x5÷5x5 punti.
	66	(42 Hex)	B	-> Font Katakana da 5x7 punti.
	67	(43 Hex)	C	-> Font Katakana da 10x14 punti.
	98	(62 Hex)	b	-> Font Europeo da 5x7 punti.
	99	(63 Hex)	c	-> Font Europeo da 10x14 punti.
	49	(31 Hex)	1	-> Interspazio di 1 punto.
	50	(32 Hex)	2	-> Interspazio di 2 punti.

Si ricorda che i primi 6 comandi di selezione tipo di font sono a mutua esclusione, mentre gli ultimi due di selezione interspazio si sommano ai primi. Da questo si ricava che ognuno dei 5 font può essere settato con interspazio di 1 o 2 punti, ottenendo un totale di 10 font diversi. Il font impostato è utilizzato solo in modalità grafica mentre in alfanumerica è sempre usato il classico font di figura B1 con interspazio fisso ad 1 punto.

A seguito di una accensione od inizializzazione, è automaticamente attivata la modalità alfanumerica e selezionato il font Katakana 5x7, interspazio 1, per la modalità grafica.

Per maggiori informazioni sul set di caratteri disponibili con i font descritti si possono vedere le apposite tabelle nell'APPENDICE B del manuale, mentre la figura 36 mostra una foto in cui sono utilizzati tre dei 10 font disponibili.

COMANDI PER LA COMUNICAZIONE 1-WIRE

Le linee di I/O a livello TTL disponibili sulla **QTP 12/R84**, sono utilizzate dai firmware per l'implementazione di due BUS di comunicazione con il protocollo **1-WIRE**. In questo modo, sfruttando i comandi illustrati nei seguenti paragrafi é possibile gestire i vari dispositivi sviluppati per questo tipo di standard come sensori di temperatura, memorie, Dallas iButton™, ecc.

Vengono utilizzate le due linee di I/O TTL disponibili sul connettore J6, con la seguente corrispondenza:

Pin 2 J6	->	Linea 1-WIRE n. 1
Pin 3 J6	->	Linea 1-WIRE n. 2

Come si potrà notare dai successivi paragrafi, i comandi ad alto livello disponibili, non supportano la presenza di più di un dispositivo per ogni linea 1-WIRE, infatti ad esempio non é presente il comando "Search ROM" relativo all'individuazione dei codici ROM dei dispositivi presenti sulla linea.

Può comunque essere gestito un collegamento di varie unità in rete, sfruttando i comandi a basso livello (reset linea, scrittura e lettura bit, scrittura e lettura bytes); in questo caso però l'implementazione risulta più articolata e richiede anche un interscambio pesante fra il programma utente ed il firmware della **QTP 12/R84**.

E' quindi consigliato collegare alla scheda al massimo due dispositivi 1-WIRE; la figura seguente riporta un esempio di connessione con due sensori di temperatura Dallas DS18S20.

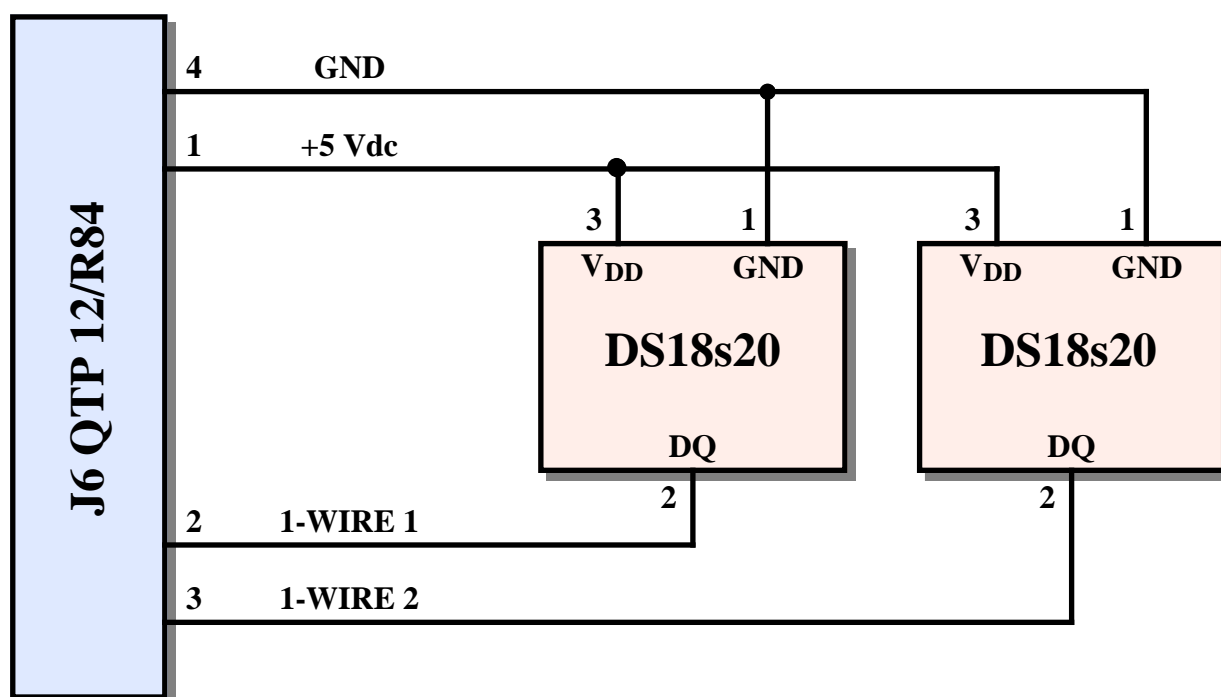


FIGURA 66: ESEMPIO DI COLLEGAMENTO CON DUE DISPOSITIVI 1-WIRE

Si consiglia all'utente di consultare anche una specifica documentazione generale sul protocollo 1-WIRE infatti questo manuale usa la nomenclatura, i dati ed i comandi di questo standard senza fornire informazioni specifiche sul loro significato.

RESET LINEA 1-WIRE

Codice: 27 181 wire
Codice Hex: 1B B5 wire
Mnemonico: ESC ASCII(181) ASCII(wire)

Viene inviata la sequenza di reset sulla linea 1-WIRE indicata dal parametro **wire**. Il valore di tale parametro deve essere **1** o **2** ed in caso contrario il comando viene ignorato.

Al termine della sequenza di reset viene acquisito e poi restituito l'impulso di presenza dall'eventuale dispositivo collegato sulla linea con la seguente corrispondenza:

0 -> Dispositivo 1-WIRE presente e pronto a ricevere il comando
1 -> Dispositivo 1-WIRE non presente

Se, ad esempio, si desidera inviare la sequenza di reset ed acquisire l'impulso di presenza sulla linea 1-WIRE n.1, sarà necessario inviare la sequenza:

27 181 1 oppure **1B B5 01 Hex** oppure **ESC ASCII(181) SOH**

SCRITTURA DI UN BIT SU LINEA 1-WIRE

Codice: 27 182 wire bit
Codice Hex: 1B B6 wire bit
Mnemonico: ESC ASCII(182) ASCII(wire) ASCII(bit)

Viene inviato il **bit** passato come parametro sulla linea 1-WIRE indicata dal parametro **wire**. I valori accettati per i due parametri sono rispettivamente **0** od **1** ed **1** e **2**; in caso contrario il comando viene ignorato.

Se, ad esempio, si desidera scrivere il bit 1 sulla linea 1-WIRE n.2, sarà necessario inviare la sequenza:

27 182 2 1 oppure **1B B6 02 01 Hex** oppure **ESC ASCII(182) STX SOH**

LETTURA DI UN BIT DA LINEA 1-WIRE

Codice: 27 183 wire
Codice Hex: 1B B7 wire
Mnemonico: ESC ASCII(183) ASCII(wire)

Viene acquisito un bit dalla linea 1-WIRE indicata dal parametro **wire**. Il valore di tale parametro deve essere **1** o **2** ed in caso contrario il comando viene ignorato.

A lettura effettuata viene restituito lo stato logico (**0** od **1**) del bit acquisito dalla linea indicata.

Se, ad esempio, si desidera leggere un bit dalla linea 1-WIRE n.1, sarà necessario inviare la sequenza:

27 183 1 oppure **1B B7 01 Hex** oppure **ESC ASCII(183) SOH**

SCRITTURA DI N BYTES SU LINEA 1-WIRE

Codice: 27 184 wire n dat0 ÷ datn

Codice Hex: 1B B8 wire n dat0 ÷ datn

Mnemonico: ESC ASCII(184) ASCII(wire) ASCII(n) ASCII(dat0) ÷ ASCII(datn)

Vengono inviati gli **n** bytes sulla linea 1-WIRE indicata dal parametro **wire**. Il valore di quest'ultimo deve essere **1** o **2**; il numero massimo di bytes da inviare a seguito del comando é **10**, pertanto il parametro **n** deve essere compreso nel range **1÷10 (01÷0A Hex)**. Infine gli **n** bytes **dat0 ÷ datn** possono essere compresi in tutto il range **0÷255 (0÷FF Hex)**.

Tale comando viene ignorato nel caso in cui la sequenza contenga dei dati non validi.

In caso di problemi seulla linea 1-WIRE o di nessun dispositivo collegato, il comando viene comunque eseguito senza ottenere alcun effetto.

Se, ad esempio, si desidera scrivere i tre bytes 23, 118 e 80 sulla linea 1-WIRE n. 2, sarà necessario inviare la sequenza:

27 184 2 3 23 118 80 oppure **1B B8 02 03 17 76 50 Hex** oppure
ESC ASCII(184) STX ETX ETB u P

LETTURA DI N BYTES DA LINEA 1-WIRE

Codice: 27 185 wire n

Codice Hex: 1B B9 wire n

Mnemonico: ESC ASCII(185) ASCII(wire) ASCII(n)

Vengono letti **n** bytes dalla linea 1-WIRE indicata dal parametro **wire**. Il valore di quest'ultimo deve essere **1** o **2**; il numero massimo di bytes leggibili con un unico comando é **10**, pertanto il parametro **n** deve essere compreso nel range **1÷10 (01÷0A Hex)**.

Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

A lettura effettuata vengono restituiti gli **n** bytes, letti dalla linea 1-WIRE nello stesso ordine in cui sono stati letti.

In caso di problemi seulla linea 1-WIRE o di nessun dispositivo collegato, il comando viene comunque eseguito senza ottenere alcun effetto e tutti i dati restituiti assumono il valore 255 (FF Hex).

Se, ad esempio, si desidera leggere quattro bytes dalla linea 1-WIRE n. 2, sarà necessario inviare la sequenza:

27 185 2 4 oppure **1B B9 02 04 Hex** oppure **ESC ASCII(185) STX EOT**
 e supponendo che il dispositivo 1-WIRE fornisca alla lettura i dati 23, 118, 80 e 13, verranno restituiti i seguenti dati:

23 118 80 13 oppure **17 76 50 0D Hex** oppure **ETB u P CR**

"READ ROM" SU LINEA 1-WIRE

Codice: 27 186 wire
Codice Hex: 1B BA wire
Mnemonic: ESC ASCII(186) ASCII(wire)

A seguito di questo comando, sulla linea 1-WIRE indicata dal parametro **wire**, vengono effettuate le seguenti operazioni:

- Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.
- Se il dispositivo é presente, viene inviato il comando "**Read ROM**" (codice **33 Hex** del protocollo 1-WIRE) ed acquisito il relativo codice ROM.

Il valore del parametro **wire** deve essere **1** o **2** ed in caso contrario il comando viene ignorato.

A seguito di questo comando vengono restituiti 8 bytes (**rom0÷rom7**) con il seguente significato:

Dispositivo 1-WIRE presente e comando eseguito correttamente:

gli 8 bytes della risposta corrispondono al codice ROM del dispositivo: **rom0**=Codice famiglia, **rom1÷rom6**=Numero seriale e **rom7**=CRC.

Dispositivo 1-WIRE non presente e comando non inviato:

gli 8 bytes della risposta contengono tutti il valore **0**.

Se, ad esempio, si vuole acquisire il codice ROM del dispositivo collegato alla linea 1-WIRE n. 2, sarà necessario inviare la sequenza:

27 186 2 oppure **1B BA 02 Hex** oppure **ESC ASCII(186) STX**

ed ipotizzando che il codice ROM del dispositivo 1-WIRE sia: Codice famiglia=16; Numero seriale=56, 198, 13, 0, 8, 0; CRC=226; la risposta a tale comando sarà:

16 56 198 13 0 8 0 226 oppure **10 38 C6 0D 00 08 00 E2 Hex** oppure
DLE 8 ASCII(198) CR NUL BS NUL ASCII(226)

"SKIP ROM" SU LINEA 1-WIRE

Codice: 27 188 wire
Codice Hex: 1B BC wire
Mnemonic: ESC ASCII(188) ASCII(wire)

A seguito di questo comando, sulla linea 1-WIRE indicata dal parametro **wire**, vengono effettuate le seguenti operazioni:

- Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.
- Se il dispositivo é presente, viene inviato il comando "**Skip ROM**" (codice **CC Hex** del protocollo 1-WIRE).

Il valore del parametro **wire** deve essere **1** o **2** ed in caso contrario il comando viene ignorato.

Al termine delle operazioni elencate viene restituito un dato che può assumere i seguenti valori:

0 -> *Dispositivo 1-WIRE presente e comando inviato*
1 -> *Dispositivo 1-WIRE non presente e comando non inviato*

Se, ad esempio, si vuole inviare il comando "Skip ROM" al dispositivo collegato alla linea 1-WIRE n. 1, sarà necessario inviare la sequenza:

27 188 1 oppure **1B BC 01 Hex** oppure **ESC ASCII(188) SOH**

"MATCH ROM" SU LINEA 1-WIRE

Codice: 27 187 wire rom0 ÷ rom7

Codice Hex: 1B BB wire rom0 ÷ rom7

Mnemonico: ESC ASCII(187) ASCII(wire) ASCII(rom0) ÷ ASCII(rom7)

A seguito di questo comando, sulla linea 1-WIRE indicata dal parametro **wire**, vengono effettuate le seguenti operazioni:

- Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.
- Se il dispositivo é presente, viene inviato il comando "**Match ROM**" (codice **55 Hex** del protocollo 1-WIRE) seguito dal codice ROM contenuto nei bytes: **rom0**=Codice famiglia, **rom1÷rom6**=Numero seriale, **rom7**=CRC.

Il valore del parametro **wire** deve essere **1** o **2**, mentre gli 8 bytes **rom0÷rom7** possono essere compresi in tutto il range **0÷255 (00÷FF Hex)**.

Tale comando viene ignorato nel caso in cui la sequenza contenga dei dati non validi.

Al termine delle operazioni elencate viene restituito un dato che può assumere i seguenti valori:

- | | | |
|----------|----|---|
| 0 | -> | Dispositivo 1-WIRE presente e comando inviato |
| 1 | -> | Dispositivo 1-WIRE non presente e comando non inviato |

Se, ad esempio, si vuole inviare il comando "Match ROM" al dispositivo collegato alla linea 1-WIRE n. 1 ed ipotizzando che il codice ROM sia: Codice famiglia=16; Numero seriale=56, 198, 13, 0, 8, 0; CRC=226; sarà necessario inviare la sequenza:

27 187 1 16 56 198 13 0 8 0 226 oppure 1B BB 01 10 38 C6 0D 00 08 00 E2 Hex
oppure ESC ASCII(187) SOH DLE 8 ASCII(198) CR NUL BS NUL ASCII(226)

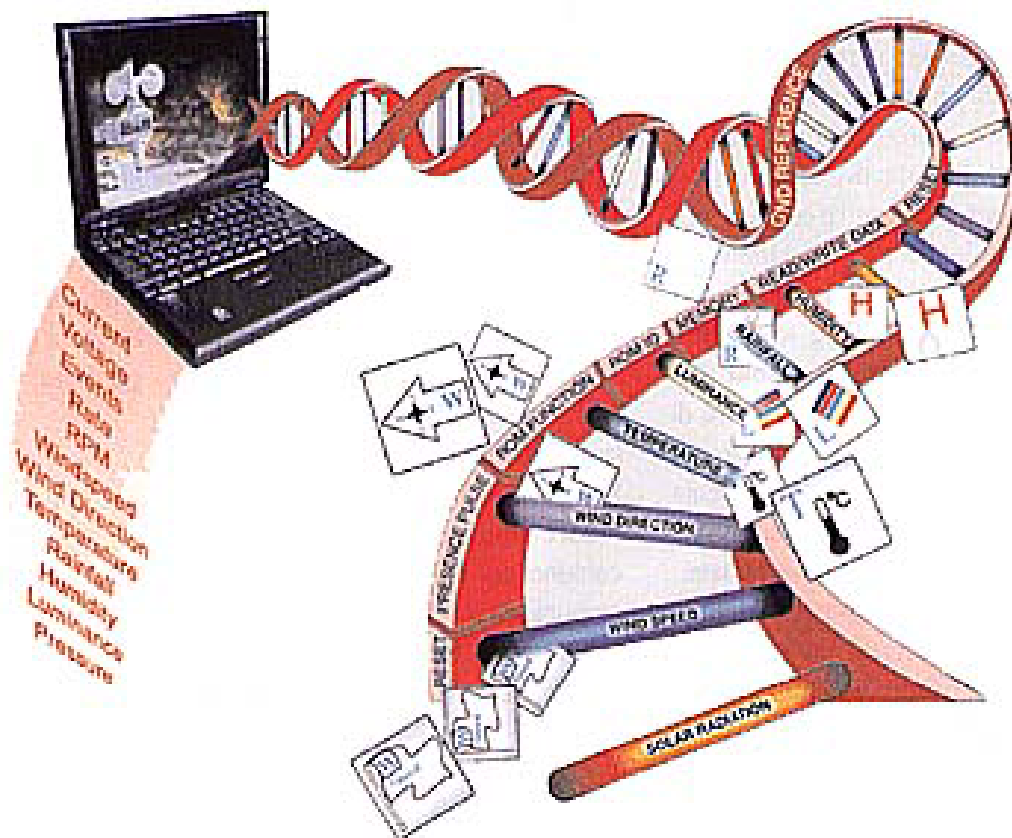


FIGURA 67: SCHEMA FUNZIONALE PROTOCOLLO 1-WIRE

"ALARM SEARCH" SU LINEA 1-WIRE

Codice: 27 189 wire

Codice Hex: 1B BD wire

Mnemonico: ESC ASCII(189) ASCII(wire)

A seguito di questo comando, sulla linea 1-WIRE indicata dal parametro **wire**, vengono effettuate le seguenti operazioni:

- a) Invio della sequenza di reset per verificare la presenza del dispositivo sulla linea e di conseguenza, predisporre lo stesso alla ricezione del successivo comando.
- b) Se il dispositivo é presente, viene inviato il comando "**Alarm search**" (codice **EC Hex** del protocollo 1-WIRE) e viene determinato se questo ha il flag di allarme settato o meno.

Il valore del parametro **wire** deve essere **1** o **2** ed in caso contrario il comando viene ignorato.

Al termine delle operazioni elencate viene restituito un dato che può assumere i seguenti valori:

- | | | |
|----------|----|---|
| 0 | -> | Dispositivo 1-WIRE presente con flag di allarme non settato |
| 1 | -> | Dispositivo 1-WIRE presente con flag di allarme settato |
| 7 | -> | Dispositivo 1-WIRE non presente e comando non inviato |

Se, ad esempio, si vuole inviare il comando "Alarm search" al dispositivo collegato alla linea 1-WIRE n. 1, sarà necessario inviare la sequenza:

27 189 1 oppure **1B BD 01 Hex** oppure **ESC ASCII(189) SOH**

SCHEDE ESTERNE

Le applicazioni caratteristiche della **QTP 12/R84** sono tutte quelle che comportano l'interazione con un operatore al fine di scambiare informazioni, misure, stati, ecc. relativi alla macchina o all'impianto controllato. La scheda ha la possibilità di essere affiancata alla maggior parte dei dispositivi programmabili, provvisti di una linea seriale di comunicazione; a titolo di esempio ne riportiamo un elenco con una breve descrizione delle caratteristiche di massima; per maggiori informazioni richiedere la documentazione specifica:

MSI 01

Multi Serial Interface 1 linea

Interfaccia per linea seriale TTL e linea bufferata in RS 232, RS 422 o Current loop. La seriale TTL é su un connettore a morsettiera e quella bufferata su un connettore plug standard.

IBC 01

Interface Block Communication

Scheda di conversioni per comunicazioni seriali. 2 linee RS 232; 1 linea RS 422-485; 1 linea in fibra ottica; interfaccia DTE/DCE selezionabile; attacco rapido per guide tipo DIN 46277-1 e 3.

GPC® 553

General Purpose Controller 80C552

Microprocessore 80C552 a 22 MHz. Completa implementazione CMOS; 32K EPROM; 32 K RAM; 32 K EEPROM o RAM; RTC; EEPROM; 1 linea RS 232 + 1 RS 232 o RS 422-485 o Current loop; 16 I/O TTL; 2 linee di PWM; timer/counter da 16 bits; watch dog; dip switch; 8 linee di A/D da 12 bit; interfaccia per **ABACO®** I/O BUS. Alimentazione in DC o AC; attacco rapido per guide DIN 46277-1 e 3.

GPC® 884

General Purpose Controller 80C188ES

Microprocessore AMD 80C188ES fino a 40M Hz. Completa implementazione CMOS; formato serie 4; 512K EPROM o FLASH; 512K RAM tamponata con batteria al litio; RTC; 1 linea RS 232 + 1 RS 232 o RS 422-485 o Current loop; 16 I/O TTL; 3 timer counter; 2 canali DMA; watch dog; EEPROM seriale; 11 linee di A/D da 12 bit; power failure; interfaccia per **ABACO®** I/O BUS.

GPC® 150

General Purpose Controller 84C15

Microprocessore Z80 a 16 MHz. completa implementazione CMOS; 512K EPROM o FLASH; 512K RAM; RTC; Back-Up con batteria al litio esterna; 4M FLASH seriale; 1 linea RS 232 + 1 RS 232 o RS 422-485 o Current loop; 40 I/O TTL; 2 timer/counter; 2 watch dog; dip switch; EEPROM linee di A/D da 12 bit; LED di attività.

GPC® R/T94

General Purpose Controller Relé/Transistor 9 ingressi 4 uscite

Microprocessore 89C4051 a 14 MHz. 4K FLASH; 128 Byte RAM; 256 Byte SRAM tamponata+RTC; 1K EEPROM seriale; 1 linea seriale TTL, RS 232, RS 422, RS 485 o Current loop; 9 ingressi galvanicamente isolati NPN visualizzati da LED; 4 uscite a relé (5 A) o transistor (4A 45 Vdc) galvanicamente isolate e visualizzate; 1 counter a 16 bit; ingresso analogico da 11 bits. Connettori a rapida estrazione; alimentazione a +5 Vdc o ampio range 8÷24 Vac; fornita in contenitore per aggancio a guide Ω tipo DIN 46277-1 e DIN 46277-3.

QTP G28

Quick Terminal Panel 28 tasti con LCD grafico

Interfaccia operatore provvista di display LCD grafico da 240x128 pixel retroilluminato con lampada a catodo freddo; tastiera a membrana da 28 tasti di cui 6 configurabili dall'utente; 16 LEDs di stato; alimentatore a bordo scheda; interdaccia seriale in RS 232, RS 422-485 o Current loop; linea seriale ausiliaria in RS 232. Tasti ed etichette personalizzabili dall'utente tramite serigrafie da inserire in apposite tasche; contenitore metallico e plastico; EEPROM di set up; 256K EPROM o FLASH; Real Time Clock; 128K RAM; buzzer. Firmware di gestione che svolge funzione di terminale con primitive grafiche.

UCC A2

UART Comunication Card 2 linee

2 indipendenti linee seriali in RS 232, RS 422, RS 485 o Current loop. Per ogni linea: buffer di 3 caratteri; comunicazione gestita dall'UART SCC 85C30; baud rate (da 50 a 115K baud), parità, stop bit e lunghezza dato programmabili via software; 4 dip switch. BUS a 8 bit; indirizzamento normale.

GMB HR84

grifo® Mini Block Housing, 8 opto inputs, 4 relays outputs

8 ingressi optoisolati NPN o PNP visualizzati da LEDs; Alcuni ingressi possono svolgere funzioni di conteggio ed interrupt; 4 uscite a relé da 5 A visualizzate da 4 LEDs. Alcune uscite possono svolgere funzioni evolute per comandi temporizzati automatici. Linea seriale in RS 232, RS 422, RS 485, Current loop e TTL. Connettori di espansione per linea I2C BUS, eventuale linea CAN ed eventuali segnali analogici. Alimentatore switching incorporato. Protezione su alimentazione della logica di bordo, tramite TransZorb. Alimentazione in DC o in AC a partire da 12Vdc, fino a 24Vac.

CAN GM Zero

CAN - **grifo®** Mini Modulo Zero

grifo® CAN Mini Modulo da 28 pins basata sulla CPU Atmel T89C51CC03 con 64K FLASH; 256 Bytes RAM; 2K ERAM; 2K FLASH per Bootloader; 2K EEPROM; 3 Timer Counter e 5 sezioni di Timer Counter ad alta funzionalita' (PWM, watch dog, comparazione); RTC + 240 Bytes SRAM, tamponati con batteria al Litio; I²C BUS; 17 linee di I/O TTL; 8 A/D 10 bit; linea seriale RS 232 o TTL; CAN; 2 LEDs di stato; Dip switch di configurazione; ecc.

SBP 02

Switch BLOCK Power 2 A

Alimentatore switching a basso costo in grado di generare una tensione fino a +40 Vdc con carico di 2 A; ingresso da 12 a 24 Vac; connettori a morsettiera a rapida estrazione; montaggio su guide ad Ω ; ingombro ridottissimo.

EXPS-2

EXternal Power Supply 2 tensioni

Alimentatore da spina da 75x55x90 mm completo di contenitore plastico. Tensione d'ingresso: 230 Vac, 50 Hz. Doppia tensione d'uscita galvanicamente isolate: 24 Vdc, 200 mA e 18 Vac 400 mA. Collegamenti normalizzati con spina da rete per la tensione d'ingresso e cavo intestato con capicorda per tensioni d'uscita.

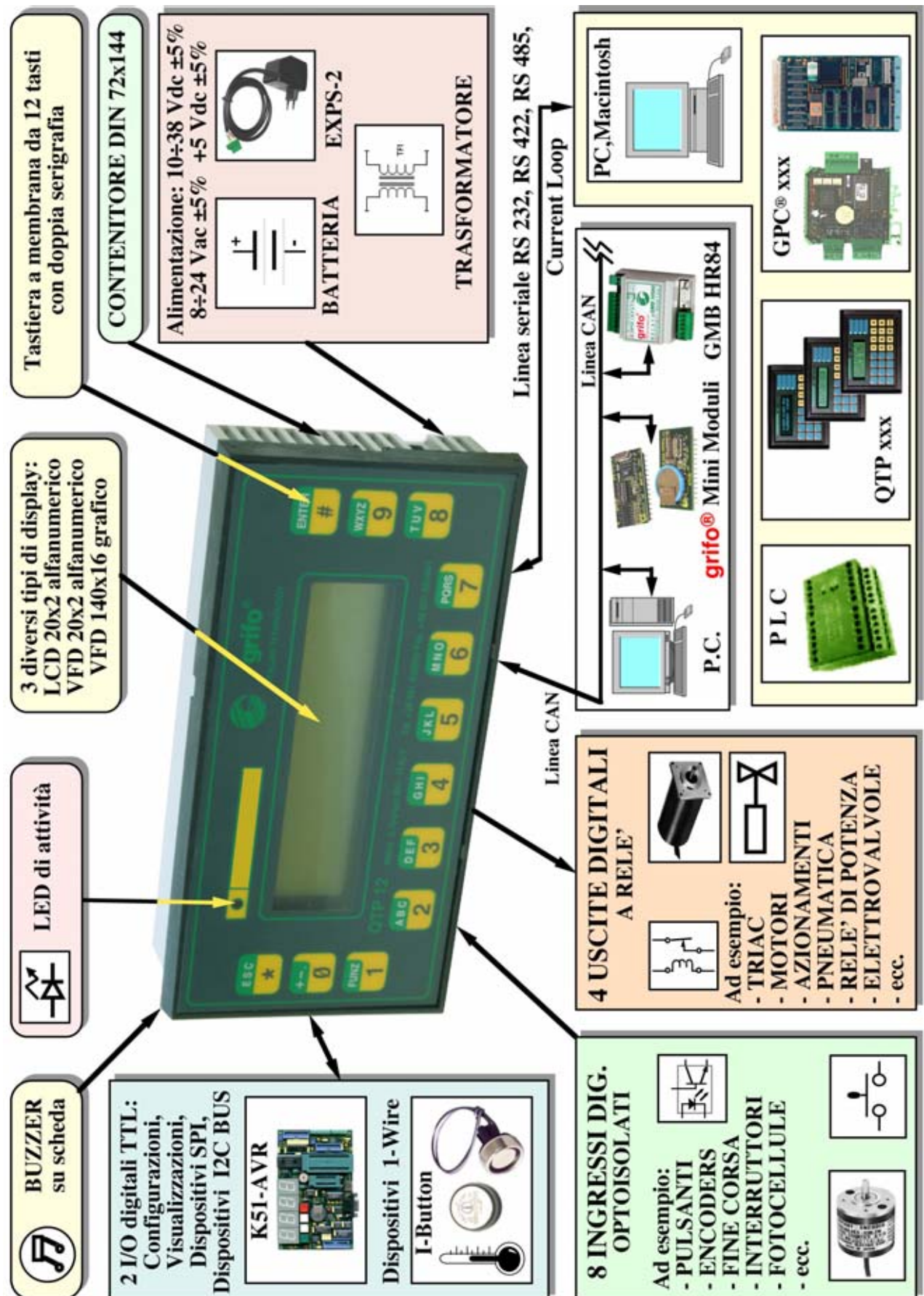


FIGURA 68: SCHEMA DELLE POSSIBILI CONNESSIONI

BIBLIOGRAFIA

E' riportato di seguito, un elenco di manuali e note tecniche, a cui l'utente può fare riferimento per avere maggiori chiarimenti, sui vari componenti montati a bordo della scheda **QTP 12/R84**.

Manuale ATMEL:	<i>Microcontroller - AT89 series</i>
Manuale HEWLETT PACKARD:	<i>Optoelectronics Designer's Catalog</i>
Manuale MAXIM:	<i>New Releases Data Book - Volume IV</i>
Fogli tecnici NATIONAL:	<i>LM2825 - Simple Switcer</i>
Manuale PHILIPS:	<i>Application notes and development tools for 80C51 microcontrollers</i>
Manuale PHILIPS:	<i>80C51 - Based 8-Bit Microcontrollers</i>
Manuale PHILIPS:	<i>I²C-bus compatible ICs</i>
Fogli tecnici S.E.:	<i>SI series - Switching power supply</i>
Manuale SGS-THOMSON:	<i>Small signal transistor - Data Book</i>
Manuale TAKAMISAWA:	<i>Relays index Book</i>
Manuale TEXAS INSTRUMENTS:	<i>The TTL Data Book - SN54/74 Families</i>
Manuale TEXAS INSTRUMENTS:	<i>RS-422 and RS-485 Interface Circuits</i>
Manuale TOSHIBA:	<i>Photo couplers - Data Book</i>

Per reperire questi manuali fare riferimento alle case produttrici ed ai relativi distributori locali. In alternativa si possono ricercare le medesime informazioni o gli eventuali aggiornamenti ai siti internet delle case elencate.

APPENDICE A: TABELLE RIASSUNTIVE COMANDI

Sono di seguito riportate le tabelle riassuntive con i comandi dei firmware della **QTP 12/R84**. Come in tutto il manuale le sequenze di comando sono riportate nelle tre forme decimale, esadecimale e mnemonico mentre l'ultima colonna riporta il numero dei dati restituiti dal comando.

Comando	Codice	Codice HEX	Mnemonico	Ris.
Cursore a inizio	01	01	SOH	0
Cursore a sinistra	21	15	NACK	0
Cursore a destra	06	06	ACK	0
Cursore in basso	10	0A	LF	0
Cursore in alto	26	1A	SUB	0
Ritorno a capo riga	13	0D	CR	0
Ritorno a capo+nuova riga	29	1D	GS	0
Posizione assoluta cursore alfanumerico	27 89 r c	1B 59 r c	ESC Y ASCII(r) ASCII(c)	0
Spazio indietro	08	08	BS	0
Cancella pagina	12	0C	FF	0
Cancella riga	25	19	EM	0
Cancella fino a fine riga	27 75	1B 4B	ESC K	0
Cancella fino a fine pagina	27 107	1B 6B	ESC k	0
Disattivazione cursore	27 80	1B 50	ESC P	0
Attivazione cursore fisso	27 79	1B 4F	ESC O	0
Attivazione cursore “blocco” lampeggiante	27 81	1B 51	ESC Q	0
Lettura numero di versione	27 86	1B 56	ESC V	3
Lettura codice scheda	27 160	1B A0	ESC ASCII(160)	1
Settaggio modalità operativa	27 65 modo	1B 41 modo	ESC A ASCII(mod)	0
Reset generale	27 162	1B A2	ESC ASCII(162)	0
Reset comunicazione	27 163	1B A3	ESC ASCII(163)	0
Settaggio luminosità display fluorescente	27 108 lum	1B 6C lum	ESC I ASCII(lum)	0
Attivazione temp. BUZZER	07	07	BEL	0
Attivazione LED e BUZZER	27 50 disp attr	1B 32 disp attr	ESC 2 ASCII(dis) ASCII(attr)	0

FIGURA A1: TABELLA CODICI DEI COMANDI (1 DI 4)

Comando	Codice	Codice HEX	Mnemonico	Ris.
Richiesta disponibilità EEPROM	27 51	1B 33	ESC 3	1
Scrittura byte di presenza	27 33 78 byte	1B 21 4E byte	ESC ! N ASCII(byte)	0
Lettura byte di presenza	27 33 110	1B 21 6E	ESC ! n	1
Scrittura byte su EEPROM	27 164 addl addh byte	1B A4 addl addh byte	ESC ASCII(164) ASCII(addl) ASCII(addh) ASCII(byte)	0
Lettura byte da EEPROM	27 165 addl addh	1B A5 addl addh	ESC ASCII(165) ASCII(addl) ASCII(addh)	1
Riconfigurazione tasto	27 55 n.tasto cod.	1B 37 n.tasto cod.	ESC 7 ASCII(n.tasto) ASCII(cod.)	0
Attivazione keyclick senza memorizzazione	27 53	1B 35	ESC 5	0
Disattivazione keyclick senza memorizzazione	27 54	1B 36	ESC 6	0
Attivazione keyclick con memorizzazione	27 33 53	1B 21 35	ESC ! 5	0
Disattivazione keyclick con memorizzazione	27 33 54	1B 21 36	ESC ! 6	0
Definizione di un carattere utente	27 66 ncar Pat0÷Pat7	1B 42 ncar Pat0÷Pat7	ESC B ASCII(ncar) ASCII(Pat0)÷ASCII(Pat7)	0
Definizione e salvataggio di un carattere utente	27 33 66 ncar Pat0÷Pat7	1B 21 42 ncar Pat0÷Pat7	ESC ! B ASCII(ncar) ASCII(Pat0)÷ASCII(Pat7)	0
Lettura numero masimo messaggi	27 110	1B 6E	ESC n	1
Memorizzazione messaggio	27 33 67 n.mess. car.0÷car.19	1B 21 43 n.mess. car.0÷car.13	ESC ! C ASCII(n.mess.) ASCII(car.0)÷ASCII(car.19)	0
Lettura messaggio	27 33 69 n.mess.	1B 21 45 n.mess.	ESC ! E ASCII(n.mess.)	20
Visualizzazione di n messaggi	27 33 68 n.mess. n	1B 21 44 n.mess. n	ESC ! D ASCII(n.mess.) ASCII(n)	0
Visualizzazione di messaggi a scorrimento	27 33 83 n.mess. n.car	1B 21 53 n.mess. n.car	ESC ! S ASCII(n.mess.) ASCII(n.car)	0

FIGURA A2: TABELLA CODICI DEI COMANDI (2 DI 4)

Comando	Codice	Codice HEX	Mnemonico	Ris.
Scrittura di tutte le uscite digitali	27 166 out	1B A6 out	ESC ASCII(166) ASCII(out)	0
Lettura di tutti gli ingressi digitali	27 167	1B A7	ESC ASCII(167)	1
Attivazione singola uscita digitale	27 168 bit	1B A8 bit	ESC ASCII(168) ASCII(bit)	0
Attivazione singola uscita digitale temporizzata	27 169 bit tmp	1B A9 bit tmp	ESC ASCII(169) ASCII(bit) ASCII(tmp)	0
Disattivazione singola uscita digitale	27 170 bit	1B AA bit	ESC ASCII(170) ASCII(bit)	0
Disattivazione singola uscita digitale temporizzata	27 171 bit tmp	1B AB bit tmp	ESC ASCII(171) ASCII(bit) ASCII(tmp)	0
Acquisizione singolo ingresso digitale	27 172 bit	1B AC bit	ESC ASCII(172) ASCII(bit)	1
Acquisizione singolo ingresso digitale con debouncing	27 173 bit tmp	1B AD bit tmp	ESC ASCII(173) ASCII(bit) ASCII(tmp)	1
Onda quadra temporizzata da "1" su singola uscita digitale	27 174 bit tmp nper	1B AE bit tmp nper	ESC ASCII(174) ASCII(bit) ASCII(tmp) ASCII(nper)	0
Onda quadra temporizzata da "0" su singola uscita digitale	27 175 bit tmp nper	1B AF bit tmp nper	ESC ASCII(175) ASCII(bit) ASCII(tmp) ASCII(nper)	0
Onda quadra su singola uscita digitale	27 176 bit tmp	1B B0 bit tmp	ESC ASCII(176) ASCII(bit) ASCII(tmp)	0
Lettura contatore	27 177 cnt	1B B1 cnt	ESC ASCII(177) ASCII(cnt)	2
Azzeramento contatore	27 178 cnt	1B B2 cnt	ESC ASCII(178) ASCII(cnt)	0
Settaggio dei pin multifunzione	27 179 bit cnf	1B B3 bit cnf	ESC ASCII(179) ASCII(bit) ASCII(cnf)	0
Lettura settaggio dei pin multifunzione	27 180 bit	1B B4 bit	ESC ASCII(180) ASCII(bit)	1

FIGURA A3: TABELLA CODICI DEI COMANDI (3 DI 4)

Comando	Codice	Codice HEX	Mnemonico	Ris.
Reset linea 1-WIRE	27 181 wire	1B B5 wire	ESC ASCII(181) ASCII(wire)	1
Scrittura bit su linea 1-WIRE	27 182 wire bit	1B B6 wire bit	ESC ASCII(182) ASCII(wire) ASCII(bit)	0
Lettura bit da linea 1-WIRE	27 183 wire	1B B7 wire	ESC ASCII(183) ASCII(wire)	1
Scrittura n bytes su linea 1-WIRE	27 184 wire n dat0÷datn	1B B8 wire n dat0÷datn	ESC ASCII(184) ASCII(wire) ASCII(n) ASCII(dat0)÷ASCII(datn)	0
Lettura n bytes da linea 1-WIRE	27 185 wire n	1B B9 wire n	ESC ASCII(185) ASCII(wire) ASCII(n)	n
"Read ROM" su linea 1-WIRE	27 186 wire	1B BA wire	ESC ASCII(186) ASCII(wire)	8
"Match ROM" su linea 1-WIRE	27 187 wire rom0÷rom7	1B BB wire rom0÷rom7	ESC ASCII(187) ASCII(wire) ASCII(rom0)÷ASCII(rom7)	1
"Skip ROM" su linea 1-WIRE	27 188 wire	1B BC wire	ESC ASCII(188) ASCII(wire)	1
"Alarm search" su linea 1-WIRE	27 189 wire	1B BD wire	ESC ASCII(189) ASCII(wire)	1
Posizione assoluta cursore grafico	27 206 y x	1B CE y x	ESC ASCII(206) ASCII(y) ASCII(x)	0
Settaggio modalità alfanumerica	27 208	1B D0	ESC ASCII(208)	0
Settaggio modalità grafica	27 209	1B D1	ESC ASCII(209)	0
Settaggio area grafica	27 241 x1 y1 x2 y2 cmd	1B F1 x1 y1 x2 y2 cmd	ESC ASCII(241) ASCII(x1) ASCII(y1) ASCII(x2) ASCII(y2) ASCII(cmd)	0
Settaggio font grafico	27 242 font	1B F2 font	ESC ASCII(242) ASCII(font)	0

FIGURA A4: TABELLA CODICI DEI COMANDI (4 DI 4)

APPENDICE B: CARATTERI DEI DISPLAY

Le seguenti tabelle riportano i set di caratteri che vengono rappresentati dalla **QTP 12/R84** per tutti i possibili codici ricevuti, differenziati a seconda del display, e quindi del modello, ordinato ed a seconda delle impostazioni effettuate tramite gli appositi comandi.

Anche i caratteri non ASCII (o caratteri speciali) si differenziano a seconda del display e qualora l'utente necessiti di caratteri diversi da quelli descritti nelle seguenti figure, può contattare direttamente la **grifo®**.

L \ H	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00	User chr 0			0	Q	P	`	P	Ä	E		—	9	3	0	P
01	User chr 1		!	1	A	Q	a	9	Ä	e	a	7	7	4	Ä	9
02	User chr 2		"	2	B	R	b	r	Ä	E	r	ı	ı	ı	E	0
03	User chr 3		#	3	C	S	c	s	Ä	R	j	ı	ı	E	0	0
04	User chr 4		\$	4	D	T	d	t	Ä	e	\	ı	ı	ı	ı	0
05	User chr 5		%	5	E	U	e	u	E	0	.	ı	ı	ı	ı	0
06	User chr 6		&	6	F	V	f	v	0	*	ı	ı	ı	ı	ı	ı
07	User chr 7		'	7	G	W	g	w	0	ı	ı	ı	ı	ı	ı	ı
08	User chr 0		(8	H	X	h	x	0	ı	ı	ı	ı	ı	ı	ı
09	User chr 1)	9	I	Y	i	y	0	ı	ı	ı	ı	ı	ı	ı
0A	User chr 2		*	:	J	Z	j	z	0	ı	ı	ı	ı	ı	ı	ı
0B	User chr 3		+	;	K	L	k	l	0	ı	ı	ı	ı	ı	ı	ı
0C	User chr 4		,	<	L	*	ı	ı	\	ı	ı	ı	ı	ı	ı	ı
0D	User chr 5		—	=	M	I	m	ı	*	ı	ı	ı	ı	ı	ı	ı
0E	User chr 6		.	>	N	^	n	ı	ı	ı	ı	ı	ı	ı	ı	ı
0F	User chr 7		/	?	0	_	0	ı	ı	ı	ı	ı	ı	ı	ı	ı

FIGURA B1: TABELLA CARATTERI QTP 12/R84-F2, GF2 IN MODALITÀ ALFANUMERICA

		Higher 4-bit (D4 to D7) of Character Code (Hexadecimal)															
L	H	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		User chr 0	User chr 1	User chr 2	User chr 3	User chr 4	User chr 5	User chr 6	User chr 7	User chr 0	User chr 1	User chr 2	User chr 3	User chr 4	User chr 5	User chr 6	User chr 7
Lower 4-bit (D0 to D3) of Character Code (Hexadecimal)	0	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	1	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	2	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	3	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	4	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	5	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	6	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	7	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	8	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	9	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	A	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	B	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	C	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	D	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	E	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆
	F	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆	⬆

FIGURA B2: TABELLA CARATTERI QTP 12/R84-C2

L \ H	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00				0	a	P										
01			:	1	A	G										
02			L	2	E	R										
03			H	3	C	S										
04			h	4	D	T										
05			%	5	E	U										
06			s	h	F	U										
07			9	7	G	N										
08			(8	H	N										
09)	9	I	Y										
0A			x	:	T	Z										
0B			÷	:	H	E										
0C			.	<	L	B										
0D			-	=	M	J										
0E			.	>	N	^										
0F			/	?	O	...										

FIGURA B3: TABELLA CARATTERI QTP 12/R84-GF2 IN MODALITÀ GRAFICA CON MINIFONT

L \ H	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00		I		0	Q	P	`	P	Ä	E		—	ヲ	ミ	α	P
01		I	!	1	A	Q	a	9	Ä	8	„	7	チ	4	Ä	9
02		I	“	2	B	R	b	r	Ä	E	“	イ	ツ	×	E	θ
03		I	#	3	C	S	c	s	Ä	R	„	ウ	テ	E	ε	ω
04		I	\$	4	D	T	d	t	Ä	8	„	エ	ト	ト	Π	α
05		I	%	5	E	U	e	u	E	O	„	オ	ナ	1	ε	U
06		I	&	6	F	V	f	v	O	8	ヲ	カ	ニ	ヨ	ρ	Σ
07		I	'	7	G	W	g	w	ä	◊	ア	チ	ヌ	ヲ	9	π
08		I	(8	H	X	h	x	Ø	l	イ	ウ	ホ	リ	γ	Σ
09		I)	9	I	Y	i	y	ø	5	◊	ア	リ	ル	γ	9
0A		I	*	:	J	Z	j	z	U	Δ	エ	コ	ハ	ル	j	チ
0B		I	+	;	K	L	k	l	U	Δ	エ	サ	ヒ	ロ	×	ア
0C		I	,	<	L	*	l	l	\	Δ	エ	サ	ヒ	ロ	×	ア
0D		I	—	=	M	I	m	l	Δ	エ	ユ	ズ	ハ	コ	ト	÷
0E		I	„	>	N	^	n	÷	ω	↑	ヨ	エ	ホ	γ	ñ	
0F		I	/	?	O	_	o	←	5	↓	ッ	リ	マ	“	ö	I

FIGURA B4: TABELLA CARATTERI QTP 12/R84-GF2IN MODALITÀ GRAFICA CON FONT KATAKANA

L \ H	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00				0	a	P	`	P	E	Z		"	A	D	À	á
01			!	1	A	Q	a	q	°	Q	i	±	A	N	á	ñ
02			"	2	B	R	b	r	°	E	q	2	A	ò	ä	ö
03			#	3	C	S	c	s	l	x	E	3	A	ó	å	ö
04			\$	4	D	T	d	t	L	÷	Q	'	A	ö	ä	ö
05			%	5	E	U	e	u	w	0	*	°	A	ö	ä	ö
06			&	6	F	V	f	v	r	°	i	°	E	ö	æ	ö
07			'	7	G	W	g	w	ä	E	S	-	°	X	°	÷
08			(8	H	X	h	x	e	Δ	"	.	E	ø	è	ø
09)	9	I	Y	i	y	h	Δ	0	1	E	ò	é	ò
0A			*	:	J	Z	j	z	0	*	Δ	°	E	ó	ë	ó
0B			+	;	K	L	k	l	λ	Γ	Δ	°	E	ö	ë	ö
0C			,	<	L	*	l	l	π	0	°	°	E	ü	ü	ü
0D			-	=	M	I	m	l	τ	J		°	E	ý	ý	ý
0E			.	>	N	^	n	÷	φ	0	0	°	E	ï	ï	ï
0F			/	?	O	_	o	÷	0	°	°	°	E	ï	ï	ï

FIGURA B5: TABELLA CARATTERI QTP 12/R84-GF2 IN MODALITÀ GRAFICA CON FONT EUROPEO



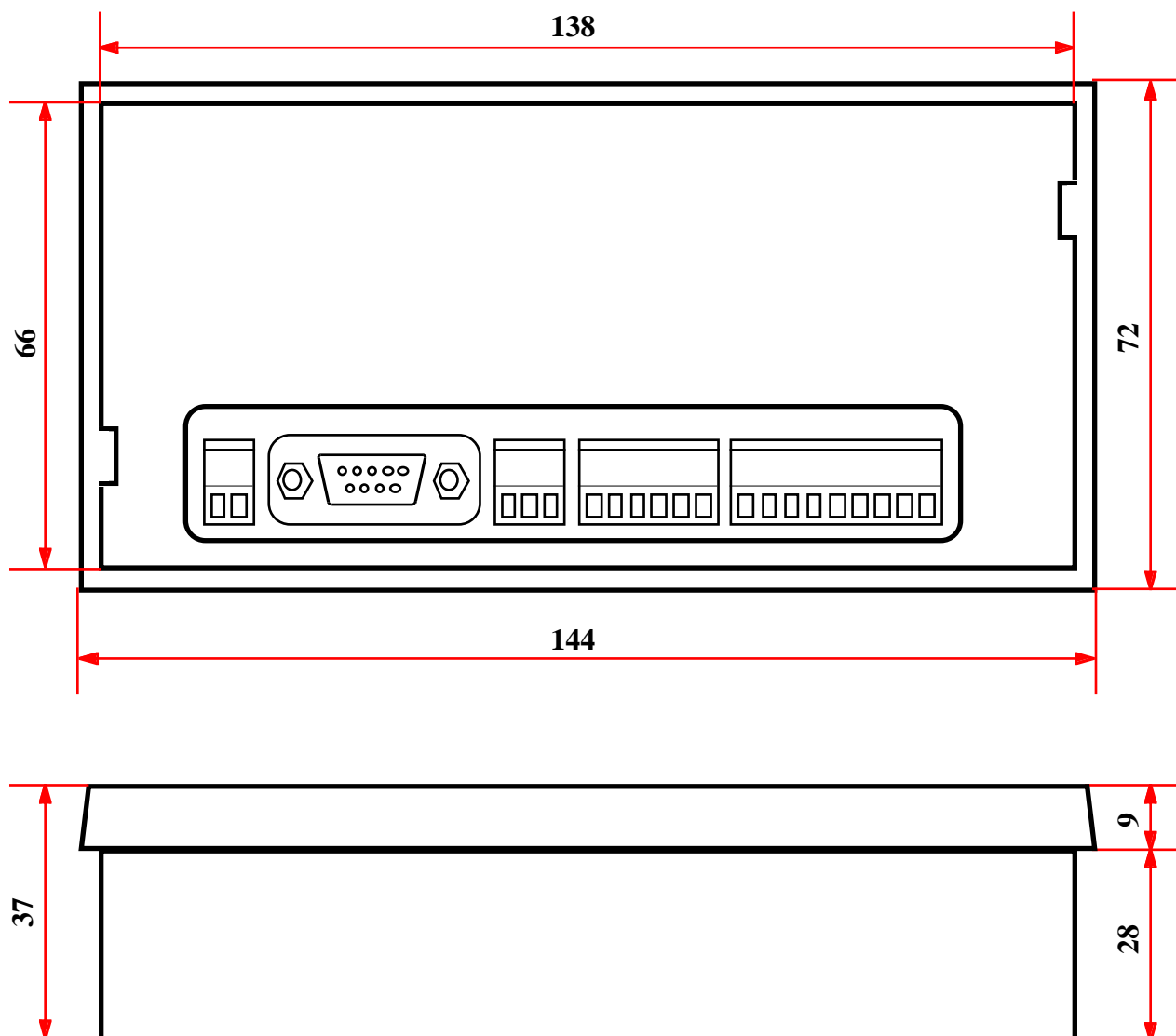
ITALIAN TECHNOLOGY



APPENDICE C: NOTE PER IL MONTAGGIO

QUOTE DEL TERMINALE

Nella successiva figura sono riportate le quote del terminale **QTP 12/R84** relative al contenitore metallico esterno e della allegata cornice plastica anteriore. Tali quote sono espresse in **mm** ed i disegni sono in scala 1:1.

**FIGURA C1: QUOTE QTP 12/R84**

Si ricorda che le quote sono riferite al solo contenitore e che l'area occupata può essere leggermente superiore se si tiene conto delle staffe di montaggio e delle viti di chiusura, fino ad un massimo di 156 x 72 x 80 mm (L x A x P).

MONTAGGIO MODALITÀ AVANQUADRO

Il montaggio previsto é quello in modalità avanquadro su un qualsiasi pannello di spessore massimo 10 mm ed il fissaggio avviene tramite due apposite staffe fornite assieme alla **QTP 12/R84**. Le operazioni da effettuare per un corretto montaggio sono di estrema facilità e possono essere così riassunte:

- 1) praticare uno scasso rettangolare sul pannello di montaggio come quello descritto nella seguente figura;

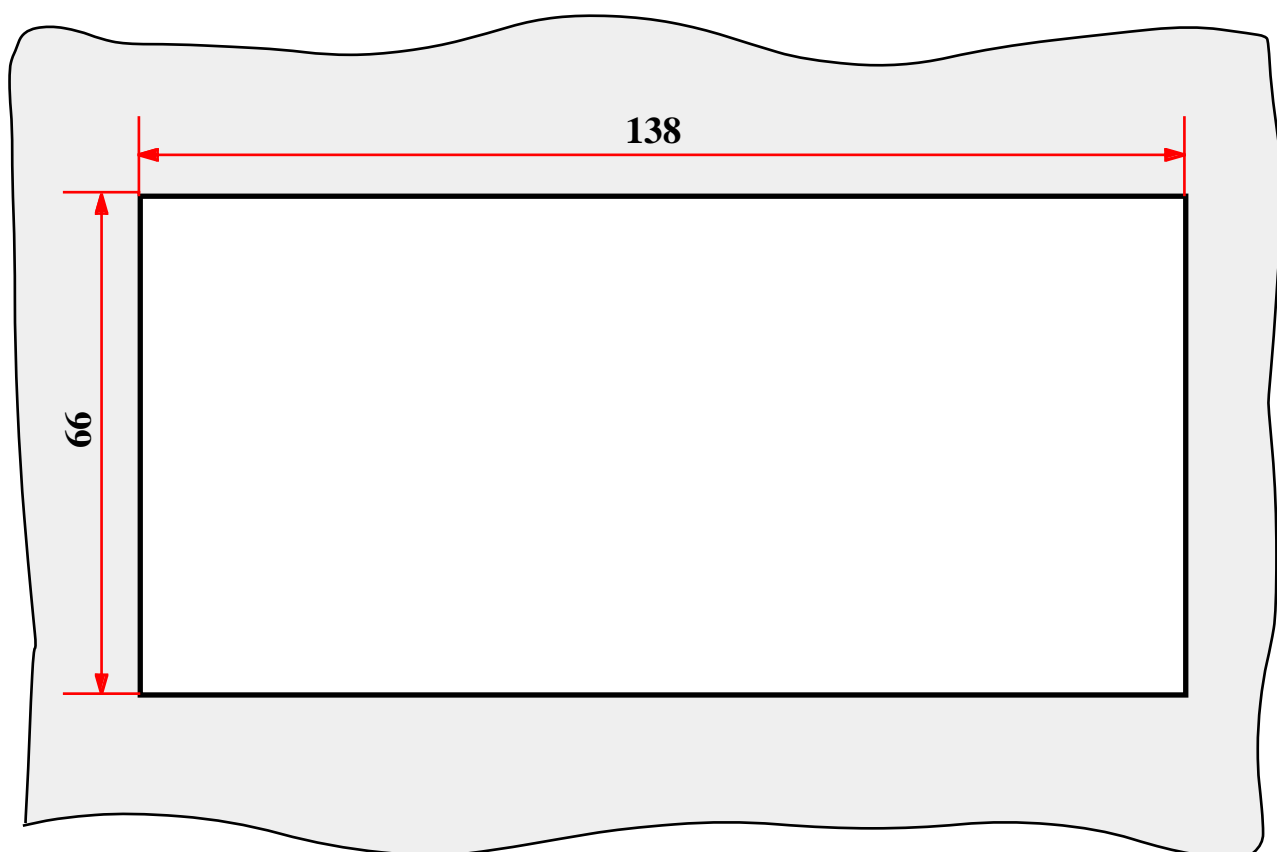


FIGURA C2: SCASSO DI MONTAGGIO

- 2) avvitare le due viti sulle due staffe a **C**, mantenendo la parte appuntita vicino al foro filettato della staffa;
- 3) infilare la **QTP 12/R84** nello scasso effettuato al punto 1;
- 4) agganciare le due staffe preparate al punto 2 negli appositi incastri laterali della **QTP 12/R84** facendo attenzione che il primo gancio della staffa, quello vicino al foro filettato, si incastri correttamente nell'asola del contenitore;
- 5) avvitare le due viti delle staffe fino a quando il contenitore della **QTP 12/R84** non é ben ancorato al pannello di montaggio;
- 6) collegare i connettori.

INSERIMENTO ETICHETTA DI PERSONALIZZAZIONE

Il frontale della **QTP 12/R84** è provvisto di una tasca di personalizzazione, in cui l'utente può mettere un'etichetta con proprio logo, un codice di identificazione, la funzione del LED di attività, od altro. Se si desidera inserire un'etichetta conviene farlo prima di montare la QTP. Questa deve essere realizzata dall'utente utilizzando un materiale sottile, ma nello stesso tempo abbastanza rigido, come ad esempio carta da 160 g/m² oppure un foglio di poliestere o di polycarbonato. Nella figura riportata di seguito sono illustrate le dimensioni, in millimetri, consigliate di tale etichetta; da notare che la zona bianca é quella che si troverà all'interno della relativa finestra, quindi l'unica visibile.

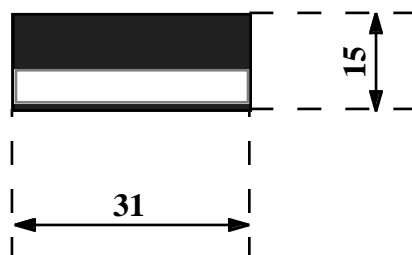


FIGURA C3: DIMENSIONI DELL'ETICHETTA DI PERSONALIZZAZIONE

Di seguito vengono riportate le operazioni da eseguire per inserire l'etichetta di personalizzazione all'interno della **QTP 12/R84**.

- 1) Svitare le due viti nere dal pannello frontale (se presenti).
- 2) Rimuovere il gruppo contenitore posteriore+cornice dal gruppo tastiera+circuito stampato. Per eseguire comodamente questa operazione é sufficiente effettuare una pressione sui connettori della **QTP 12/R84**.
- 3) Ora la tastiera é pronta per l'inserimento dell'etichetta di personalizzazione; questa deve essere introdotta, dall'alto, sfruttando l'apposita finestra presente nella parte posteriore del pannello tastiera, come illustrato nella figura seguente. Da notare che, come previsto in figura C3, le dimensioni dell'etichetta devono essere superiori a quelle della relativa finestra in modo da facilitare l'inserimento e l'estrazione.
- 4) Rimontare il terminale **QTP 12/R84**, seguendo le precedenti indicazioni in ordine inverso.

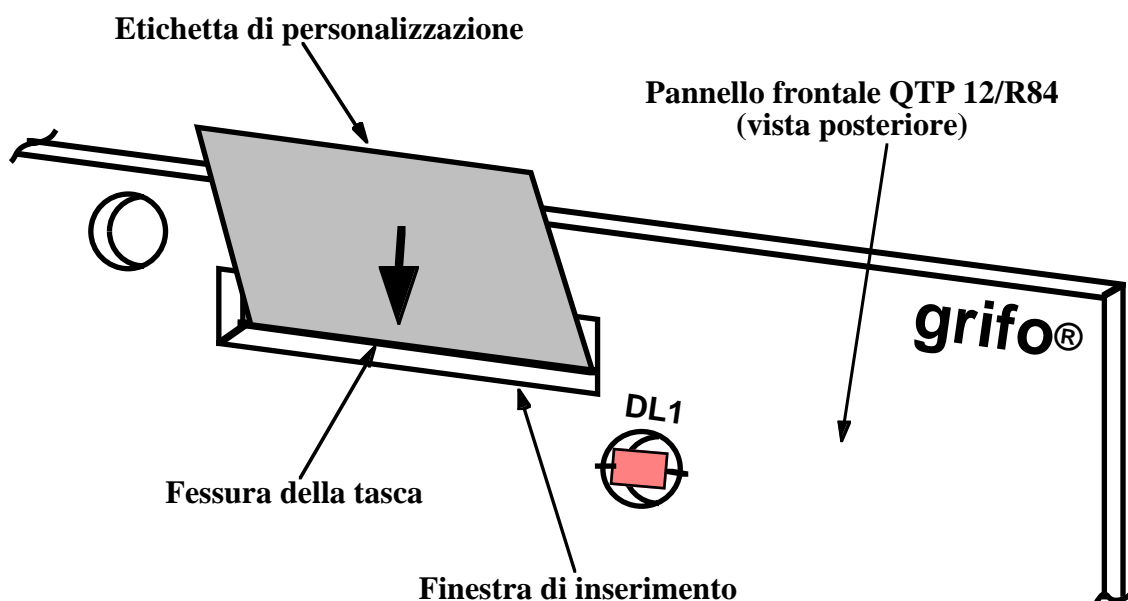


FIGURA C4: INSERIMENTO ETICHETTA DI PERSONALIZZAZIONE

FISSAGGIO DEL PANNELLO FRONTALE AL CONTENITORE

La **QTP 12/R84**, viene normalmente fornita con il pannello frontale (tastiera+circuito stampato), fissato ad incastro, nella cornice plastica del contenitore posteriore. Il terminale é comunque predisposto per un migliore ancoraggio meccanico tra questi due componenti, effettuato con due apposite viti, in modo da evitare eventuali fuori uscite accidentali del pannello frontale.

Le operazioni da eseguire per assicurare tale ancoraggio sono riportate di seguito:

- 1) Rimuovere il gruppo carter posteriore+cornice dal gruppo tastiera+circuito stampato. Per eseguire comodamente questa operazione é sufficiente effettuare una pressione sui connettori della **QTP 12/R84**, oppure sullo stampato raggiungibile dalla fessura del posteriore del contenitore dedicata appunto ai connettori.
- 2) Svitare le due viti centrali, delle sei che bloccano la cornice al carter posteriore.
- 3) Sul pannello frontale con tastiera, in corrispondenza di queste viti, sono già presenti due fori, correlativa svasatura, visibili solo dalla parte posteriore. E' quindi sufficiente bucare il rivestimento in policarbonato, che ricopre il frontale, in modo da rendere accessibili i due fori.
- 4) Rimontare il tutto, utilizzando le stesse due viti del punto 2. Queste però saranno avvitate sul pannello frontale, invece che sulla cornice anteriore.



APPENDICE D: DATI TECNICI

La **grifo®** fornisce un servizio di documentazione tecnica totalmente gratuito attraverso il proprio sito internet in cui possono essere scaricati i data sheets completi dei componenti usati a bordo scheda. Si rimanda quindi l'utente a tali documenti, di cui viene riportato il percorso sia tramite i link che tramite l'URL completo, assieme alle prime pagine degli stessi documenti.

T89C51CC01 E T89C51AC2

Link: Home | Servizio Documentazione Tecnica | ATMEL W&µC | Data-Sheet AT89C51CC01

URL: <http://www.grifo.it/PRESS/DOC/Temic/AT89C51CC01.pdf>

Features

- 80C51 Core Architecture
- 256 Bytes of On-chip RAM
- 1K Bytes of On-chip ERAM
- 32K Bytes of On-chip Flash Memory
 - Data Retention: 10 Years at 85°C
 - Read/Write Cycle: 10K
- 2K Bytes of On-chip Flash for Bootloader
- 2K Bytes of On-chip EEPROM
 - Read/Write Cycle: 100K
- 14-sources 4-level Interrupts
- Three 16-bit Timers/Counters
- Full Duplex UART Compatible 80C51
- Maximum Crystal Frequency 40 MHz
 - In X2 Mode, 20 MHz (CPU Core, 40 MHz)
- Five Ports: 32 + 2 Digital I/O Lines
- Five-channel 16-bit PCA with:
 - PWM (8-bit)
 - High-speed Output
 - Timer and Edge Capture
- Double Data Pointer
- 21-bit WatchDog Timer (7 Programmable Bits)
- A 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs
- Full CAN Controller:
 - Fully Compliant with CAN Rev2.0A and 2.0B
 - Optimized Structure for Communication Management (Via SFR)
 - 15 Independent Message Objects:
 - Each Message Object Programmable on Transmission or Reception
 - Individual Tag and Mask Filters up to 29-bit Identifier/Channel
 - 8-byte Cyclic Data Register (FIFO)/Message Object
 - 16-bit Status and Control Register/Message Object
 - 16-bit Time-Stamping Register/Message Object
 - CAN Specification 2.0 Part A or 2.0 Part B Programmable for Each Message Object
 - Access to Message Object Control and Data Registers Via SFR
 - Programmable Reception Buffer Length Up To 15 Message Objects
 - Priority Management of Reception of Hits on Several Message Objects at the Same Time (Basic CAN Feature)
 - Priority Management for Transmission
 - Message Object Overrun Interrupt
 - Supports:
 - Time Triggered Communication
 - Autobaud and Listening Mode
 - Programmable Automatic Reply Mode
 - 1-Mbit/s Maximum Transfer Rate at 8 MHz ⁽¹⁾ Crystal Frequency in X2 Mode
 - Readable Error Counters
 - Programmable Link to On-chip Timer for Time Stamping and Network Synchronization
 - Independent Baud Rate Prescaler
 - Data, Remote, Error and Overload Frame Handling
- On-chip Emulation Logic (Enhanced Hook System)
- Power Saving Modes:
 - Idle Mode
 - Power-down Mode

1. At BRP = 1 sampling point will be fixed.



**Enhanced 8-bit
MCU with CAN
Controller and
Flash Memory**

T89C51CC01

Rev. 4129E-8051-03/02





- Power Supply: 5V \pm 10% (or 3V ⁽¹⁾ \pm 10%)
- Temperature Range: Industrial (-40° to +85°C)
- Packages: VQFP44, PLCC44, CA-BGA64

Description

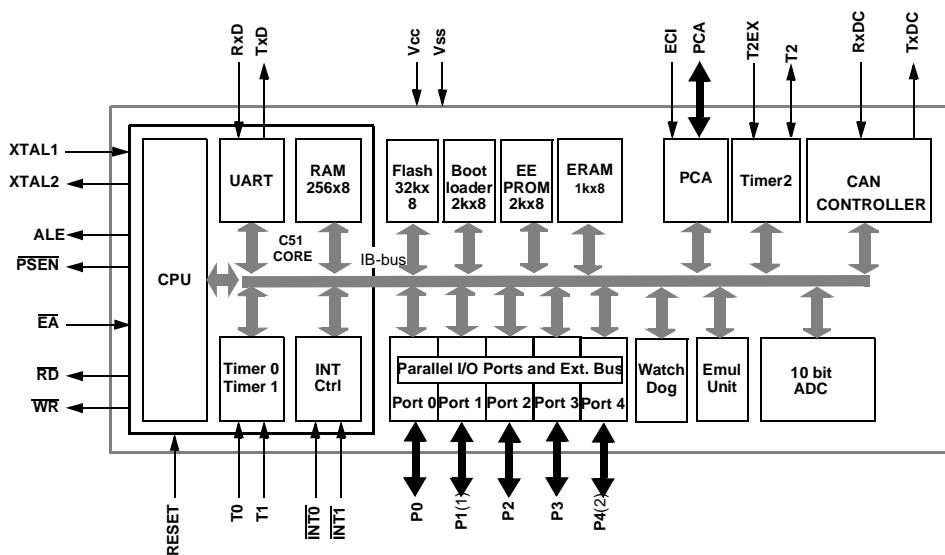
The T89C51CC01 is the first member of the CANary™ family of 8-bit microcontrollers dedicated to CAN network applications.

In X2 mode a maximum external clock rate of 20 MHz reaches a 300 ns cycle time.

Besides the full CAN controller T89C51CC01 provides 32K Bytes of Flash memory including In-System-Programming (ISP), 2K Bytes Boot Flash Memory, 2K Bytes EEPROM and 1.2-Kbyte RAM.

Primary attention is paid to the reduction of the electro-magnetic emission of T89C51CC01.

Block Diagram



- Notes:
1. 8 analog Inputs/8 Digital I/O
 2. 2-Bit I/O Port

1. Ask for availability

2 **T89C51CC01**

4129E-8051-03/02



APPENDICE E: INDICE ANALITICO

Simboli

+V opto 8, 12, 26, 35, 36
.LIB 38, 48
.SER 38, 42
/INT0 35, 53
/INT1 35, 53
 μ C/51 55
1-WIRE 8, 28, 35, 93
9 bit 10, 43, 45

A

Accessori 28, 46, 56, 99
Acquisizione della tastiera 40
ACQUISIZIONE SINGOLO INGRESSO DIG. CON DEBOUNCING, comando 80
ACQUISIZIONE SINGOLO INGRESSO DIGITALE, comando 80
ADDS View Point 61
"ALARM SEARCH" SU LINEA 1-WIRE, comando 98
Alimentazione 7, 12, 36
Alimentazione current loop 21
Ambienti di sviluppo 54
AMP4 CABLE 28
Antirimbalzo 80
Area codice 7, 49
Area dati 7, 51
Area stack 7, 53
Area grafica 86, 88
ASCII 39, B-1
Assistenza 1
ATTIVAZIONE CURSORE "BLOCCO" LAMPEGGIANTE, comando 76
ATTIVAZIONE CURSORE FISSO, comando 76
ATTIVAZIONE KEYCLICK CON MEMORIZZAZIONE, comando 73
ATTIVAZIONE KEYCLICK SENZA MEMORIZZAZIONE, comando 72
ATTIVAZIONE LED E BUZZER, comando 68
ATTIVAZIONE SINGOLA USCITA DIGITALE, comando 78
ATTIVAZIONE SINGOLA USCITA DIGITALE TEMP., comando 78
ATTIVAZIONE TEMPORIZZATA DEL BUZZER, comando 68
Autorepeat 11, 40
Avanquadro 11, C-2
AZZERAMENTO CONTATORE, comando 85

B

BASCOM 8051 55
BasicCAN 8
Baud rate 10, 42, 47, 56, 57
Bibliografia 102

Bit 1-WIRE 94
Bit per chr 10, 43, 47, 56
Blocchi 3
Boot loader 54
Buffer 10
Buffer di ricezione 40, 67, 71, 75, 80
Buzzer 7, 10, 29, 40, 68, 72, 73
Byte 1-WIRE 95
Byte di presenza 38, 64

C

CAN 8, 10, 22, 34, 35, 54
CANCELLA FINO A FINE PAGINA, comando 63
CANCELLA FINO A FINE RIGA, comando 63
CANCELLA PAGINA, comando 63
CANCELLA RIGA, comando 63
Caratteri 11, 39, 92, B-1
Caratteri definibili 39, 75, B-1
Caratteri speciali 39, 67, B-1
Caratteri utente 38, 74, B-1
Caratteristiche elettriche 12
Caratteristiche fisiche 11
Caratteristiche generali 10
Cariche elettrostatiche 1, 36
CCITT 16
CKS.AMP4 28
Clock 10
CN1, connettore alimentazione 14
CN2, connettore linea seriale 16
CN3, connettore linea CAN 32
CN4, connettore per uscite a relè 24
CN5, connettore per ingressi digitali optoisolati 26
Codice 7, 49
Codici tasti 38, 41, 72
Comandi 61, A-1
Comandi per cancellazione caratteri 61
Comandi per caratteri utente 74
Comandi per funzioni varie 66
Comandi per gestione attributi del cursore 76
Comandi per gestione EEPROM 64
Comandi per gestione I/O digitali bufferati 77
Comandi per gestione messaggi 69
Comandi per gestione tastiera 72
Comandi per gestioni alternative I/O digitali 84
Comandi per grafica 86
Comandi per la comunicazione 1-WIRE 93
Comandi per posizionamento cursore 61
Comunicazione seriale 16, 32, 42

Configurazione base 30, 32, 36, 38, 42

Connessioni 14, 101

Connettori 11, 14, 29

CN1 14

CN2 16

CN3 22

CN4 24

CN5 26

J6 28

Console 50

Consumi 12

Contatori 26, 84, 85

Contatti 24, 35

Contentitore 1, 101, C-1

Contrasto 30

Controllo di flusso 40, 47, 56

Coordinate grafiche 86

Corrente 12

Creazione del codice 59

Current loop 4, 10, 16, 20, 32, 35, 43

Cursore 61, 76, 86, 88

CURSORE A DESTRA, comando 61

CURSORE A INIZIO, comando 62

CURSORE E A SINISTRA, comando 61

CURSORE IN ALTO, comando 62

CURSORE IN BASSO, comando 61

D

Dati 7, 51

Dati grafici 90, 91

Dati in EEPROM 38

Debouncing 80

DEBUG 30, 54

DEFINIZIONE DI UN CARATTERE UTENTE, comando 75

DEFINIZIONE E SALVATAGGIO DI UN CARATTERE UTENTE, comando 75

Descrizione software 38

Dimensioni 11, C-1

DIN 72x144 11, C-1

Direttive 1, 35

DISATTIVAZIONE DEL CURSORE, comando 76

DISATTIVAZIONE KEYCLICK CON MEMORIZZAZIONE, comando 73

DISATTIVAZIONE KEYCLICK SENZA MEMORIZZAZIONE comando 72

DISATTIVAZIONE SINGOLA USCITA DIGITALE, comando 79

DISATTIVAZIONE SINGOLA USCITA DIGITALE TEMP., comando 79

Display 6, 10, B-1

Disposizione jumpers, connettori, trimmer, ecc. 29

Disturbi 34, 36

Driver seriali 33

DS18s20 93

E

EEPROM 7, 10, 38, 64, 65

EEPROM utente 38, 65

Espansione 99

Etichetta di personalizzazione C-3

Europeo, font 92, B-5

EXPS-1 36

EXPS-2 36, 100

F

Filtri 27, 36

Firmware 2, 38

Firmware di libreria 48

Firmware seriale 42

Fissaggio pannello frontale C-4

FLASH EPROM 7, 10, 49, 54

FLIP 54, 57

Font alfanumerici B-1

Font grafici 92, B-3

Forzatura 19, 34

Foto 5, 9, 13

Frequenze 10

Frontale 73, C-4

Funzioni alternative 84

G

Garanzia 1

H

Handshake 40, 47, 56

Home 39, 62

I

I2C BUS 28, 35

Impedenza 23, 34

Informazioni generali 2

Ingressi optoisolati 12, 26, 35, 77, 80

Inizializzazione 42, 52, 67

Installazione 14, C-2

Integrazione firmware 48

Interfacciamento I/O 35

Interrupt 35, 53, 84, 85

Introduzione 1

ISP 54, 57

J

J6, connettore per I/O TTL, 1-WIRE, I2C BUS 28

Jumpers 29, 30, 32, 54

K

Katakana, font 92, B-4

Keyclick 7, 40, 42, 72, 73

L

LADDER WORK 55

Lampeggio LED 68

Lato componenti 31

Lato stagnature 31

LED 6, 68

LETTURA BYTE DA EEPROM, comando 65

LETTURA CONTATORE, comando 85

LETTURA DEL BYTE DI PRESENZA, comando 64

LETTURA DEL CODICE SCHEDA, comando 66

LETTURA DEL NUMERO DI VERSIONE, comando 66

LETTURA DI N BYTES DA LINEA 1-WIRE, comando 95

LETTURA DI TUTTI GLI INGRESSI DIGITALI, comando 77

LETTURA DI UN BIT DA LINEA 1-WIRE, comando 94

LETTURA DI UN MESSAGGIO, comando 71

LETTURA NUMERO DELL'ULTIMO MESSAGGIO, comando 69

LETTURA SETTAGGIO DEI PIN MULTIFUNZIONE, comando 84

M

Malfunzionamento 7

Master 43

Master slave 43

"MATCH ROM" SU LINEA 1-WIRE, comando 97

MEMORIZZAZIONE DI UN MESSAGGIO, comando 70

Messaggi 11, 38, 69

Microcontrollore 8, 35, 57, D-1

Minifont 92, B-3

Misure 11, C-1

Modalità alfanumerica 86

Modalità comunicazione 43

Modalità grafica 88

Modalità operativa 39, 67

Modelli 5, 11, B-1

Montaggio 11, C-1

N

Normale **42**

Normative **1**

O

ONDA QUADRA SU SINGOLA USCITA DIG., comando **83**

ONDA QUADRA TEMP. DA 0 SU SINGOLA USCITA DIG., comando **82**

ONDA QUADRA TEMP. DA 1 SU SINGOLA USCITA DIG., comando **81**

Optoisolatori **26**

Opzioni **4, 32, 36, 38**

P

Parità **10, 43, 44, 47, 56**

PeliCAN **8**

Periferiche I2C BUS **28**

Periodi **81**

Personalizzazione **C-3**

Peso **11**

Pianta componenti **31**

POSIZIONAMENTO CURSORE ALFANUMERICO, comando **62**

POSIZIONAMENTO CURSORE GRAFICO, comando **88**

Programma applicativo **42, 48**

Protocollo elettrico **32**

Protocollo fisico **10, 42, 47, 56**

Protocollo logico **10, 43**

Punti di entrata **49, 50**

Q

Quote **C-1**

R

RAM **7, 10, 51, 53**

Rappresentazione caratteri **42**

"READ ROM" SU LINEA 1-WIRE, comando **96**

Relé **8, 24, 35, 77**

Reset **10, 67**

RESET DELLA COMUNICAZIONE, comando **67**

RESET GENERALE, comando **67**

RESET LINEA 1-WIRE, comando **94**

Rete **43**

Rete CAN **23**

Rete current loop **21**

Rete RS 485 **19**

RICHIESTA DISPONIBILITA' AD USARE EEPROM, comando **64**

RICONFIGURAZIONE DI UN TASTO, comando **72**

Risorse di bordo 10
Risorse usate 52
Ritardi 40
RITORNO A CAPO + NUOVA RIGA, comando 62
RITORNO A CAPO RIGA, comando 62
RS 232 4, 10, 16, 18, 32, 35, 43, 46, 56
RS 422 4, 10, 12, 16, 18, 32, 35, 43
RS 485 4, 10, 12, 16, 18, 34, 35, 43
RUN 30, 54

S

Scasso di montaggio C-2
Schede esterne 99
SCRITTURA BYTE SU EEPROM, comando 65
SCRITTURA DEL BYTE DI PRESENZA, comando 64
SCRITTURA DI N BYTES SU LINEA 1-WIRE, comando 95
SCRITTURA DI TUTTE LE USCITE DIGITALI, comando 77
SCRITTURA DI UN BIT SU LINEA 1-WIRE, comando 94
Seriale 4, 8, 10, 32, 42, 43, 46, 56
SETTAGGIO AREA GRAFICA, comando 88
SETTAGGIO DEI PIN MULTIFUNZIONE, comando 84
SETTAGGIO FONT GRAFICO 92
SETTAGGIO LUMINOSITA' DISPLAY, comando 66
SETTAGGIO MODALITA' ALFANUMERICA, comando 86
SETTAGGIO MODALITA' GRAFICA, comando 88
SETTAGGIO MODALITA' OPERATIVA, comando 67
Setup locale 42
Sicurezza 1
"SKIP ROM" SU LINEA 1-WIRE, comando 96
Slave 43
Software 38
Sovratensioni 36
SPAZIO INDIETRO, comando 63
Specifiche tecniche 10
Stack 7, 53
Staffe C-1, C-2
Stop bit 10, 42, 47, 56

T

T89C51AC2 8, 54, D-1
T89C51CC01 8, 54, D-1
Tabelle riassuntive comandi A-1
Tasca per etichetta C-3
Tastiera 6, 10, 40, 72
Temperatura 11, 93
Temporizzazioni 44, 53, 78, 79, 81, 82, 83
Tensione di alimentazione 12

Terminazione 12, 19, 23, 34
Time Out 44
TIMER 53, 84
Trimmer 29, 30
TTL 28, 35

U

Umidità 11
Uscite digitali a relé 8, 24, 77

V

Versione firmware 2, 66
Vista complessiva 9
VISUALIZZAZIONE DI MESSAGGI A SCORRIMENTO, comando 71
VISUALIZZAZIONE DI MESSAGGI, comando 70